Image Retrieval with Sketches and Compositions

Raj Kumar Rajendran and Shih-Fu Chang
Department of Electrical Engineering, Columbia University
New York, NY 10027

ABSTRACT

We present an image search technique that is based on characterizing the strongest edges of an image. It allows queries in the form of rough line-sketches that outline the basic form and composition of an image. This permits the user to quickly and easily transform a mental picture of an image into a query. Characterizing images with signatures that represents their strongest edges, and comparing these to a dynamically generated signature of the user's sketch achieves this search. The edge signature is generated over multiple scales to account for the variation in detail of the user's line-sketch. Edge-coherence, a measure of the perceptual strength of an edge, based on its continuity, is used in generating the signature.

Our search technique is fast enough to allow the system to respond with results at the completion of each stroke of a query sketch. The algorithm is robust and invariant to scale, rotation and translation. We also show how this edge-based search can be merged seamlessly with the more traditional color-region and shape-based searches to produce a search interface that is expressive, intuitive, simple, and works in the absence of sample images.

1. INTRODUCTION

Because of the proliferation of multimedia on the Internet, it is important to be able to search and sift through images. While there has been considerable research on the problem of finding images that are perceptually similar to a given image, little work has been done on searching for images when a sample image is not available. A solution to the latter problem requires additional steps: first, an interface that allows the user to quickly and intuitively express the pictured image needs to be designed, then the user's query has to be transformed into a form that can be compared to candidate images.

The lack of significant work on these two problems leaves image search in a state whose parallel in text-search would be one where search engines only accepted "cut-and-paste" of full-sentences as input. Such a restriction would diminish much of the usefulness of search engines whose power lies in allowing the user to quickly compose queries and in requiring that the query needs to only convey the general idea for the search rather than the details.

Most of the research to date has concentrated on the simpler problem of similarity-search -finding images that are perceptually similar to a given image. Color and texture have been the most used features while shape and form measures have been occasionally used. These features are calculated a-priori for database images and dynamically for the query image when a search is initiated. The search is conducted by ranking database images by the similarity of their features to the query's features as measured by a chosen distance measure [4,5,6].

Work where shape is the primary feature, or where sketches and compositions can be used as queries are less common. Ravela et al [2] have attempted to capture information about the shapes and forms of an image by characterizing the intensity surface of an image. The curvature and the phase (direction of the largest slope) are recorded at various points in the image. A search is carried out by calculating the curvature and phase of the query image and comparing it at various scales to candidate images. Their system differs from ours by requiring a candidate image to conduct the search. Huttenlocher et al [3] have used the Hausdorff distance to compare images on the basis of their edges. However the Hausdorff distance is not translation-invariant, which makes it computationally expensive. Salesin et al [1] have developed one of the few image search systems that allow the user to query the system by composing a sketch on a palette, rather than using an existing image. Similarity between the composed query and database images is computed by comparing the large wavelet-coefficients of their color-planes. The implementation is fast and allows the user to search for images that have a spatial distribution of color similar to the query composition. Results are also updated at the completion of each stroke.

The system of Ravela is limited, as it does not allow the use of composed queries. While the system of Salesin does allow a color-composition to be the query, it does not directly index edge-information and is not invariant to translation, rotation and scale.

We have implemented a system that allows a user to search for images by making a rough sketch or composition of the basic forms and shapes of an image, and tested it on a database of 5000 images. It was fast enough for results to be updated at the completion of each stroke of the sketch. The interface was intuitive and simple enough for third graders to use, and the results were robust with respect to the scale, location and orientation of the query sketch.

Before we continue, we will define a few terms that are used through the paper. A *line-sketch* is a line drawing created with a few strokes of the mouse and a *color-composition* is a rough composition in color created by choosing a pen-color and line-width, then creating various color swatches on a blank canvas. We distinguish between *similarity-search* where a natural image is used as the query and *sketch-based search* where a synthetic image is used. Sketch-based search is further broken down into *color-region search* where the query is a synthetic composition in color, and an *edge-search*, where the query is a line-sketch created by the user. If the primary feature of a similarity search is shape, we refer to it as a *shape-based search*. The top-left images of figures 2a, 2b and 2c show examples of edge-based, shape-based and color-region based search.

