Chapter 25

APPLICATION: FINDING IN DIGITAL LIBRARIES

Many collections of pictures are being digitized in the hope of better conservation, easier distribution, and better access. Once the pictures have been digitized, there remains the problem of finding the image we want in the collection. This is a subtle problem, because preparing a good text description of an image is difficult, so that text indexing systems are not always very much help. Furthermore, some collections of digital images are quite disorganised — for example, there is little prospect of preparing a text index for each image on the web, which contains a huge number of pictures.

Finding something in a collection of pictures or of video clips is a problem that appears in a variety of contexts. The increasing use of digital media is generating huge searchable collections which need to be managed. This **digital asset management** is a generalisation of the search problem. For example, digital projectors mean that film distributors can: easily change the film on display at a particular cinema; add or remove scenes according to local sensibilities; show alternate endings; etc. Presumably distributors will do this in ways that maximise revenue, but to do so they need to make sure that the particular item of digital content is where it needs to be, when it is wanted.

This is an interesting problem because collections are often enormous. Some of the collections of images described in [Enser, 1995] contain tens of millions of pictures. Indexing a large collection by hand involves a substantial volume of work. Furthermore, there is the prospect of having to reindex sections of the collection; for example, if a news event makes a previously unknown person famous, it would be nice to know if the collection contained pictures of that person. Finally, it is very hard to know what a picture is about:

> "the engineer requests an image of a misaligned mounting bracket . . . which only exists as an image described by a cataloguer as astronaut training . . ." [Seloff, 1990]

These observations mean it would be pleasant to have automated tools that can describe and find pictures in large collections.

The first issue to think about in an application is what users want; how do they look for pictures, and what does it mean to describe a picture? Vision is difficult, so it is quite often very difficult to do what they want, but their needs should inform our thinking about what to do. For this problem, relatively little is known about what users need or want, or what there work practices are. We then discuss, briefly and at a high level, what tools might be able to do, prefatory to a more detailed discussion of current approaches to the problem.

25.1 Background

Information retrieval is the study of systems that recover texts from collections using various kinds of information. The topic is interesting to us because information retrieval researchers have become adept at performance analysis. Typically, the performance of information retrieval systems is described in terms of **recall** — the percentage of relevant items actually recovered — and **precision** — the percentage of recovered items actually relevant. Usually systems are assessed by plotting precision at various different levels of recall (obtained by varying match thresholds), and averaging these plots over "typical" queries. Bad experiments are easy to do, because it is often quite hard to tell what is relevant (i.e. what should have been recovered by a query), and even harder to tell how many relevant items appear in a large collection. Researchers are occasionally tempted to believe that good systems should have high recall and high precision. In fact, different applications can have sharply different requirements. High recall can be a serious problem. For example, consider a high recall search of the net for pictures of children; a huge percentage of the people who have both a personal web page and children have put pictures of their offspring on their web page, so the result will be a gigantic collection of pictures.

In many applications it is sufficient to find a few pictures satisfying the search criteria, that is low recall is not a problem. In the Enser study, for example, requesters are seldom burdened with more than 10 pictures, whatever the subject matter. As another example, consider filtering internet connections for offensive images; as long as a manager can be reasonably certain that any protracted flow of such images will result in an alarm — say 50% recall — the tool is usable. Usually, retrieving many incorrect pictures (low precision) is a greater problem because it will confuse and annoy a user. There *are* applications where high recall is essential: for example, a user searching for pictures that invalidate a patent will want to see every possibly relevant picture.

25.1.1 What do users want?

The most comprehensive study of the behaviour of users of image collections is Enser's work on the then Hulton-Deutsch collection [Armitage and Enser, 1997; Enser, 1993; Enser, 1995] (the collection has been acquired by a new owner since these papers were written, and is now known as the Hulton-Getty collection). This

is a collection of prints, negatives, slides and the like, used mainly by media professionals. Enser studied the request forms on which client requests are logged; he classified requests into four semantic categories, depending on whether a unique instance of an object class is required or not and whether that instance is refined. Significant points include the fact that the specialised indexing language used gives only a "blunt pointer to regions of the Hulton collections" ([Enser, 1993], p. 35) and the broad and abstract semantics used to describe images. For example, users requested images of hangovers, physicists and the smoking of kippers. All these concepts are well beyond the reach of current image analysis techniques. As a result, there are few cases where one can obtain a tool that directly addresses a need. For the foreseeable future, the main constraint on the design of tools for finding images will be our quite limited understanding of vision.

However, useful tools can be built even with a limited understanding of vision (this extremely important point seems to be quite widely missed). It is hard to measure success. Enser suggests that the most reliable measure of the success of Hulton-Getty's indexing system is that the organisation is profitable. This test is a bit difficult to apply in practice, but there are a number of products available. IBM has produced a product for image search — QBIC (for Query By Image Content) — which has appeared in mass market advertising and appears to be successful. Similarly, Virage — a company whose main product is an image search engine — appears to be thriving (the company is described at [vir,]; a description of their technology appears in [Hampapur *et al.*, 1997]).

The main source of value in any large collection is being able to find items, so that we can expect to see more search tools. Potential application areas include:

- **military intelligence:** vast quantities of satellite imagery of the globe exist, and typical queries involve finding militarily interesting changes — for example, concentrations of force — occuring at particular places (e.g. [Mundy, 1995; Mundy, 1997; Mundy and Vrobel, 1994]).

- **planning and government:** satellite imagery can be used to measure development, changes in vegetation, regrowth after fires, etc. (e.g. [Smith, 1996]).

- **stock photo and stock footage:** commercial libraries — which often have extremely large and very diverse collections — sell the right to use particular images (e.g. [Armitage and Enser, 1997; Enser, 1995; Enser, 1993]).

- **access to museums:** museums are increasingly releasing collections, typically at restricted resolutions, to entice viewers into visiting the museum (e.g. [Holt and Hartwick, 1994a; Holt and Hartwick, 1994b; Psarrou *et al.*, 1997]).

- **trademark enforcement:** as electronic commerce grows, so does the opportunity for automatic searches to find violations of trademark (e.g. [Eakins *et al.*, 1998; Jain and Vailaya, 1998; Kato *et al.*, 1988; Kato and Fujimura, 1989; **?**; **?**]).

Missing Figure

Figure 25.1. *Iconic matching systems look for images that have exactly the same pixels as the image sought. Images can be matched very quickly using this criterion if they are appropriately coded. However, there are relatively few applications where this criterion applies, because it requires the user to have a fairly precise notion of what the picture sought actually looks like — in this example, the user would need to know how Scheile's model had posed.* figure from the paper "Fast Multiresolution Image Querying", Jacobs, Finkelstein and Salesin, SIGGRAPH-95, p unknown, in the fervent hope that permission will be granted

- **indexing the web:** indexing web pages appears to be a profitable activity. Users may also wish to have tools that allow them to avoid offensive images or advertising. A number of tools have been built to support searches for images on the web using techniques described below (e.g. [Cascia *et al.*, 1998; Chang *et al.*, 1997b; Smith and Chang, 1997]).
- **medical information systems:** recovering medical images "similar" to a given query example might give more information on which to base a diagnosis or to conduct epidemiological studies (e.g. [Congiu *et al.*, 1995; Wong, 1998]).

25.1.2 What can tools do?

There are three ways to look for an image: one can search for an exact match, for an image that "looks similar", or for an image with particular object-level semantics.

In **iconic matching**, we are seeking an image that looks as much like an example picture — which we might draw, or supply — as possible. The ideal match would have exactly the same pixel value in each location. This might be useful for users who have a clear memory of images that appear in the collection. The best known system of this form is due to Jacobs *et al.* of the University of Washington [**?**], illustrated in figure 25.1. Iconic matching tends to be used relatively seldom, because it is usually too difficult to remember what the image being sought looks like.

Appearance: In some applications — for example, finding trademark infringements — the structure of the whole image is important. In these applications, we think of the image as an arrangement of coloured pixels, rather than a picture of objects. This abstraction is often called **appearance**. The distinction between appearance and object semantics is somewhat empty — how do we know what's in an image except by its appearance? — but the approach is very important in

practice, because it is quite easy to match appearance automatically. Appearance is particularly helpful when the *composition* of the image is important. For example, one could search for stock photos using a combination of appearance cues and keywords, and require the user to exclude images with the right composition but the wrong semantics. The central technical problem in building a tool that searches on appearance is defining a useful notion of image similarity; section 25.2 illustrates a variety of different strategies.

