1994 ARPA Image Understanding
Workshop, Monterey, CA,
November 1994

Image Descriptions for Browsing and Retrieval

Carlo Tomasi and **Leonidas J. Guibas**
Computer Science Department
Stanford University
Stanford, CA 94305

Abstract

At Stanford we started an effort to develop techniques for image data base browsing and retrieval based on picture content. Queries are pictorial descriptions of the desired images, and are formulated from a point-and-click graphical query editor that lets the user navigate in the space of description parameters. The retriever extracts a set of indices from the query and searches the data base with efficient, approximate nearest neighbor algorithms from computational geometry. The same navigator used by the query editor enables the user to view the possibly large set of retrieved images, or browse the whole data base.

1 Introduction

The bandwidth of today's communication systems and the capacity of information storage devices makes it possible to exchange and store large amounts of pictures in little time and at a low cost. Libraries of images, movies, catalogs, maps, aerial surveys, pictures from books and journals are widely available in electronic form. However, current retrieval systems make ineffective use of this wealth of pictorial information. In fact, still and moving images are by-and-large left uninterpreted in ever growing data bases. If the usefulness of information is a function of how well it can be accessed, image data bases are becoming more and more useless because of their very size.

At Stanford, we started an effort to develop techniques for browsing and retrieval based on picture content. As discussed for instance in [Gupta *et al.*, 1991], [Jain, 1992], [Faloutsos *et al.*, 1993], this task requires an investigation of fundamental issues of image representation. In this paper, we outline both the kind of system we plan to build and the conceptual challenges that lie ahead of us.

A query into an image data base can take many forms. Sometimes a user has a specific idea of the image she needs, leading to a focussed retrieval problem, but often the need is more generic, and an exploratory interaction (browsing) with the system is more appropriate. In either case, the user needs to tell the system what she wants, that is, she needs to formulate a query. The system then searches the data base for matches, and returns an image set that can be empty, small, or large.

Accordingly, the design of a browsing and retrieval system must address the following issues:

query language: what primitives and constructs are available to specify the query;

query editor: what tools are available to formulate a query and edit it if necessary;

image indexing: how to arrange images in the data base and how to generate indices into them for fast retrieval;

search algorithms: how to use the indices to respond to a query;

output presentation: how to display the results of the query, particularly when a large number of images is retrieved.

The design of a query language and of an image indexing scheme are closely related to each other, since the retrieval system must match queries to indices. We propose to unify these two aspects by defining a language for the description of both images and queries. Thus, a query is a *pictorial description* of what is being looked for: queries are "image-like". The far-reaching consequence of this decision is that image retrieval occurs at the syntactic level, and no attempt is made to extract semantics from the images. We thereby avoid image recognition which, although a worthy pursuit, has shown to be rather elusive over three decades of computer vision research. The price we pay for this choice is the need to provide a pictorial language that is rich and flexible enough for the user to describe potentially large and complex sets of images with a reasonably small number of constructs. At the same time, we need to enable the user to build the description in an intuitive way and by successive refinements, so the design of an appropriate graphical user interface with the retrieval system is of paramount importance.

Because queries and images can be described in the same language, browsing through a set of images is similar to navigating through queries. The query editor, the system for the presentation of the retrieved images, and the data base browser can now all be based on the unified notion of navigating in the space of descriptions. In fact, a query can be formulated by starting with a tentative description and modifying its parameters to satisfaction at the controls of a parameter-space navigation tool. The output of the retrieval can be displayed in a query-sensitive manner by emphasizing the parts of each image that are relevant to the query and by letting the user navigate through the images in a perceptually meaningful way, just as would be done during data base browsing.

The design of an image indexing scheme entails defining what part of an image description can be pre-stored to provide reliable retrieval cues. In general, a language combines primitives (words) into constructs (sentences). To avoid combinatorial complexity, indices for retrieval should be the language primitives, without constructs. This is similar to what is done in text retrieval, which is usually based on words rather than sentences [Salton, 1989]. Finally, search algorithms break down descriptions into indices, look up the corresponding images, and recombine the retrieved sets of images into the query result.

In conclusion, the identification of image and query descriptions not only lowers the level of processing from semantic image interpretation to syntactic matching, but also leads to the concept of a description language as the unifying theme of our project. The design of primitives is a computer vision problem, because it entails identifying the image features that can be used to describe an image in a perceptually meaningful way. Defining composition constructs is a more traditional language design problem, and has a strong geometrical and topological component.

