

Active Image Database Management

Jau-Yuen Chen
4.3.2000

1. Application

This document is applied to the active image database management.

2. Concept

The goal is to provide user with a systematic navigation tool for image database access. In this system, user can use an example image to retrieve similar images or use a 3-D location to retrieve a snapshot of the similarity pyramid of the database.

To use this system, user first creates a feature database by extracting the features from the images in the database, and then organizes the images into a hierarchical structure by clustering the individual images into a similarity pyramid.

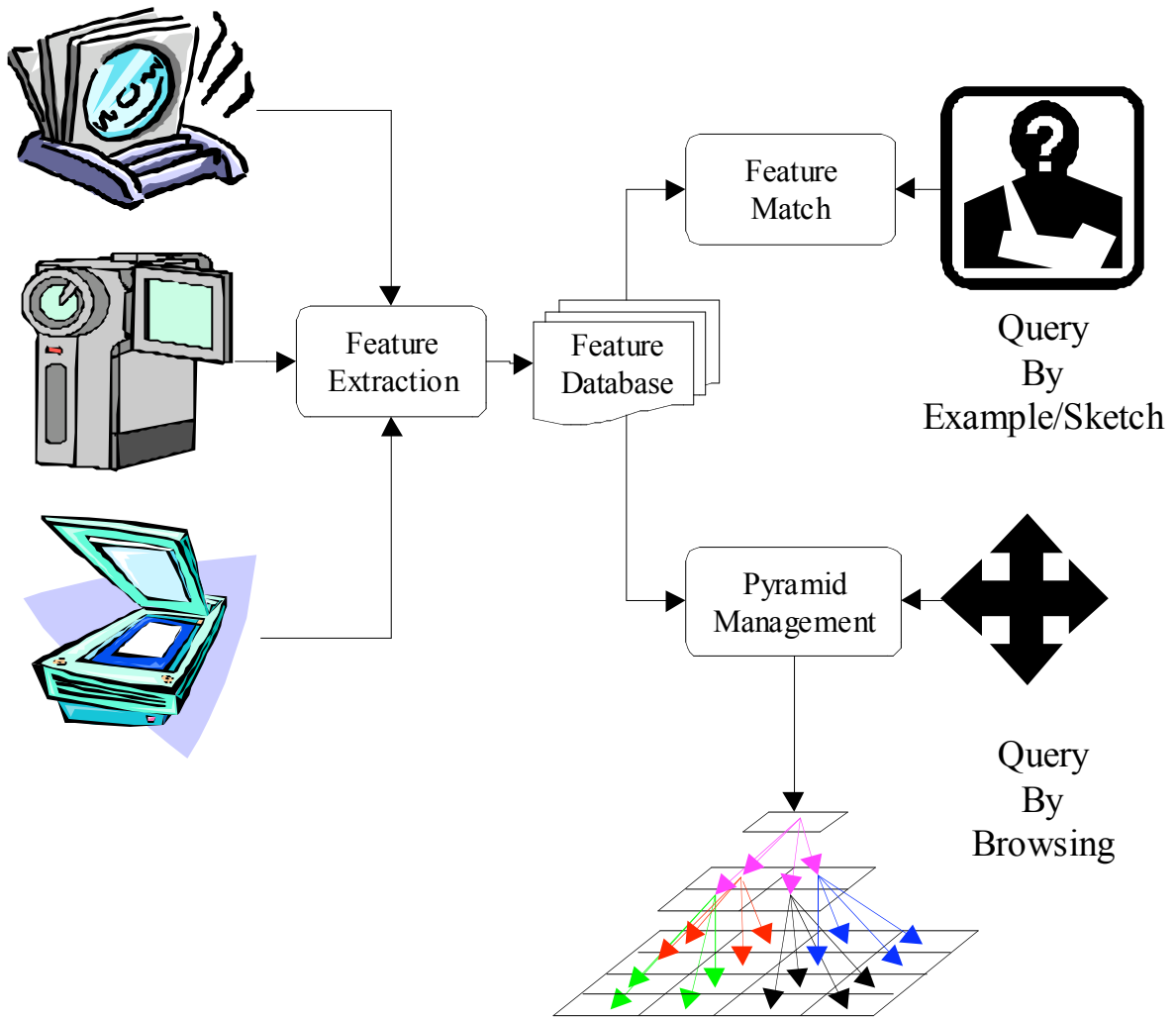


Figure 1: Image Database Management Flowchart

2.1 Image Feature

An image feature is defined by the color, edge, and texture histograms of the raw images. The color feature are formed by independently histogramming the three components of CIE $L^*a^*b^*$ color space. The texture features are formed by histogramming the magnitude of the gradient of each color component. The edge features are formed by thresholding the edge gradient and then computing the angle of the edge for all points that exceed the threshold. These values are then histogrammed.