Object level semantics: It is very difficult to cope with high-level semantic queries ("a picture of the Pope, kissing a baby") using appearance or browsing tools. Finding tools use elements of the currently limited understanding of object recognition to help a user query for images based on this kind of semantics, at a variety of levels. It is not known how to build finding tools that can handle high-level semantic queries, nor how to build a user interface for a general finding tool; nonetheless, current technology can produce quite useful tools for various special cases (section 25.3).

Browsing

Searching for images raises some very difficult problems (for example, assume you had a perfect object recognition system; how would you describe the picture you wanted? as [Armitage and Enser, 1997; Enser, 1995; Enser, 1993] show, using human indexers and language doesn't seem to work even close to perfectly). This means that tools tend to be quite erratic in practice. Fortunately, image collections are often highly correlated (usually as a result of the way they are created).

This correlated structure can be exploited if we provide a tool for browsing. A user then searches with the search tool, and can choose to look at items "near" to any hits returned by the search tool. Ideally, a browsing tools will display images that are "similar" — this could mean that they look similar, or have similar appearance, or lie close to one another in the collection, etc. — in a way that makes their similarity apparent, and provide some form of interaction that makes it possible to move through the collection in different "directions".

Building useful browsing tools also requires an effective notion of image similarity. Constructing a good user interface for such systems is difficult; desirable features include a clear and simple query specification process, and a clear presentation of the internal representation used by the program, so that failures are not excessively puzzling. Typically, users are expected to offer an example image or to fill in a form-based interface to search for the first image, and then can move around the collection by clicking on samples offered by the browsing tool.

25.2 Appearance

Images are often highly stylised, particularly when the intent of the artist is to emphasize a particular object or a mood. This means that the overall layout of an image can be a guide to what it depicts, so that useful query mechanisms can

be built by looking for images that "look similar" to a sample image, a sketched sample, or textual specification of appearance. The success of such methods rests on the sense in which images look similar. It is important to convey to the user the sense in which images look similar, because otherwise mildly annoying errors can become extremely puzzling. A good notion of similarity is also important for efficient browsing, because a user interface that can tell how different images are, can lay out a display of images to suggests the overall structure of the section of the collection being displayed. We will concentrate on discussing appearance matching rather than browsing, because the technical issues are so similar.

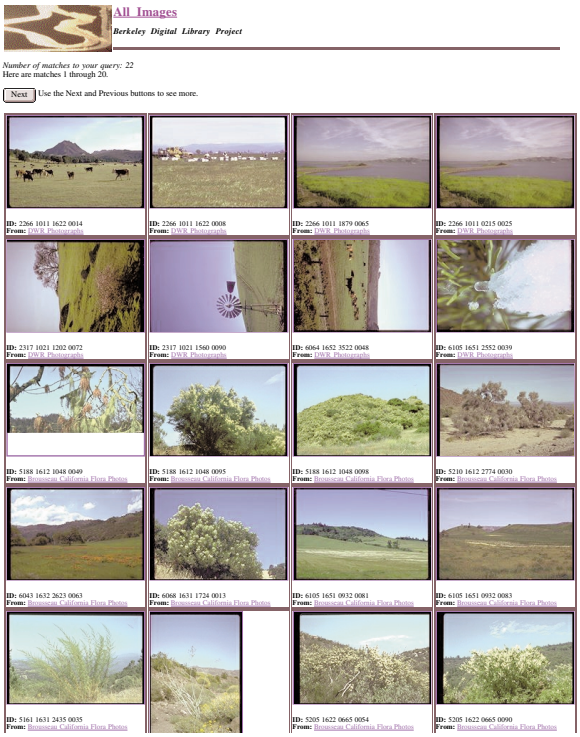

Figure 25.2. *Results from a query to the Calphotos collection that sought pastoral scenes, composed by searching for images that contain many green and light blue pixels. As the results suggest, such colour histogram queries can be quite effective.*

25.2.1 Histograms and correlograms

A popular measurement of similarity compares counts of the number of pixels in particular colour categories. For example, a sunset scene and a pastoral scene would be very different by this measure, because the sunset scene contains many red,

orange and yellow pixels and the pastoral scen will have a preponderance of green (grass), blue (sky) and perhaps white (cloud) pixels (e.g. figure 25.2). Furthermore, sunset scenes will tend to be similar; all will have many red, orange and yellow pixels and few others.

A **colour histogram** is a record of the number of pixels in an image or a region that fall into particular quantization buckets in some colour space (RGB is popular, for reasons we cannot explain). If the colour histogram for an image of an object fits into the histogram for an image (in the sense illustrated in figure **??**), then it is possible that that object is present in the image — if the illumination is not expected to vary all that much. This test can be quite sensitive to viewing direction and scale, because the relative number of pixels of a given colour can change sharply. Nonetheless, it has the advantage of being quick and easy, and applies to things like clothing which may have bright colours but little or no recognisable shape.

Colour histogram matching has been extremely popular; it dates back at least to the work of Swain and Ballard [Swain and Ballard, 1991], and has been used in a number of systems used in practice [Flickner *et al.*, 1995; Holt and Hartwick, 1994b; Ogle and Stonebraker, 1995]. The usefulness of colour histograms is slightly surprising, given how much image information the representation discards; for example, Chappelle *et al.* at ATT have shown that images from the Corel collection[1] can be classified by their category in the collection using colour histogram information alone [Chapelle *et al.*, 1999].

There is no record in a colour histogram of *where* coloured pixels are with respect to one another. Thus, for example, pictures of the French and UK flags are extremely similar according to a colour histogram measure — each has red, blue and white pixels in about the same number; it is the spatial layout of the pixels that differs. One problem that can result is that pictures taken from slightly different viewing positions look substantially different by a colour histogram measure (figure 25.3). This effect can be alleviated by considering the probability that a pixel of some colour lies within a particular pixel of another colour (which can be measured by counting the number of pixels at various distances). For small movements of the camera, these probabilities will be largely unchanged, so that similarity between these **colour correlograms** yields a measure of similarity between images. Requiring that colour correlograms be similar provides another measure of image similarity. The computational details — which have been worked out by Zabih and colleagues [Huang *et al.*, 1997; Huang and Zabih, 1998].

25.2.2 Textures and textures of textures

Colour histograms contain no information about the layout of colour pixels. An explicit record of layout is the next step. For example, a snowy mountain image will have bluer regions on top, whiter regions in the middle, then a bluer region at the bottom (the lake at the foot of the mountain), whereas a waterfall image

[1] A collection of 60, 000 images quite commonly used in vision research; available in three series from the Corel corporation, ********

Figure 25.3. *The top two figures have the same colour histogram; the red patch on the golfers shirt appears in the other image as brown looking flowers. These flowers are redder than they look (their hue is changed somewhat by the fact that they are small, and don't contrast strongly with their background), although not quite as red as the shirt. However, in the scheme of colour categories used the colours are regarded as equivalent. The bottom two figures have similar content, but quite different colour histograms; the person in the peach shirt appears in only one figure, and the person in the blue occupies more space in one than in the other. However, if one looks at a representation of the extent to which pixels of a given colour lie near pixels of some other colour, the pictures are quite similar — many blue pixels lie near either blue pixels, green ones or white ones. This information is captured by the colour correlogram. More information on colour correlograms can be found in [Huang* et al.*, 1997; Huang and Zabih, 1998]; picture by kind permission of R. Zabih.* Permission granted for Library Trends article; reproduced in the fervent hope that permission will be granted for the book, too

will have a darker region on the left and right and lighter vertical stripe in the center. These layout templates were introduced by Lipson, Grimson and Sinha at MIT; they can be learned for a range of images, and appear to provide a significant improvement over a colour histogram [Lipson *et al.*, 1997].