In the next section, we outline the criteria for the design of indices, that is, of the primitives of our image description language. Then, in section 3 through 5, we discuss the structure of a query, including some of the language composition constructs, the issues in the design of search algorithms, and the problem of displaying the query results.

2 Indices

The success of our enterprise will depend on the extent to which we can discover and efficiently compute indices to an image that correlate well with the perceptual characteristics of the image: its partition into certain kinds of objects and their sizes, colors, textures, mutual positions, etc. These indices need to capture the way that a user remembers the image, or would describe the image to someone else. We can then hope that an image query drawn by the user would yield similar indices, and therefore lead us into that portion of index space where the original image hashes to.

Although there is evidence [Caelli *et al.*, 1988] that the notions of object size, relative position, color, texture, etc. are the right primitives on which to index, the selection of a particular computational representation of the indices will require research exploration. It would be especially desirable if the indices we compute remain relatively unchanged under certain transformations in the image induced by localized changes in the environment, such as a small changes in the lighting conditions, or in the viewpoint from which the scene was observed. Thus as part of our research in this area, we are exploring ways to make textures canonical under small projective, or at least affine, deformations, and we are experimenting with techniques drawn from [Leu, 1989], [Lee *et al.*, 1989], [Malik and Rosenholtz, 1993], [Binford and Levitt, 1993], [Shi and Tomasi, 1994], [Sato and Cipolla, 1994], and others.

The primary class of index functions we have been exploring are the coefficients of basis functions in various wavelet representations of the image. Multiresolution analysis has been a very active area of research in the last four to five years and has yielded very good results in a number of applications, especially image compression [Mallat, 1987], [Daubechies, 1992], [Cohen *et al.*, 1993]. A multiresolution representation is very natural for our image query application for a variety of reasons. For example, a user may want to search a data base of images using as a query a much lower resolution image, such as that obtained by a video camera. In the setting where the user draws the image query using the tools described above, it is natural that she may specify parts of the desired image at very different levels of detail: she may want to search for an outdoor scene with a grassy field in the lower part of the image, a blue sky above (both low detail), and a very particular type of house in a specific area of the image (high detail).

In this setting, we treat as image indices the significant wavelet coefficients – say those whose value is above some threshold.[1] Although this might do well in matching a video image to a high-resolution photograph, for the general image retrieval application there are several serious drawbacks of the existing wavelet technology. One is that the wavelet coefficients computed are very sensitive to the exact positioning of the image data on the image plane. An area that we are currently exploring is that of "anchoring" the wavelet basis elements to the image, so that they become independent of image translation. One possibility here is to simply compute a set of centered wavelet coefficients for every pixel, by convolving the image with a copy, centered at that pixel, of each wavelet function from a translation class.

A second problem is that every wavelet basis makes a tradeoff between spatial localization and frequency localization [Daubechies, 1992]. It has seemed to us that in describing an image hierarchically, spatial localization is most important in the upper levels of the hierarchy (corresponding to a coarse segmentation of the image into objects), while frequency localization is important in the lower levels (for describing the texture of objects). We are currently working on classes of wavelet functions that allow us to mix and match functions with good spatial localization at one level and others with good frequency localization at another.

A final, and possibly the most serious, problem with

[1] We remark here that for our application is is not important that we store enough coefficients to reconstruct the image. What we want to store is wavelet basis indices of those functions whose coefficients are large, and which therefore convey some significant perceptual spatial or frequency aspect of the image. We may in fact choose to record such indices for a number of different and redundant wavelet functions. Our goal is not image compression or reconstruction – it is to to test the similarity of images.

using wavelet coefficients as image indices is that all the standard wavelet bases do not respect sharp color or intensity boundaries in the image, thus effectively averaging "apples and oranges". It seems essential to be able to search for an object, even though in the query image the object is on a different background than on the final image. To this end, we are exploring ways to build multiresolution image representations that contain edge information at different scales, where the averaging implied in any multiscale representation is not done across image edges.