2.2 Similarity Pyramid

Based on the image features, we organize the images into a hierarchical data structure, which we call a similarity pyramid. The similarity pyramid organizes the image in the databases into a three dimensional pyramid structure to allow user to browse the database in a logic way. Each level of the similarity pyramid contains clusters of similar images organized on a 2-D grid. As users move up this pyramid, each image is an icon that represents a cluster of similar images. As users move down the pyramid, the clusters become smaller, with the bottom level of the pyramid containing individual images which are organized so that similar images are nearby each other. In addition, users can pan across at a single level of the pyramid to see images or image clusters that are similar. Therefore, at coarse levels of pyramid, user can get a more global perspective of the database whereas at finer levels, users see more detail variation.

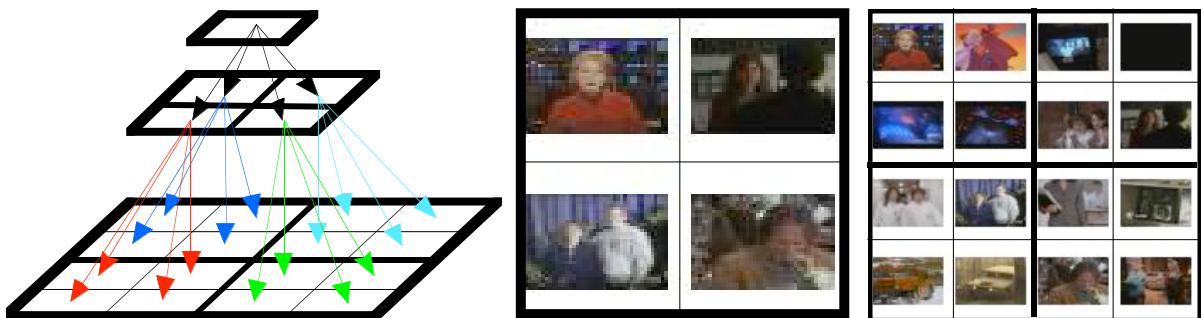


Figure 2: Similarity Pyramid

3. User Guide

The system uses a scripting language called Tcl/Tk, which can be freely downloaded from

<http://www.scriptics.com/software/download.htm>.

The recommended version is Tcl/Tk 8.0.

For easy access, the programs are controlled by “browser.tcl”, a Tcl/Tk user interface program. To execute the browser, simply double click “browser.tcl” or run “wish80 browser.tcl” (not recommend). A window with 64 (8x8) images on the left and control panel on the right should appear, otherwise, something is wrong.

Below is the browsing interface. Initially, the system shows the 8x8 level of pyramid on the left. By clicking on an image, user can zoom in to another 8x8 array of image at the next finer level of the pyramid. The control button also allows users to pan left, right, up, and down at the same level of the pyramid.