Looking at image texture is a natural next step, because texture is the difference

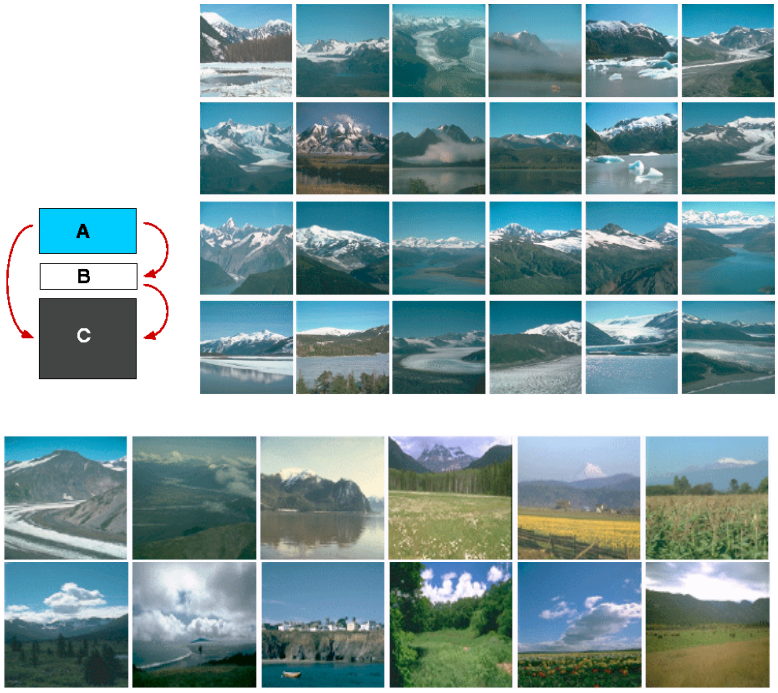

Figure 25.4. *Spatial layout of coloured regions is a natural guide to the content of many types of image. The figure on the* **top left** *shows a layout of coloured regions that suggests a scene showing snowy mountains;* **top right**, *the figures recovered by this criterion that actually do show snowy mountains;* **center**, *views of mountains that were in the collection but not recovered and* **bottom**, *images that meet the criterion but do not actually show a view of a snowy mountain. More detail appears in [Lipson* et al.*, 1997]; figure by kind permission of W.E.L. Grimson and P. Lipson.* Permission granted for Library Trends article; reproduced in the fervent hope that permission will be granted for the book, too

between, say, a field of flowers (many small orange blobs) and a single flower (one big orange blob), or a dalmation and a zebra. Most people know texture when they see it, though the concept is either difficult or impossible to define. Typically, textures are thought of as spatial arrangements of small patterns — for example, a tartan is an arrangement of small squares and lines, and the texture of a grassy field is an arrangement of thin bars.

The usual strategy for finding these subpatterns is to apply a linear filter to the image (see chapters **??** and **??**), where the kernel of the filter looks similar to the pattern element. From filter theory, we have that strong responses from these filters suggest the presence of the particular pattern; several different filters can be applied, and the statistics of the responses in different places then yield a decomposition of

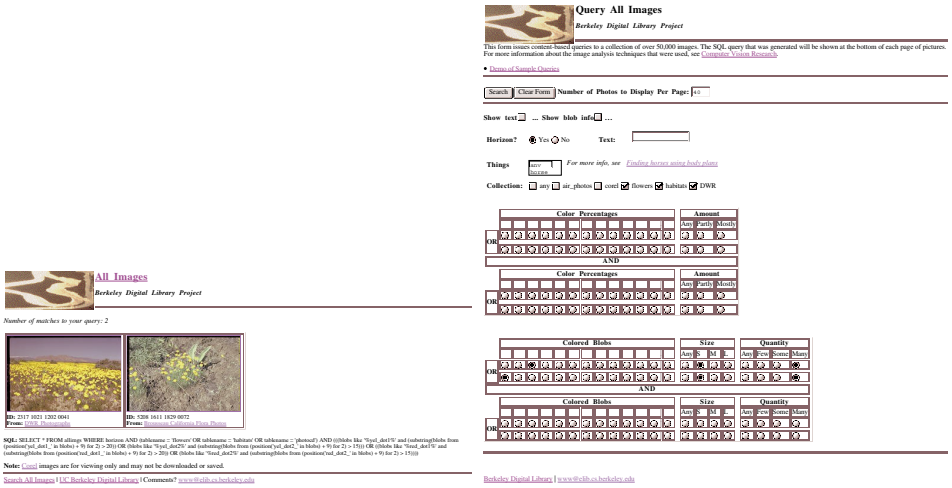

Figure 25.5. Left: *Specifying a query to the Calphotos collection using colour and texture information. I have selected images that have a horizon, and some red or yellow blobs, with the intention of finding views of fields of flowers; results appear in figure 25.5.* **Right:** *Images obtained from the Calphotos collection using the query of figure 25.5.*

the picture into spotty regions, barred regions, and the like [Ma and Manjunath, 1997a; Malik and Perona, 1989; Malik and Perona, 1990].

A histogram of filter responses is a first possible description of texture. For example, one might query for images with few small yellow blobs. This mechanism is used quite successfully in the Calphotos collection at Berkeley (`http://elib.cs.berkeley.edu/photos`; there are many thousands of images of California natural resources, flowers and wildlife). As figures 25.5-25.6 illustrate, a combination of colour and blob queries can be used to find quite complex images.

Texture histograms have some problems with camera motion; as the person in figure 25.3 approaches the camera, the checks on his shirt get bigger in the image. The pattern of texture responses could change quite substantially as a result. This is another manifestation of a problem we saw earlier with colour histograms; the size of a region spanned by an object changes as the camera moves closer to or further from the object.

A strategy for minimizing the impact of this effect is to define a family of allowable transformations on the image — for example, scaling the image by a factor in some range. We now apply each of these transformations, and measure the similarity between two images as the smallest difference that can be obtained using a transformation. For example, we could scale one image by each legal factor and look for the smallest difference between colour and texture histograms. This **earth-movers distance** — due to Rubner, Tomasi and Guibas at Stanford — allows a

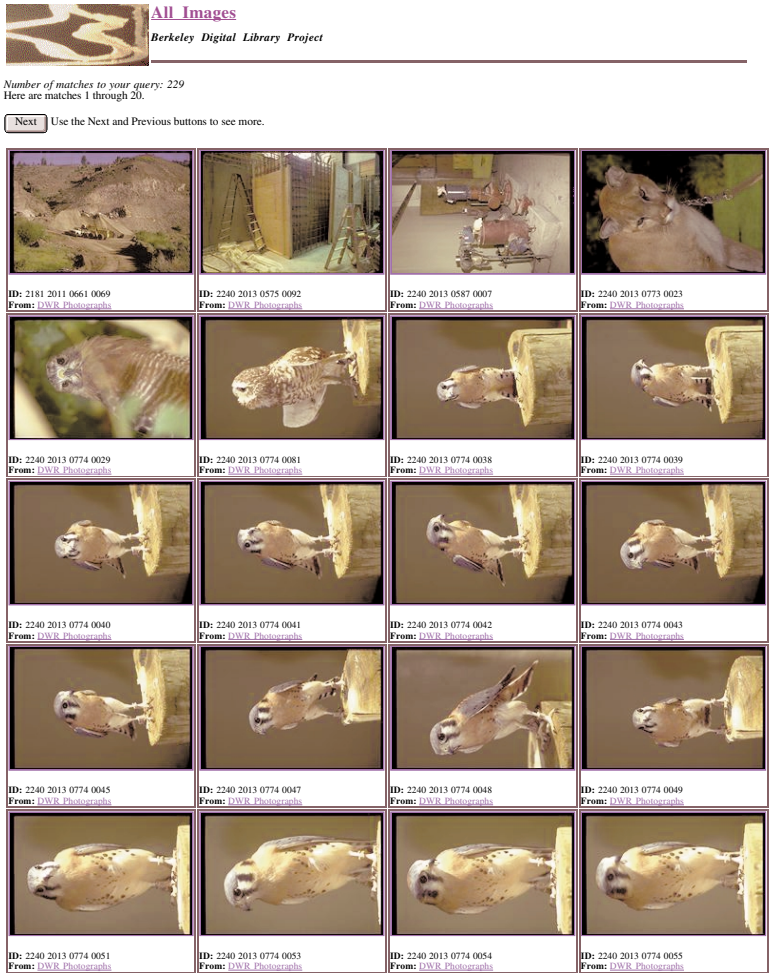

Figure 25.6. *Response images obtained by querying the Calphotos collection for images with at least one large brown blob, more than one small black blobs, and some green; this query is intended to find animals or birds.*

wide variety of transformations; furthermore, in [Rubner *et al.*, 1998], it has been coupled with a process for laying out images that makes the distance between images in the display reflect the dissimilarity between images in the collection. This approach allows for rapid and intuitive browsing (figure 25.7).