The rest of the paper is organized as follows: Section 2 presents our search model, Section 3 details our methodology and presents our implementation of edge-based search. Section 4 briefly discusses composition-search and similarity-search in our system and shows how the three different kinds of search presented can be merged into one seamless system. Section 5 presents our results and Section 7 concludes by summarizing our work.

2. THE MODEL

When searching for images, users often have an object or picture in mind, representations of which they hope to find in images. This mental-image can be material and concrete, such as "a teacup", or abstract such as "a fuzzy recollection of a painting with the sun painted in blue on a black background". If the database of images is large, browsing is not an option; annotations are often unavailable, and similarity searching is not viable if a sample image is not available. However, if an effective search can be carried out based on a rough-sketch of the object, or a simple composition of the image, relevant images can be found easily.

Traditionally, artists have used two forms of rough sketches to capture a mental image or idea for a painting: the *croquis,* which is a line-sketch that serves as a reminder of the structure of a scene or event, and the *pochade* which is a rough composition, often in color, that records the mood and general impression of a scene. Such rough sketches, we believe, are effective expressions of the general idea of an image and are a natural form of query expression. We believe that providing the user the ability to express queries in both these forms creates an excellent user interface for expressing image queries.

2.1 Edge-based Search

Since the first of the two forms mentioned, the line-sketch, is an attempt to capture the most salient *forms* of an image, it has a high correlation to the strong edges of an image. In a physical interpretation of an image, strong edges can be seen as object boundaries: delineations of one object from another or a foreground from a background. Therefore if the edge thresholds are chosen carefully, the edges will have a high correlation to the boundaries of the most visibly significant objects in the image, or in other words, they will correspond to the *form* or *structure* of the image.

2.1.1 The Physical Model

Since the strong-edges of an image are highly correlated to object boundaries, they are a natural choice to match against a user's sketch. However sketches vary widely from user to user or even between two sketches of the same object by the same user. The variants are the level-of-detail in the sketch, the size and orientation of objects depicted, and the physical distribution of the strokes. Therefore, for a sketch-based search technique to be useful, it has to be robust with respect to scale, translation and the level-of-detail. We use a multi-scale representation to make our algorithm robust with respect to detail and a curvature-direction representation to make the algorithm insensitive to translation and scale. We create a multi-scale representation by decomposing edges, once identified, into multiple-scales based on their strength and length. The scales of this decomposition will vary in the *number* and *detail* of the dominant objects of the image represented. It is useful to visualize this decomposition as a representation of the shapes of the dominant objects broken down by number and level-of-detail. As we include weaker edges, we add less-dominant objects and add detail to more dominant ones.

For ease of exposition, we will refer to two variants of the multi-scale decomposition: an absolute decomposition and a cumulative decomposition. In the cumulative version the detail scales also contain the coarser scales, while in the absolute version, they do not. The version being referred to will be clear from the context.

2.1.2 Computational Model

The computational model of our edge-based search can be broken down into four parts: edge-detection, multi-scale decomposition of the edges, representation of the decompositions by signatures at each scale and the matching algorithm. Each image is initially transformed into an edge-based representation by passing it through a gradient filter, which results in a representation of the image as a set of edge-points. Then, connected edge-points are collected into edge-trees and decomposed into multiple-scales based on edge-gradients and edge-lengths. Statistics are then compiled for the edge-trees at each scale and concatenated to yield the signature of an image.

When a sketch is presented as a query, a similar signature of just *one* scale is constructed for it, since its edge-gradients are constant. As the sketch is already in the form of edges the edge-detection and multi-scale decomposition steps are dispensed with, and the signature calculated directly. Additionally, the scale of the image-decomposition with corresponding detail is determined and passed to the distance measure, which computes the distance by comparing the sketch's signature to this *one* level in the image's decomposition. Figure 1 shows the components discussed.

