



Fast Multiresolution Image Querying

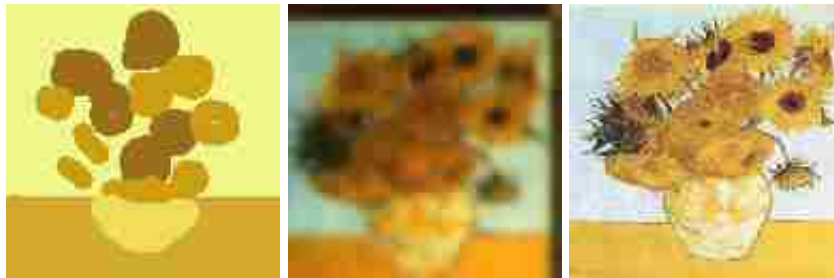
Overview

We are exploring a strategy for searching through an image database, in which the query is expressed either as a low-resolution image from a scanner or video camera, or as a rough sketch painted by the user. Our searching algorithm makes use of multiresolution (wavelet) decompositions of the query and database images. The method is both effective and fast.

If you have an Adobe Acrobat reader or plug-in, [click here](#) to read the paper.

To acquire a Postscript version of the paper by ftp [click here](#). It appears in Proceedings of SIGGRAPH 1995.

Query by Content



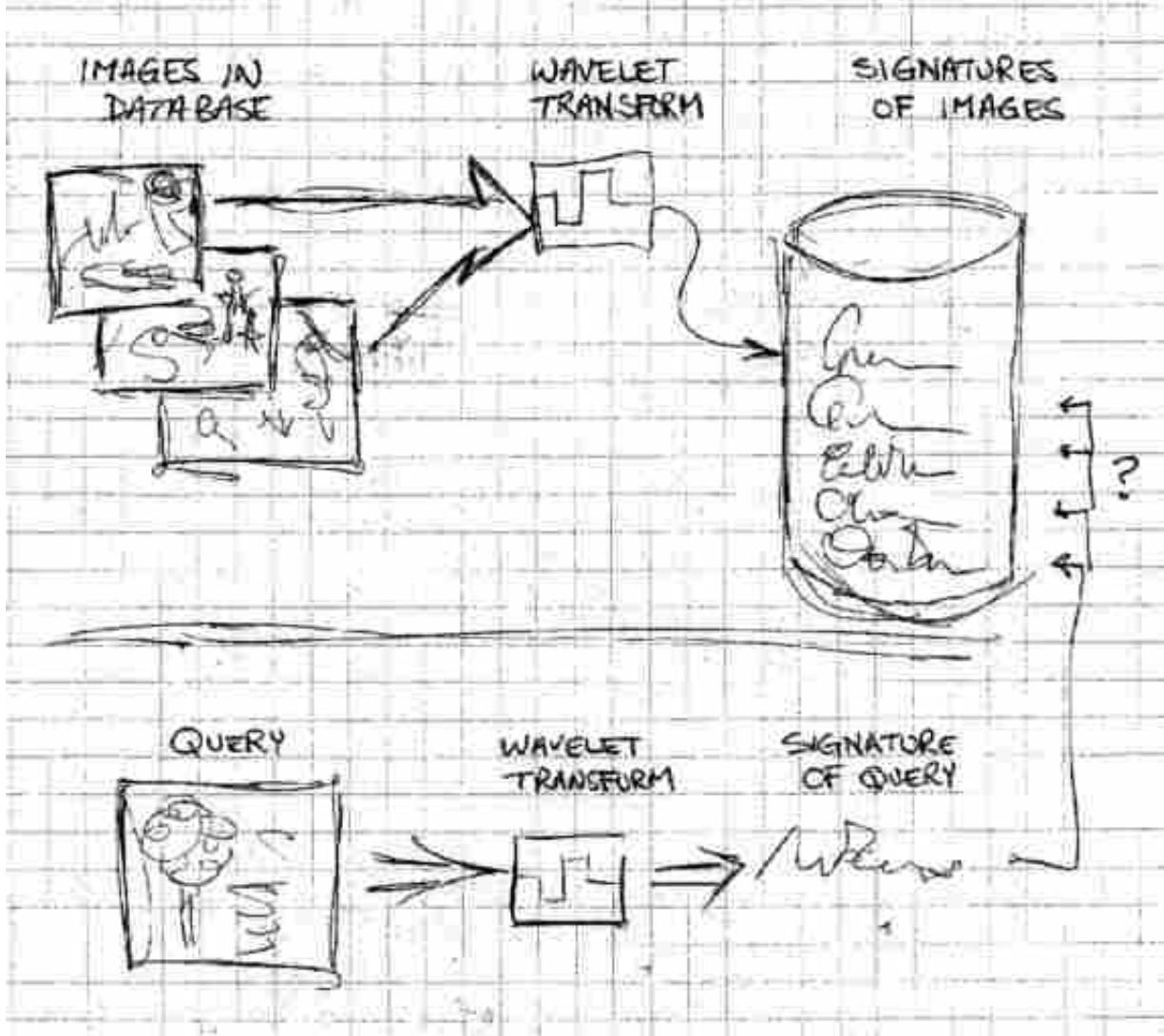
Painted

Scanned

Target

In this paradigm, the user expresses a query to the database either by painting a crude picture or by showing an example of the image to a video camera or scanner. Our algorithm considers the basic shape and color information of the query when looking through the database for potential matches. Several factors make this matching process difficult. The query image is typically different from the target image, so the retrieval method must allow for some distortions. If the query is scanned, it may suffer artifacts such as color shift, poor resolution, dithering effects, and misregistration. If the query is painted, it is limited by perceptual error in both shape and color, as well as by the artistic prowess and patience of the user. Furthermore, we would like the retrieval to be fast enough to handle databases with thousands of images at interactive rates.