The spatial layout of textures is a powerful cue. For example, in aerial images, housing developments have a fairly characteristic texture, and the layout of this texture gives cues to the region sought. In the Netra system, built by Ma and

Figure 25.7. *Images laid out according to their similarity using the earth mover's distance (EMD). The EMD can be computed very fast so that displays like this — where distances between images on the display reflect the EMDs between them as faithfully as possible — can be created online. Large numbers of pictures returned from a query into an image database can thus be viewed at a glance, and a mouse click in the neighborhood of pictures that look similar to what the user is looking for tells the retrieval system where to search next. With this technology, users browse and navigate in an image database, just as they would browse through a department store. Because of the large number of images displayed, and their spatially intuitive layout, users quickly form a mental model of what is in the database, and rapidly learn where to find the pictures they need. More information on the EMD can be found in [Rubner* et al.*, 1998]; figure by kind permission of C. Tomasi.* Permission granted for Library Trends article; reproduced in the fervent hope that permission will be granted for the book, too

Manjunath at U.C. Santa Barbara, textures are classified into into stylised families (yielding a "texture thesaurus") which are used to segment very large aerial images; this approach exploits the fact that, while there is a very large family of possible textures, only some texture distinctions are significant. Users can then use example regions to query a collection for similar views; for example, obtaining aerial pictures of a particular region at a different time or date to keep track of such matters as the progress of development, traffic patterns, or vegetation growth (figure 25.8; [Ma

and Manjunath, 1998; Ma and Manjunath, 1997a; Manjunath and Ma, 1996b; Manjunath and Ma, 1996a]).

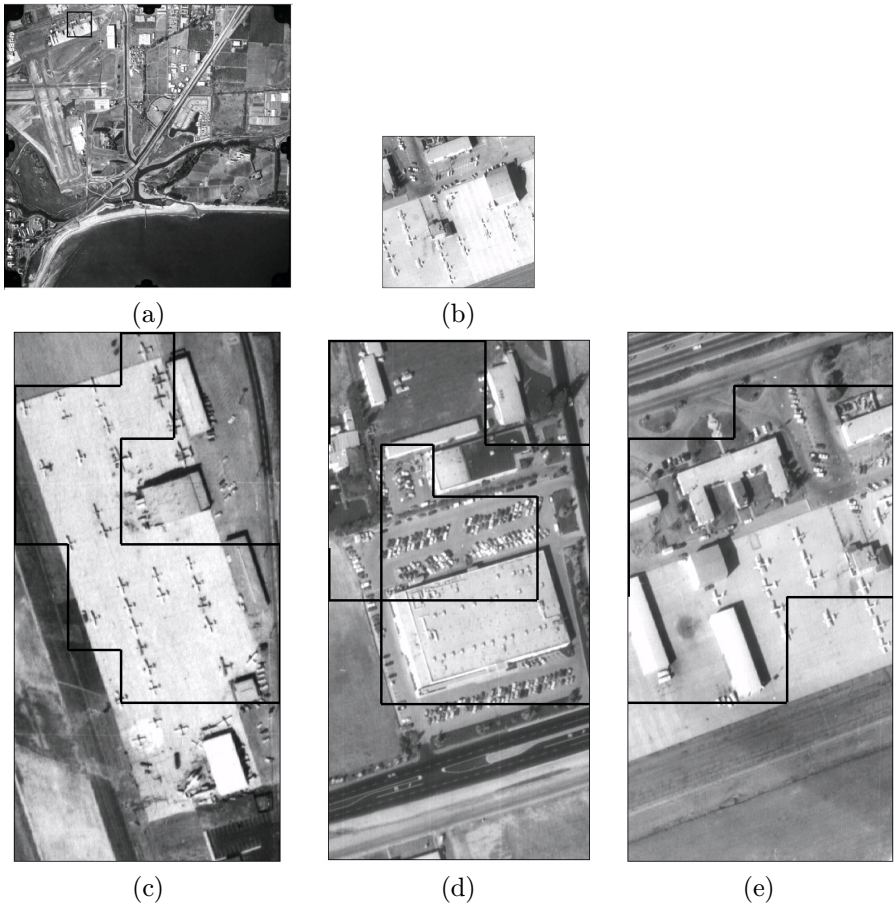

Figure 25.8. *A texture-based search in an aerial image. (a) shows the down-sampled version of the aerial photograph from which the query is derived. (b) shows a full-resolution detail of the region used for the query. The region contains aircraft, cars and buildings. (c)-(e) show the ordered three best results of the query. Once again, the results come from three different aerial photographs. This time, the second and third results are from the same year (1972) as the query photograph but the first match is from a different year (1966). More details appear in [Ma and Manjunath, 1997b]; figure by kind permission of B.S. Manjunath.* Permission granted for Library Trends article; reproduced in the fervent hope that permission will be granted for the book, too

Regions of texture responses form patterns, too. For example, if an image shows

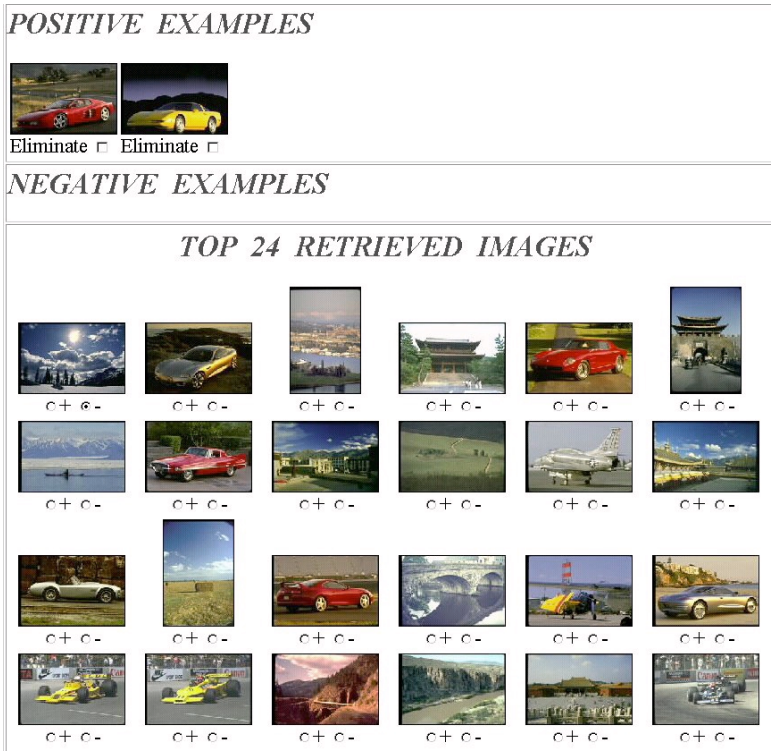

Figure 25.9. *Querying using the "texture of textures" approach. The user has identified two pictures of cars as positive examples; these would respond strongly to large horizontal bar filters, among others. This query results in a number of returned images, containing several images of cars; figure 25.10 shows the effect of refining this query by exhibiting negative examples. Figure by kind permission of P. Viola.* Permission granted for Library Trends article; reproduced in the fervent hope that permission will be granted for the book, too

a pedestrian in a spotted shirt, then there will be many strong responses from spot detecting filters; the region of strong responses will look roughly like a large bar. A group of pedestrians in spotted shirts will look like a family of bars, which is itself a texture. These observations suggest applying texture finding filters to the outputs of texture finding filters — perhaps recurring several times — and using measures of similarity of these responses as a measure of image similarity. This approach — due to DeBonet and Viola at MIT — involves a large number of features, so it is impractical to ask users to fill in a form. Instead, as in figure 25.9 and figure 25.10, the authors use an approach where users select positive and negative example images, and the system searches for images that are similar to the positive examples and dissimilar to the negative examples [De Bonet and Viola, 1997].