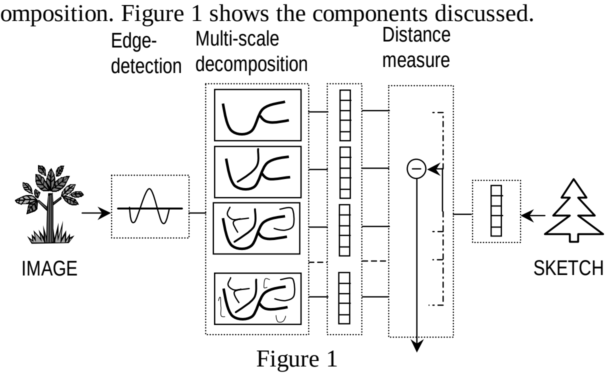

Figure 1

Mathematically the processes described till now can be seen as a transformation of an image so that it is represented in the same space as the query -as a set of edges. A multi-scale representation of these edges is required because the query can vary in the amount of detail present in its edges. By breaking up the edges in the image into a multi-scale representation, some combination of scales can be made to yield a representation that contains an amount of detail similar to that of the query.

2.2 Similarity Search based on Shape

A slight variant of the sketch-based search model outlined above can be used to carry out similarity searches based on shape. In this case the query is a natural image and therefore a multi-scale representation of the edges can be extracted -as opposed to the edge-based query where only a single-scale representation is available. Since both query and image have a multi-scale representation, the distance metric computes the distance across all scales rather than just one.

2.3 Color-Composition Search

For color-region search, the image is first transformed into a perceptually uniform color model such as HSV. The color space is then quantified and the image transformed so that it uses a small number of colors. Region segmentation algorithms are used to find large areas of similar color for which features such as dominant-color, texture, shape, and location are computed. These identified regions and their features are stored as a global list across all images in the database.

When the user composes a color-region query, it is also segmented into regions based on color, and features similar to those computed for the candidate images are computed for each region. A list of good matches for each region is drawn up from the global list of regions. These lists are merged and the images that best match are chosen based on an aggregate measure or the spatial relationships of candidate regions in an image. The reader is referred to [5,6] for a more detailed exposition.

3. EDGE-BASED SEARCH

Intuitively, a line-sketch attempts to capture the significant *lines* of a form or object. By significant lines, we mean those lines that most evoke and express the form of an object or image. These lines will have a rough correspondence to the strong and long edges in an image. By strong, we mean those edges that have large gradients and by long we mean continuous edges. While it is simple to compute the strength of edges measured as their gradient, it is not yet possible to synthesize long semantically meaningful edges. When two edges meet, or conversely when a fork is encountered in an edge being synthesized, it is nearly impossible, with current technology, to decide if either one of the two paths correspond to a continuation, or if the fork is an end point for the current edge.

One technique used to improve the probability of making the right decision is to extract edges at different scales, starting at a coarse subsampled image, and to add details to the edges at finer scales. The idea is that only semantically significant edges show up in highly subsampled images, and fewer decisions need to be made about branches in a coarse representation of an image. This method breaks down if many branching decisions need to be made at the subsampled image. A second technique is to use statistical properties to help decisions at forks. Since it is likely that the geometrical properties of a semantically meaningful curve will remain constant along its entire length, an edge is more likely, at a fork, to continue along the path that has similar curvature, direction and other geometrical properties. This technique breaks down, for example, at an image of the corner of a room, which appears as three lines with similar geometric properties meeting at a point.

Since it is clear that it is not possible to classify edge-points into semantically meaningful edges, we deal with ensembles of edges rather than individual edges. We call this ensemble an *edge-tree* and define it to be a set of connected edge-points whose nodes are the fork-points encountered earlier and its arcs the detected edges. To realize these edge-trees, Canny's edge-detector is applied after the image is smoothed with a Gaussian filter. First an edge-map is synthesized by applying two thresholds: a high-threshold is used to identify strong edges, and these strong edges are extended by following edge-points that fall above a low-threshold. All other edge-points are discarded. To construct edge-trees, end-points of edges are identified in the edge-map, and the edges traced recursively till all edge-ends are reached. After this second round of edge synthesis, we have a *set of edge-trees* while we had an edge-map before. A length-threshold is applied at this point and all edge-trees that have fewer than a predetermined number of edge-points are dropped.