An alternate approach to the same problem consists of first doing a coarse segmentation of the image into regions. We then treat each region as a separate image and build a multiresolution structure for it using some standard wavelet basis. For a particular region, pixels outside of it are treated as "transparent" – so we actually need to build two multiresolution structures for each region, one representing the usual color/intensity/frequency data, and the other representing transparency (*alpha* in the computer graphics lingo). This separates the foreground from the background, as we need for searching, but it also allows us to composite the structures for the individual regions into one for the whole image, so we can, for example, display a reduced version of it [Berman *et al.*, 1994].

3 Queries

In the previous section, the primitives for image description were shown to be rather traditional features and patterns. Queries must be expressed in terms of these primitives. Consequently, what cannot be described pictorially cannot be the subject of a query. In the first phase of our research, we are restricting ourselves mainly to classifying *types* of images, rather than specific objects. For instance, we want to be able to distinguish between a natural landscape, the picture of a city, and a portrait. To this end, texture plays obviously a major role. However, we do not intend for textures, or even for more complex queries, to capture the essence of a particular type of image. With pictures of Manhattan, we expect and accept the retriever to also return images of printed circuit boards, which can be similar to city pictures in the regularity and type of their textures. If this response reduces a large data base to a substantially smaller set of images that still contains most of the city pictures, good progress will have been made.

Although our indices are a rich vocabulary of textures, colors, contour elements, contour junctions and so forth, we will illustrate how a query works through the example, again, of textural descriptors. Menus let the user select texture categories (deterministic, statistical, color, edges, intensity, and so forth). Within a given category, texture varies according to a set of parameters. Each vector of parameter values, a particular texture, is therefore a point in a vector space. In the user interface, this space is sparsely sampled to provide the user with examples of textures, each displayed in a small window on the screen. Clicking on these windows allows a first, coarse-level navigation within texture space. A set of cursors, the multi-dimensional equivalent of a joystick, lets the user navigate more locally and continuously by adjusting the parameter values of the texture descriptor. During this navigation, samples of the current texture and of those in a small neighborhood are displayed and updated for orientation.

A similar interaction can let the user select colors, contour junctions, and other patterns to be used for retrieval. The various patterns can then be interconnected with suitable constructs. While sophisticated visual languages are being investigated by several authors (see for instance [Rabitti and Savino, 1991] for a discussion of related issues), we will use only simple logical (*and*, *not*, *or*) and topological (*next-to*) operators. A stronger linguistic component can be introduced at a later stage, when the basic perceptual and computational issues have been appropriately addressed.

4 Search

Once the data base images have been hashed into some (possibly large) number of indices, we have the search problem of retrieving all the images whose indices match those of the query image. Our assumption here will be that the image data base is relatively static, and therefore it is advantageous to spend some time organizing it so that we can make queries efficient. The cost of a query will of course be output-sensitive, that is it will depend on the number of images that match the query. But there will also be an overhead term for accessing the data base that we want to make as small as possible, and definitely sublinear in the number of images stored. We now have a geometric problem in a space determined by the indices we have selected. A complicating factor over standard data base queries is that we do not expect to find exact matches. Rather, what we will do in the simplest case is *near-neighbor* searches for images whose indices are within some tolerance of those of the query image. We intend to use techniques from computational geometry [Arya and Mount, 1994] that provide efficient algorithms for such approximate nearest neighbor problems.

The way in which the data base returns to us the images matching the query conveys information about why each particular image returned was selected. Typically the matched images will be returned to us in a number of canonical collections where the images in each such collection share the "same reason" for matching the query. We intend to exploit this additional structure in displaying the results for the user and allowing her to navigate among the potential matches returned, as discussed in the following section.

5 Display

Suppose that a query returns 2,000 images. The user may choose to browse through them in order to pick a few, or perhaps in order to determine how the query should be refined or otherwise modified. The display of these pictures should be organized appropriately for easy browsing.

First, the display should be query-sensitive, in the sense that the part of each image that caused it to be retrieved should be emphasized. This can be achieved, for instance, by displaying the uninteresting part of the image at a lower resolution and brightness than the rest, or by blowing up the interesting regions, thereby providing a "caricature" of each image that reflects the interest of the user as expressed by the query.

Second, the pictures should be arranged in a space whose dimensions correspond to significant perceptual parameters. The user should be enabled to navigate in this space, just as she did when navigating in the space of textures (see section 3), or as would occur when navigating in the entire data base during browsing. Again, the query can provide the main dimensions of interest for this navigation, and the same navigator can be used as for the query editor.