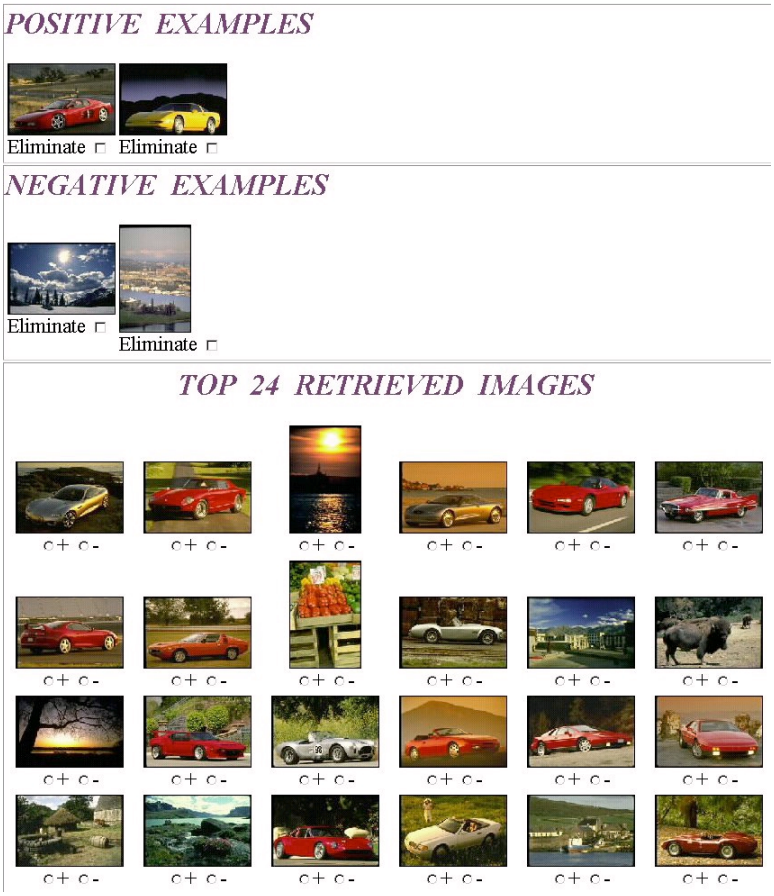

Figure 25.10. *Querying using the "texture of textures" approach. The query has been refined by providing some negative examples, yielding a response set that contains more car images. More detail on this approach appears in [De Bonet and Viola, 1997]; figure by kind permission of P. Viola.* Permission granted for Library Trends article; reproduced in the fervent hope that permission will be granted for the book, too

25.3 Finding

The distinction between appearance tools and finding tools is somewhat artificial; we can tell what objects are based on their differences in appearance. The tools described in this section try to estimate object-level semantics more or less directly. Such systems must first *segment* the image — i.e. decide which pixels lie on the object of interest. *Template matching* systems then look for characteristic patterns

associated with particular objects; finally, *correspondence* reasoning can be used to identify objects using spatial relationships between parts.

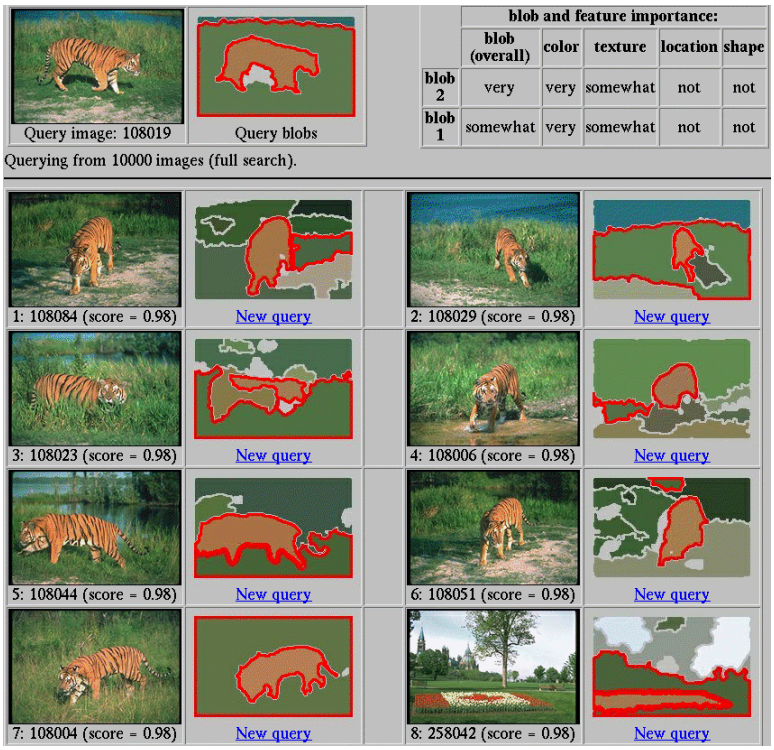

Figure 25.11. *Blobworld query for tiger images. Users of image databases generally want to find images containing particular objects, not images with particular global statistics. The Blobworld representation facilitates such queries by representing each image as a collection of regions (or "blobs") which correspond to objects or parts of objects. The image is segmented into regions automatically, and each region's color, texture, and shape characteristics are encoded. The user constructs a query by selecting regions of interest. The Blobworld version of each retrieved image is shown, with matching regions highlighted; displaying the system's internal representation in this way makes the query results more understandable and aids the user in creating and refining queries. Experiments show that queries for distinctive objects such as tigers and cheetahs have much higher precision using the Blobworld system than using a similar system based only on global color and texture descriptions. Blobworld is described in greater detail in [Belongie* et al.*, 1998; Carson* et al.*, 1999]; figure by kind permission of C. Carson.* Permission granted for Library Trends article; reproduced in the fervent hope that permission will be granted for the book, too

Structure in a collection is helpful in finding semantics, because it can be used to guide the choice of particular search mechanisms. Photobook — due to Pentland, Picard and Sclaroff at MIT — is a system that provides three main search categories: shape Photobook searches for isolated objects (for example, tools or fishes) using contour shape measured as elastic deformations of a contour; appearance Photobook can find faces using a small number of principal components; and texture Photobook uses a texture representation to find textured swatches of material [Pentland *et al.*, 1996]

25.3.1 Annotation and segmentation

A natural step in determining image semantics is to classify the type of material image patches represent; for example, "sky", "buildings", etc., as opposed to "blue", "grey". Generally, this kind of classification would need to be done with a mixture of user input (to establish appropriate categories and provide examples from those categories) and automatic annotation (for speed and efficiency). A combination of colour and texture features is often, but not always, distinctive of a region; a particular difficulty is knowing which features to use and which to ignore in classifying an image patch. For example, telling sky from grass involves looking at colour; telling concrete from sky may require ignoring colour and emphasizing texture. Foureyes — due to Minka and Picard at MIT — uses techniques from machine learning to infer appropriate features from user annotation practices, using across-image groupings (which patches have been classified as "sky" in the past?) and in-image groupings (which patches are classified as "sky" in this image?) [Minka and Picard, 1997; Picard and Minka, 1995; Minka, 1996]. As a result, a user annotating an image can benefit from past experience, as illustrated in figure 25.12.

Humans decompose images into pieces corresponding to the objects we are interested in, and classification is one way to achieve this *segmentation*. Segmentation is a crucial idea, because it means that irrelevant information can be discarded in comparing images. For example, if we are searching for an image of a tiger, it should not matter whether the background is snow or grass; the tiger is the issue. However, if the whole image is used to generate measures of similarity, a tiger on grass will look very different from a tiger on snow. These observations suggest segmenting an image into regions of pixels that belong together in an appropriate sense, and then allowing the user to search on the properties of particular regions. The most natural sense in which pixels belong together is that they come from a single object; currently, it is almost never possible to use this criterion, because we don't know how to tell when this is the case. However, objects usually result in image regions of coherent colour and texture, so that pixels that belong to the same region have a good prospect of belonging to an object.