To keep the statistical properties calculated in the next section consistent across images, the thresholds used to detect edge points are varied so that the fraction of points in the image detected as edge-points remains approximately the same across all images.

3.1 The Multi-Scale Representation

Since sketches created by users vary widely in the amount of detail present -some users will draw detailed sketches, while others may sketch just two or three lines to represent an idea or object- our representation is decomposed into different *scales***.** The edge-trees extracted in the previous section are decomposed into multiple-scales based on their strength and length. The strength of an edge-tree is calculated as the sum of the gradients of all edge-points that belong to the edge-tree. This measure classifies edge-trees according to two perceptually significant measures –the gradient of an edge and its length. All edge-trees are decomposed into different scales according to this measure. The coarse scale will only have long, strong edges while a fine scale will have short, weak edges in addition. This classification of edge-trees by length, we term *edge-coherence*.

The intuition here is that very long edges represent gross, high-level features in an image corresponding to the long strokes of a user's sketch while the shorter edges represent the embellishing edges the user adds as detail to the sketch. The idea is that if the user creates the sketch with just a few broad strokes, the first scale with only the long edges will provide a good match, and if the user includes detail one of the intermediate scales will provide a good match.

3.2 Edge-Characterization

Once we have the set of edges-trees representing an image decomposed into multiple layers, we compute a short signature characterizing each layer. This characterization needs to be robust and invariant to translation, scale, and orientation. It also needs to distinguish between different forms or shapes present in the edge-tree; different characteristic signatures should result for different shapes such as semi-circles, rectangles or parallelograms.

After some experimentation we chose to use curvature-directions histograms –we found invariant moments too sensitive to noise. Histograms have excellent invariance properties; *direction-histograms* distinguish between different shapes while *curvature-histograms* give us the ability to differentiate by size –a small circle, for example, has a large curvature, while a large circle has a small curvature. In this representation a circle will be characterized by a constant curvature-histogram and a single-spike curvature-histogram while a rectangle will be characterized by four spikes in its direction-histogram and a single small-valued spike in its curvature-histogram. Two circles, meanwhile, will differ only by the location of the spike in their curvature-histogram. Direction and curvature are computed on parametric representations of edges according to the equations below, after the curves have been appropriately smoothed.

$$C=\frac{\left|(\partial^2 x/\partial t^2 * \partial y/\partial t)-(\partial^2 y/\partial t^2 * \partial x/\partial t)\right|}{\left[(\partial x/\partial t)^2+(\partial y/\partial t)^2\right]^{3/2}} \qquad D=\arctan\left(\left|\frac{\partial y/\partial t}{\partial x/\partial t}\right|\right)$$

In summary, the image signature consists of direction-histograms and curvature-histograms calculated at different edge-length and edge-strength scales. Direction-histograms are scale-invariant, curvature-histograms are rotation-invariant, and both are translation-invariant.

3.3 The Distance Metric

A user's sketch may differ from the image's edge-map in the choice of objects depicted in addition to the level-of-detail mentioned earlier. While sketching a query the user will always sketch the object of interest, but may or may not depict secondary objects that do not necessarily have to be present. The solution to the above problem is a measure that returns a small distance when one representation matches a sub-set of another. Histogram Intersection Distance is such a metric and is defined as

$$HID = \sum_i \min\left(a(i), b(i)\right) \Big/ \min\left(\sum_i a(i), \sum_i a(i)\right)$$

where **a** and **b** are the two histograms and *a(i), b(i)* indicate the **i**th component of the two histograms. We use this HID as our distance measure.

With this we conclude our discussion of Edge-Based Search.