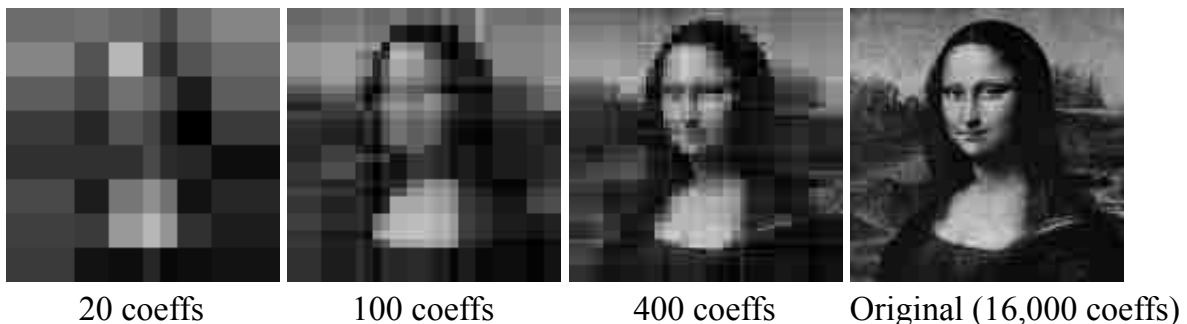
How it works...



In a preprocessing step, we perform a wavelet transform on every image in the database. By collecting just the few largest coefficients from this transform, we distill a small "signature" for each of these images. We save these signatures, organized in such a way that it is easy (and fast) to compare them all to a new signature.

Then, when the user submits a query, we perform the wavelet transform to produce a signature for the query image. This new signature is compared to the signatures of the database images, and the best matches are retrieved for the user.

The Wavelet Transform



The wavelet transform is a tool that has emerged from the mathematics community in last decade or so for analyzing functions at different levels of detail. It is similar to the Fourier transform, but encodes both frequency *and* spacial information.

Much of the work in wavelet research relating to images has been in the area of image compression. By saving the few largest wavelet coefficients for an image (and throwing away all of the smaller coefficients), it is possible to recover a fairly accurate representation of the image. This property may be exploited for compression. For example, the image shown above that incorporates 400 coefficients would require about 3% as much disk space as the original image.

In our application, we take the wavelet transform and keep just a few (20) coefficients for each color channel and distill from them a very small "signature" for each image. Because the signature is so small, it permits very fast searching in the database. Nonetheless, it has proven to be an effective method for discriminating images.

The Application



Our application allows the user to paint a query image in a Macpaint- or Photoshop-like way. Alternately, the user may load a query image from disk, perhaps one that was scanned or captured by a video camera. When the user presses a "match" button, the application retrieves the 20 most similar images (by our wavelet-based metric) from the database. These are displayed as small thumbnails in the upper right part of the window. If one of these was the desired image, the user simply clicks on that thumbnail to retrieve the full-sized image. Otherwise, if the desired image doesn't appear among the thumbnails, the user may modify the query and "match" again.

Because the queries may be matched against the database very quickly, we have implemented an "interactive" mode for the application. In this mode, the application re-evaluates the painted query in the database as each stroke is drawn. Every time the user pauses from painting for a moment, the application retrieves a new set of thumbnails. Median time to retrieve the target image appears to be around twenty seconds for a database of 1000 images.

Results

How well does it work?

- We have compiled a database of more than 1000 images.
- For painted queries, 51 of 87 test queries retrieved the correct image as a "best match".
- For scanned queries, 66 of 89 test queries retrieved the correct image as a "best match".

How fast is it?

- It takes 0.19 seconds for a database of 1000 images.
- It takes 0.44 second for a database of 10000 images.
- The algorithm is linear in the size of the database.

For a more complete description of the results, please see [the paper](#).

Or, if you have an Adobe Acrobat reader or plug-in, [click here](#) to read the paper.

What's next?

The World Wide Web version, of course. We have collected about 20,000 URLs for images which are available on the Web, thanks to [Brian Pinkerton](#) and [WebCrawler](#).

With this larger database, the user may search for images using the same painting application. But when the user clicks on a thumbnail, the full-size image is retrieved over the Web. We have yet to perform tests to see how effective our method is at retrieving specific images from this larger database.

The Authors



Chuck
[Jacobs](#)



Adam
[Finkelstein](#)



David
[Salesin](#)