VisualSEEK — due to Smith and Chang at Columbia — automatically breaks images into regions of coherent colour, and allows users to query on the spatial layout and extent of coloured regions. Thus, a query for a sunset image might specify an orange background with a yellow blob lying on that background [Smith

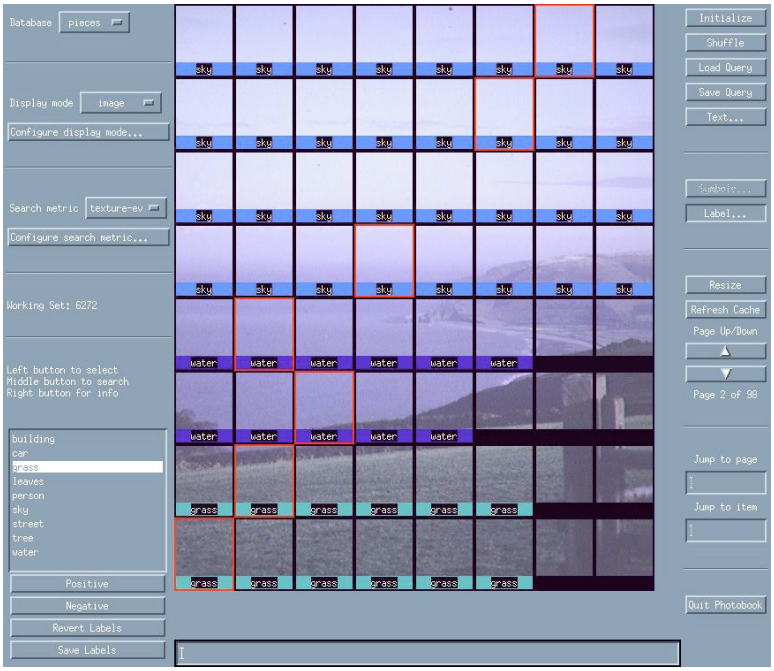

Figure 25.12. *An image annotated with Foureyes; red patches of image have been explicitly labelled "sky", "grass" or "water". Other labels are inferred by Foureyes from previous annotations and from the information given in these examples. More information appears in [Minka and Picard, 1997]; figure by kind permission of R. Picard.* Permission granted for Library Trends article; reproduced in the fervent hope that permission will be granted for the book, too

and Chang, 1996].

Blobworld is a system built at Berkeley by Carson *et al.* that represents images in terms of a collection of regions of coherent colour and texture [Belongie *et al.*, 1998; Carson *et al.*, 1999; Carson *et al.*, 1997; Carson and Ogle, 1996]. The representation is displayed to the user, with region colour and texture displayed inside elliptical blobs, which represent the shape of the image regions. The shape of these regions is represented crudely, because details of the region boundaries are not cogent. A user can query the system by specifying which blobs in an example image are important, and what spatial relations should hold (figure 25.11).

25.3.2 Template matching

Some objects have quite characteristic appearance for a wide range of viewing directions and conditions. Template matching is an object recognition strategy that finds objects by matching image patches with example templates (chapter **??**). A

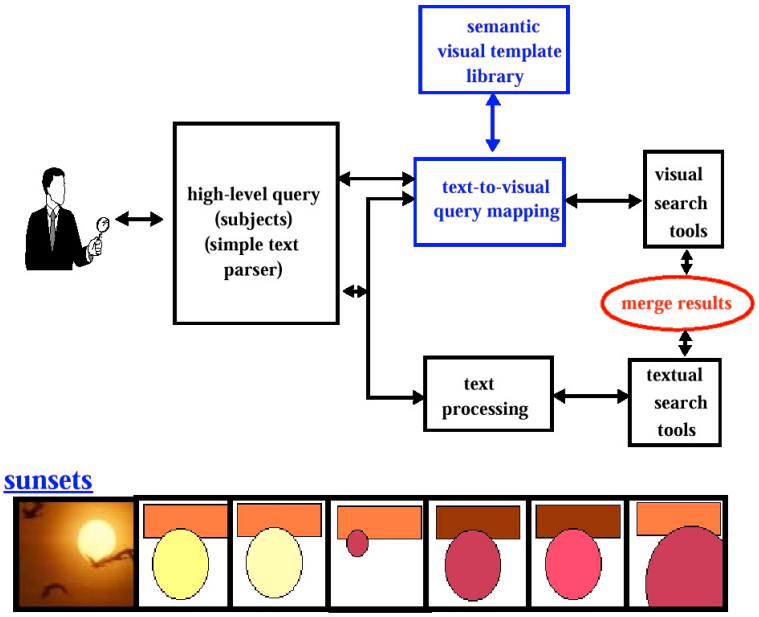

Figure 25.13. *To remove the burden of drawing detailed low-level sketches from users, the Semantic Visual Template system helps users to develop personalized search templates. The library of semantic templates can then be used to assist users in high-level multi-media query. The top figure shows the overall structure of a system using semantic templates, and the lower figure shows a template for sunset images. More details appear in [Chang* et al.*, 1998b]; figure by kind permission of S-F. Chang.* Permission granted for Library Trends article; reproduced in the fervent hope that permission will be granted for the book, too

natural application of template matching is to construct whole-image templates that correspond to particular semantic categories (figure 25.13 and [Chang *et al.*, 1998b]). These templates can be constructed off-line, and used to simplify querying by allowing a user to use an existing template, rather than compose a query.

Face finding is a particularly good case for template matching. Frontal views of faces are extremely similar, particularly when the face is viewed at low resolution — the main features are then a dark bar at the mouth, dark blobs where the eyes are, and lighter patches at the forehead, nose and mouth. This means that faces can be found, independent of the identity of the person, by looking for this pattern. Typical face finding systems extract small image windows of a fixed size, prune these windows to be oval, correct for lighting across the window, and then use a learned classifier to tell whether a face is present in the window [Rowley *et al.*, 1996a;

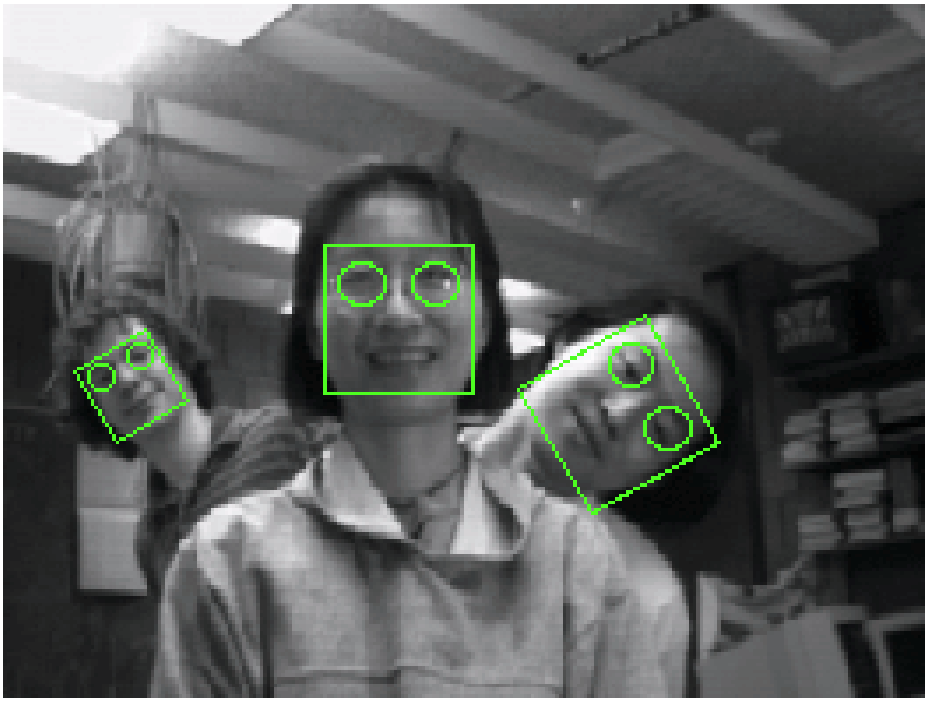

Figure 25.14. *Faces found by a learned template matching approach; the eyeholes in the green mask are to indicate the orientation of the face. More details appear in [Rowley* et al.*, 1996b; Rowley* et al.*, 1998a; Rowley* et al.*, 1998b]; figure by kind permission of T. Kanade.* Permission granted for Library Trends article; reproduced in the fervent hope that permission will be granted for the book, too

Rowley *et al.*, 1996b; Rowley *et al.*, 1998a; Poggio and Sung, 1995]. Our figure (figure 25.14) illustrates work along these lines due to Rowley, Baluja and Kanade at CMU; a similar approach has been used successfully by Sung and Poggio at MIT. This process works for both large and small faces, because windows are extracted from images at a variety of resolutions (windows from low resolution images yield large faces, and those from high resolution images yield small faces). Because the pattern changes when the face is tilted to the side, this tilt must be estimated and corrected for; this is done using a mechanism learned from data [Rowley *et al.*, 1998b]. Knowing where the faces are is extremely useful, because many natural queries refer to the people present in an image or a video.