4. MERGING SEARCHES

Our system allows the user to compose three kinds of synthetic queries: a line-sketch, a synthetic image composed on a clear screen with a painting tool or a natural image that has been modified by painting over it. When one such query is created and a search initiated, the system first computes a color-histogram of the synthetic-image, and determines the type of the query. This determination initiates different algorithms: if the image is a line-sketch, edge-based search is carried out, if it is predominantly synthetic, a composition-search comparing significant color-regions proceeds. If on the other hand, the image is mostly natural a similarity search based on shape is used. For images with significant amounts of synthetic and natural content, both composition and similarity searches are used and the results merged by weighting the two sets of results with the proportion of synthetic and natural content present in the query.

5. RESULTS

We used our technique to implement an image search system which was used by New York city-schools in K-12 art education. The database was a set of approximately 5000 images of paintings and other objects of art from the MESL database and users were allowed to search for images by using all the search techniques mentioned above. The interface was intuitive and simple enough for third graders to use. Sketch-based search was fast enough for results to be updated at the completion of each stroke of the query sketch. The results were robust with respect to the scale, location and orientation of the query sketch. Results of an edge-based search, a shape-based search and a color-region search are shown in Figure 2a, 2b and 2c respectively, where the image on the top-left is the query.

Our algorithm yielded slightly better results compared to the algorithms of Ravela for shape-based search. While slightly slower than the algorithms of Salesin for color-region search, edge-based searches, which are unique to our system, produced quick, robust results and proved the most intuitive query interface. These edge-based searches worked best for simple sketches; as the complexity of the sketch increased, increasing ambiguity, their consistency decreased.

6. SUMMARY and CONCLUSION

A novel technique for the search of images based on line-sketches and color compositions was presented in this paper. The proposed scheme is simple, intuitive, and robust to translation, rotation and scale. Many parameters can be fine-tuned to adapt this technique to specific applications.

Other than its obvious use in sifting through the growing number of images on the Internet, it can be used in education where children can learn about shapes, forms and visual composition. This scheme was the basis of an art-education application we built and was used to teach visual-literacy in K-12 schools in New York [7].

Due to the nature of our algorithms, our technique is suitable only for simple forms. Techniques need to be developed that allow the search for complex sketches and compositions. The mouse also remains an awkward drawing tool, so other input methods and interfaces need to be considered to smooth the human-computer interaction.

7. ACKNOWLEDGEMENT

This work was partly funded by a generous grant from the AT&T Foundation.

8. REFERENCES

[1] Eric J. Stollnitz, Tony D. Derose, and David Salesin, *Wavelets and Computer Graphics: Theory and Applications*, Morgan Kaufman Pub.: ISBN 1-5860-375-1.

[2] S. Ravela, R. Manmatha, *Retrieving Images by Similarity of Visual Appearance*. In the Proceedings of the IEEE Workshop on Content Based Access of Videos and Libraries, 1997.

[3] Daniel P. Huttenlocher, William J. Rucklidge, *A MultiResolution Technique for Comparing Images Using the Hausdorff Distance.*

[4] Niblack W., Barber R., Equitz W., Flickner M., Glasman E.,Petkovic D., Yanker P., Faloutsos C., and Taubin G., *The QBIC Project: Querying Images by Content Using Color, Texture and Shape*. IBM Research Journal, No 9203, February 1, 1993.

[5] J. R. Smith and S.-F. Chang, *Integrated Spatial and Feature Image Query*, ACM Multimedia Systems Journal, 1998.

[6] D. Zhong and S.-F. Chang, *Video Object Model and Segmentation for Content-Based Video Indexing*, IEEE Int. Conf. on Circuits and Systems, June 1997, Hong Kong.

[6] D. Zhong and S.-F. Chang, *Video Object Model and Segmentation for Content-Based Video Indexing*, IEEE Int. Conf. On Circuits and Systems, June, 1997, Hong-Kong.

[7] http://projects.ilt.columbia.edu/edviz/edviz.html

Figures 2a, 2b, 2c