6 Conclusion

With our work we wish to demonstrate that much can be done in image retrieval at the syntactic level alone, without having to solve the hard problems of image interpretation, semantic image segmentation, or even image compression. At the same time, however, we do plan to draw as heavily as possible from previous work in these areas. For instance, semantic segmentation is beyond both our needs and our possibilities, but good syntactic segmentation algorithms have been developed as early as the mid-seventies (see for instance [Horowitz and Pavlidis, 1976]). In our view, the main issues lie in the definition of a good set of retrieval indices, and this is the area in which we have chosen to start our research and develop our first demonstrations. A thorough understanding of these issues will be a solid foundation for the construction of effective retrieval systems, but it may also provide useful hints in the pursuit of the loftier goal of interpreting the semantics of images.

Acklowledgement: Leonidas Guibas wishes to thank David Salesin for suggesting to him the idea of multiresolution image representations. Insightful discussions with Adam Finkelstein, Dan Huttenlocher, David Salesin, and Yoav Shoham are gratefully acknowledged.

References

[Arya and Mount, 1994] S. Arya and D.M. Mount. Approximate nearest neighbor queries in fixed dimensions. In *Proceedings of the 5th SIAM Symposium on Discrete Algorithms (SODA)*, pages 271-280, 1994.

[Berman *et al.*, 1994] D.F. Berman, J.T Bartell, and D.H. Salesin. Multiresolution painting and compositing. In *ACM SIGGRAPH 94 Proceedings*, pages 85–90, 1994.

[Binford and Levitt, 1993] T.O.Binford and T. S. Levitt. Quasi-Invariants: Theory and Exploitation. ARPA Image Understanding Workshop, pages 819-830, 1993.

[Caelli *et al.*, 1988] T. Caelli, W. Bischof, and Z. Liu. Filter-based models for pattern classification. *Pattern Recognition*, 21(6):639–650, 1988.

[Cohen *et al.*, 1993] A. Cohen, I. Daubechies, and P. Vial. Wavelets on the interval and fast wavelet transforms. *Applied and Computational Harmonic Analysis*, 1:54–81, 1993.

[Daubechies, 1992] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, PA, 1992.

[Faloutsos *et al.*, 1993] C.Faloutsos,M.Flickner, W. Niblack, D. Petkovic, W. Equitz, and R. Barber. Efficient and effective querying by image content. Technical Report RJ 9453 (83074), IBM, Almaden Research Center, San Jose, CA, 1993.

[Gupta *et al.*, 1991] A. Gupta, T. Weymouth, and R. Jain. Semantic queries with pictures: The VIMSYS model. In *Proceedings of the 17th International Conference on Very Large Data Bases*, Barcelona, Spain, 1991.

[Horowitz and Pavlidis, 1976] S. L. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *Journal of the ACM*, 23(2):368–388, 1976.

[Jain, 1992] R. Jain, Editor. *NSF Workshop on Visual Information Management Systems.* Redwood, CA, 1992.

[Lee *et al.*, 1989] D. J. Lee, S. Mitra, and T. F. Krile. Analysis of sequential complex images, using feature extraction and two-dimensional cepstrum techniques. *Journal of the Optical Society of America, A*, 6(6):863–870, 1989.

[Leu, 1989] J. G. Leu. Shape normalization through compacting. *Pattern Recognition Letters*, 10:243–250, 1989.

[Malik and Rosenholtz, 1993] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. Technical Report UCB/CSD 93/775, EECS Division, University of California, Berkeley, CA, 1993.

[Mallat, 1987] S. G. Mallat. Scale change versus scale representation. In *Proceedings of the International Conference on Computer Vision*, pages 592–596, London, England, 1987.

[Rabitti and Savino, 1991] F. Rabitti and P. Savino. Query processing on image databases. In *2nd Working Conference on Visual Database Systems*, Budapest, Hungary, 1991.

[Salton, 1989] G. Salton. *Automatic text Processing - The Transformation, Analysis, and Retrieval of Information.* Addison-Wesley Pub. Co., Reading, MA, 1989.

[Sato and Cipolla, 1994] J. Sato and R. Cipolla. Extracting the affine transformation from texture moments. Technical Report CUED/F-INFENG/TR 167, Department of Engineering, University of Cambridge, 1994.

[Shi and Tomasi, 1994] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994.