25.3.3 Shape and correspondence

If object appearance can vary, template matching becomes more difficult as one is forced to adopt many more templates. There is a good template matching sys-

tem for finding pedestrians, which appears to work because pedestrians tend to be seen at low resolution with their arms at their sides [Oren *et al.*, 1997]. However, building a template matching system to find people is intractable, because clothing and configuration can vary too widely. The general strategy for dealing with this difficulty is to look for smaller templates — perhaps corresponding to "parts" — and then look for legal configurations.

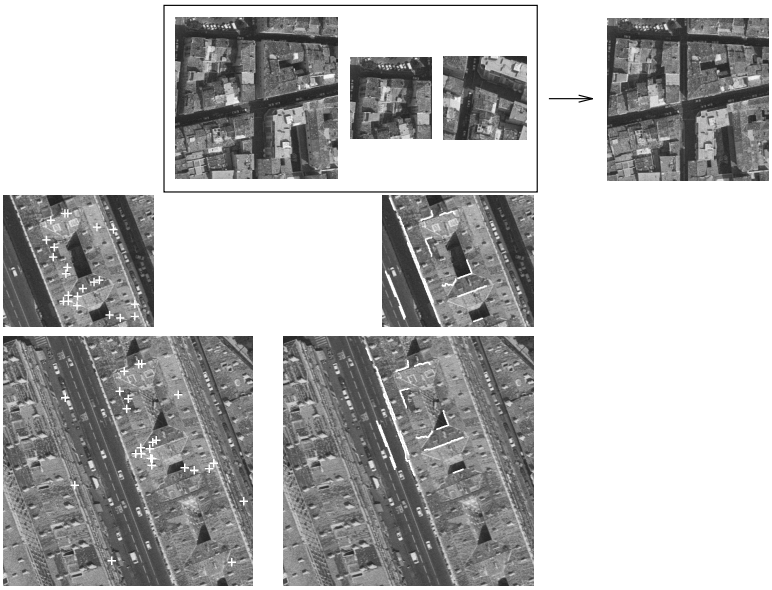

Figure 25.15. *Images can be queried by detecting "interest points" on the image and then matching on configurations of these points, based on their geometric distribution and the grey level pattern surrounding each point. The matching process is very efficient (it uses 'indexing'), and is tolerant of missing points in the configuration. In the example shown here, the image on the top right can be correctly retrieved from a collection of paintings, aerial images and images of 3D objects using any of the images on the top left. Interest points used during the matching process, shown in white, for a query image (small inset on left) and the best match (bottom left). Additional evidence is obtained from the image-image transformation to confirm the match is correct; on the right, edges which match under this transformation in the query (inset) and the result (bottom right). Notice that the two images have been taken from different viewpoints so that the building's shape differs between images. Also the scenes are not identical because cars have moved. Further details are given in [Schmid and Mohr, 1997b]; figure by kind permission of A. Zisserman.* Permission granted for Library Trends article; reproduced in the fervent hope that permission will be granted for the book, too

One version of this technique involves finding "interest points" — points where

combinations of measurements of intensity and its derivatives take on unusual values (for example, at corners). As Schmid and Mohr of INRIA and Zisserman of Oxford have shown, the spatial arrangement of these points is quite distinctive in many cases. For example (as figure 25.15 illustrates) the arrangement of interest points in an aerial view of Marseille is unaffected by the presence of cars; this means that one can recover and register aerial images of the same region taken at different times of day using this technique. Furthemore, once interest points have been matched, an image-image transformation is known, which can be used to register the images. Registration yields further evidence to support the match, and can be used to compare, say, traffic by comparing the two images at specific points.

This form of correspondence reasoning extends to matching image components with object parts at a more abstract level. People and many animals can be thought of as assemblies of cylinders (corresponding to body segments). A natural finding representation uses grouping stages assemble image components that could correspond to appropriate body segments or other components.

Forsyth of U.C. Berkeley and Fleck of HP Labs have used this representation for two cases; the first example identifies pictures containing people wearing little or no clothing. This is an interesting example: firstly, it is much easier than finding clothed people, because skin displays very little variation in colour and texture in images, whereas the appearance of clothing varies very widely; secondly, many people are interested in avoiding or finding images based on whether they contain unclad people. This program has been tested on an usually large and unusually diverse set of images; on a test collection of 565 images known to contain lightly clad people and 4289 control images with widely varying content, one tuning of the program marked 241 test images and 182 control images (more detailed information appears in [Forsyth *et al.*, 1996; Forsyth and Fleck, 1996]). The second example used a representation whose combinatorial structure — the order in which tests were applied — was built by hand, but where the tests were learned from data. This program identified pictures containing horses, and is described in greater detail in [Forsyth and Fleck, 1997]. Tests used 100 images containing horses, and 1086 control images with widely varying content; for a typical configuration, the program marks 11 images of horses and 4 control images.

25.4 Video

While video represents a richer source of information than still images, the issues remain largely the same. Videos are typically segmented into *shots* — short sequences that contain similar content — and techniques of the form described applied within shots. Because a large change between frames is a strong cue that a shot boundary has been reached, segmenting video into shots is usually done using measures of similarity like those described in section 25.2 (e.g. [Boreczky and Rowe, 1996]).

The motion of individual pixels in a video is often called *optic flow* and is measured by attempting to find pixels in the next frame that correspond to a pixel in this (correspondence being measured by similarity in colour, intensity and texture).

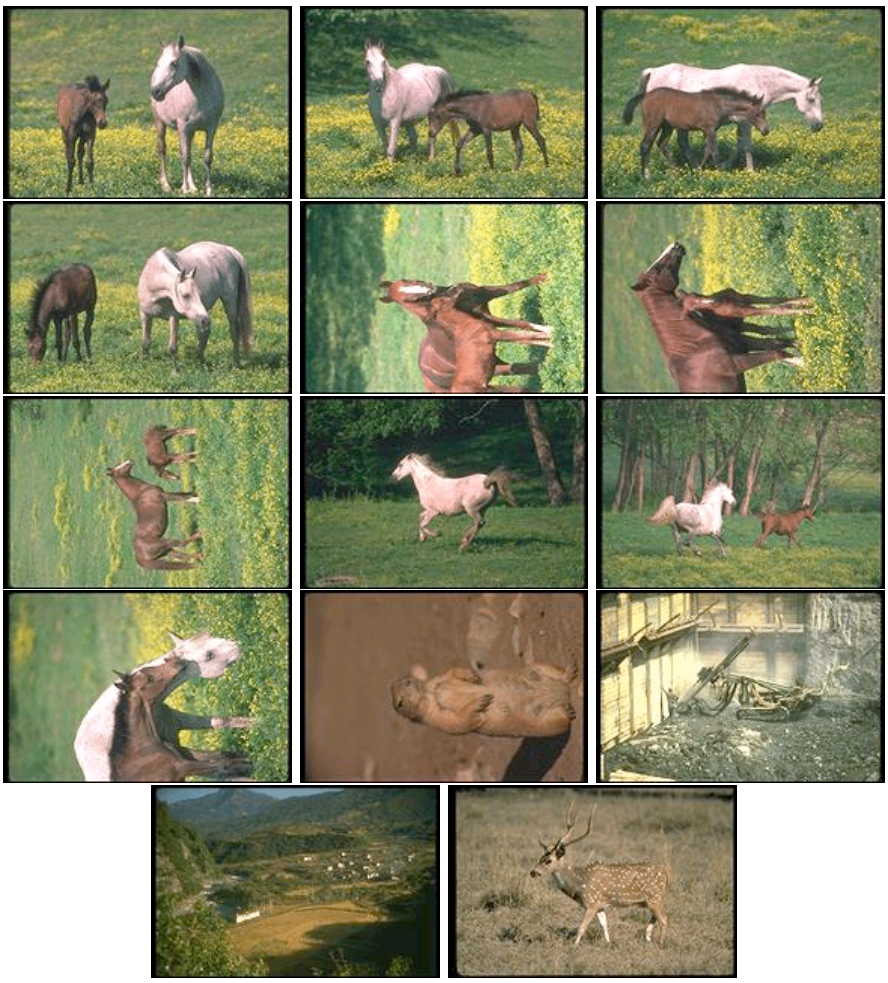

Figure 25.16. *Images of horses recovered using a body plan representation, from a test collection consisting of 100 images containing horses, and 1086 control images with widely varying content. Note that the method is relatively insensitive to aspect, but can be fooled by brown, horse-shaped regions. More details appear in [Forsyth and Fleck, 1997].*

In principle, there is an optic flow vector at each pixel, forming a *motion field*. In practice, it is extremely hard to measure optic flow reliably at featureless pixels, because they could correspond to pretty much anything. For example, consider the optic flow of an egg rotating on its axis; there is very little information about what the pixels inside the boundary of the egg are doing, because each looks like the other.

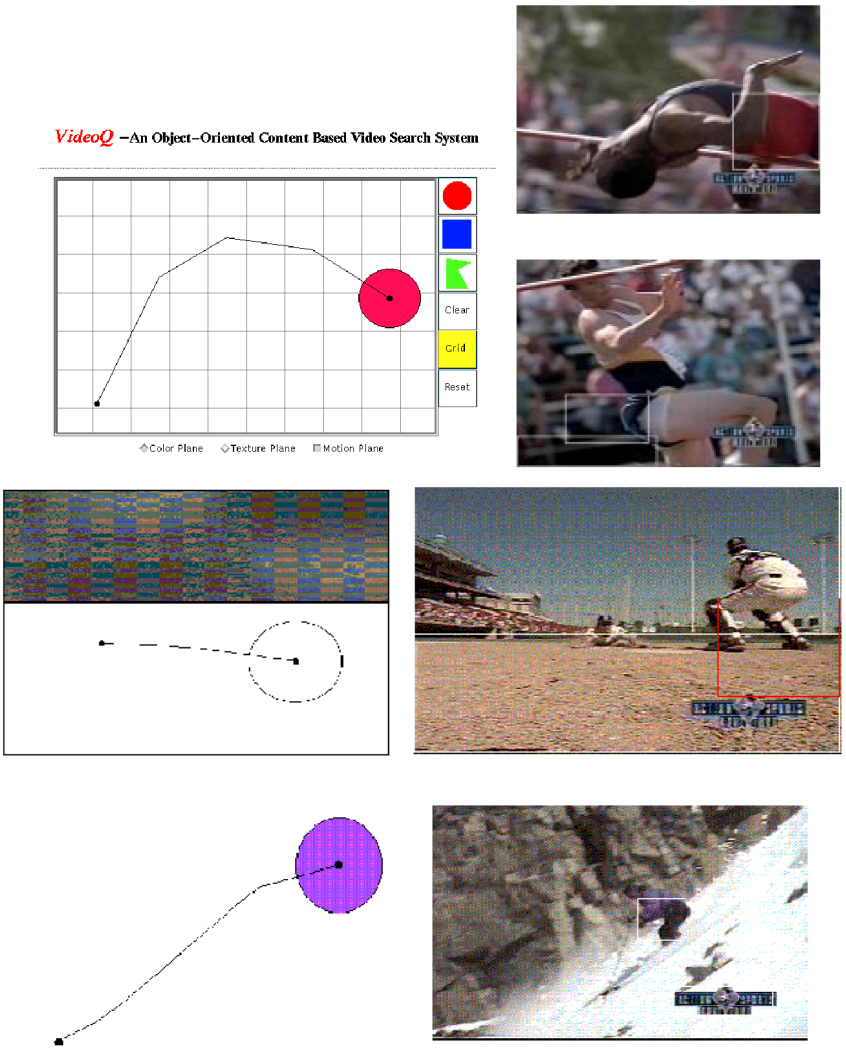

Figure 25.17. *Video can be represented by moving blobs; sequences can then be queried by specifying blob properties and motion properties desired. The top left shows a query for a blob moving along a parabolic arc, sketched in the user interface for the VideoQ system. Top right shows frames from two sequences returned. As the center (baseball) and bottom (skiing) figures show, the mechanism extends to a range of types of motion. More details appear in [Chang* et al.*, 1997a]; figure by kind permission of S-F. Chang.* Permission granted for Library Trends article; reproduced in the fervent hope that permission will be granted for the book, too

Motion fields can be extremely complex; however, particularly if there are no moving objects in the frame, it is possible to classify motion fields corresponding to the camera shot used. For example, a pan shot will lead to strong lateral motion, and a zoom leads to a radial motion field. This classification is usually obtained by comparing the measured motion field with a parametric family (e.g. [Sawhney and Ayer, 1996; Smith and Kanade, 1997]).

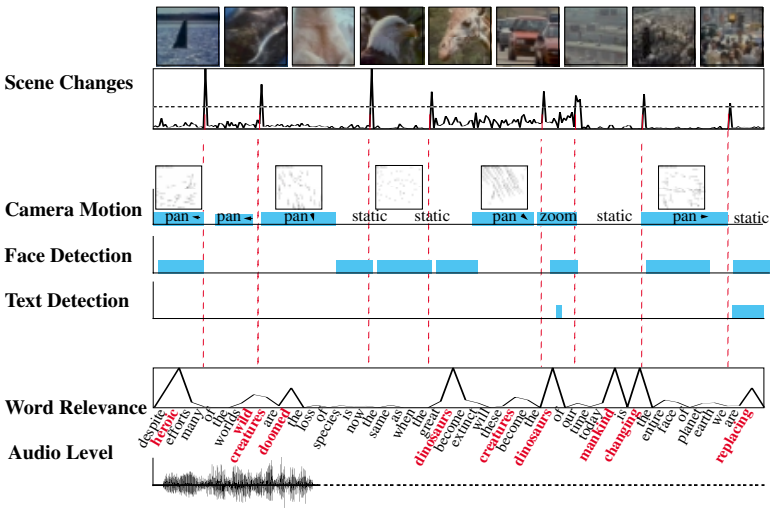

Figure 25.18. *Characterizing a video sequence to create a skim. The video is segmented into scenes. Camera motions are detected along with significant objects (faces and text). Bars indicate frames with positive results. Word relevance is evaluated in the transcript. More information appears in [Smith and Kanade, 1997]; figure by kind permission of T. Kanade.* Permission granted for Library Trends article; reproduced in the fervent hope that permission will be granted for the book, too

The Informedia project at CMU has studied preparing detailed skims of video sequences. In this case, a segment of video is broken into shots, shots are annotated with the camera motion in shot, with the presence of faces, with the presence of text in shot, with keywords from the transcript and with audio level (figure 25.18). This information yields a compact representation — the "skim" — which gives the main content of the video sequence (details in [Smith and Kanade, 1997; Wactlar

et al., 1996] and [Smith and Christel, 1995; Smith and Hauptmann, 1995].

25.5 Discussion

For applications where the colours, textures and layout of the image are all strongly correlated with the kind of content desired, a number of usable tools exist to find images based on content. There has been a substantial amount of work on user interfaces and browsing, although this work is usually done to get a system up and running, rather than through an explicit study of user practices. Because colour, texture and layout are at best a rough guide to image content, puzzling search results are pretty much guaranteed. There is not yet a clear theory of how to build interfaces that minimize the impact of this effect. The most widely adopted strategy is to allow quite fluid browsing.

When queries occur at a more semantic level, we encounter deep and poorly understood problems in object recognition. Object recognition seems to require segmenting images into coherent pieces and reasoning about the relationships between those pieces. This rather vague view of recognition can be exploited to produce segmented representations that allow searches for objects independent of their backgrounds; furthermore, some special cases of object recognition can be handled explicitly. It is not known how to build a system that could search for a wide variety of objects; building a user interface for such a system would present substantial problems, too. A natural research focus is the activities of people. In the next few years, it may be possible to find pictures of, say, a politician kissing a baby using entirely automatic methods.