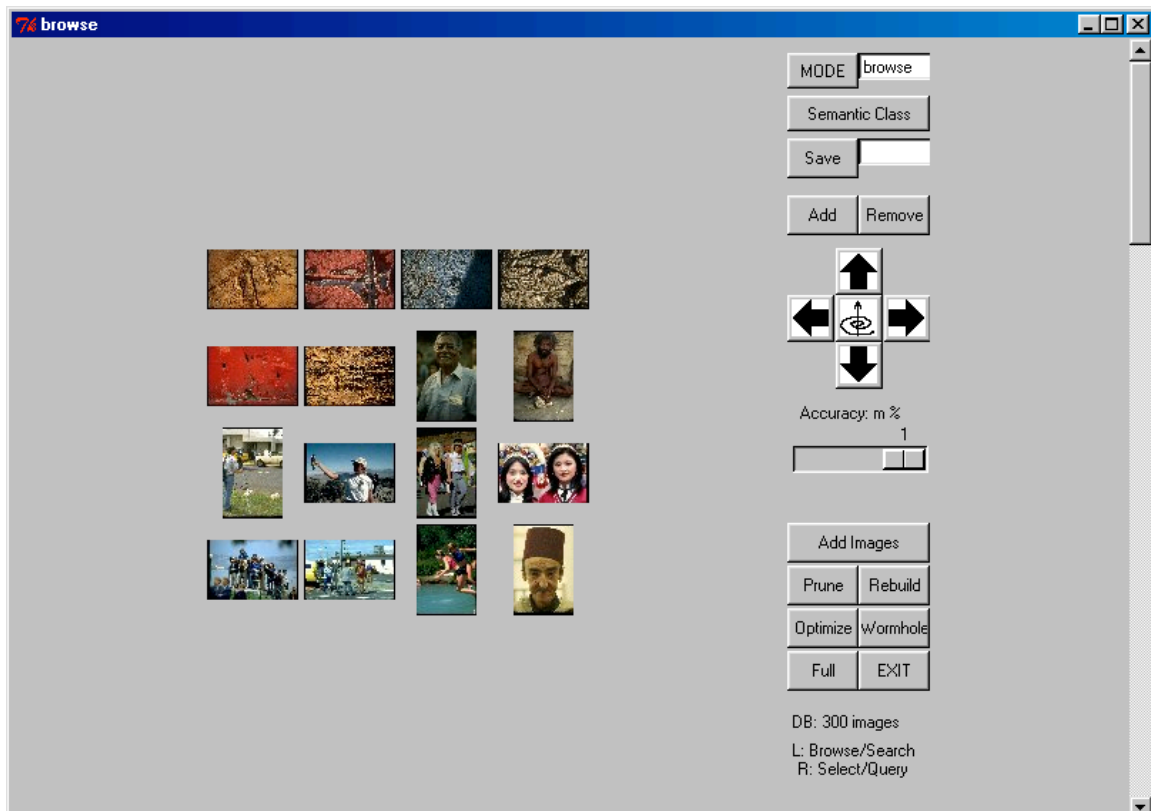


Figure 3: Browser Interface

3.1 Mouse

Clicking the left button of mouse on an image, the system will go down one level in the similarity pyramid. Clicking the right button of mouse on an image, user can select/de-select an image for later use.

3.2 Buttons

The first button of control panel is program mode: browse and search. The current mode is displayed next to the button.

The second part of control panel is the semantic class control. From the "Semantic Class" menu, user can create a new class by selecting "New", or retrieve a saved class by selecting the associated class name. After creating new class, the "classname" entry will be enabled to allow user to enter a name. The "Save" button allows user to save the relevance set as a semantic class associated with the name shown in the "classname" entry.

To add images to relevance set, simply click the right button of mouse on any image (a second click will de-select it) and then click the "Add" button to add it to the relevance set. To remove images from relevance set, simply click the right button of mouse on any image (a second click will de-select it) and then click the "Remove" button to remove it from the relevance set.

The next part is browser control, including zoom out ("Top"), which moves view window up one level in the pyramid structure, and pan left ("Left"), pan right ("Right"), pan up ("Up"), and pan down ("Down") which move the view port in corresponding direction at step of 4 images. To zoom in the view window, simply click the left button of mouse on any image. Currently, there is no indication whether the image is leaf in the pyramid.

The next item is a scale, which determines the threshold of pruning. The threshold is determined by an ordered statistic method. The scale shows the predicted accuracy, the higher the accuracy, the larger the pruned database. Currently, there is a 1000 images upper limit.

The next buttons is "Add Images", with which users can add images to the database by browse through the directory structure.

The next set of buttons are "Prune", "Rebuild", "Optimize", "Wormhole", and "Full" browse. By selecting some example images, user can prune the database, and build a subset pyramid.

The "Prune" button will prune the database to according the threshold determined by the maximum of nearest neighbor distance among relevance set and user selected

coefficient. The result will be shown in the form of a similarity pyramid in standard similarity metric.

The “Rebuild” button will rebuild the similarity pyramid based on the standard weighting which weights color, texture, edge equally. The “Optimize” button will rebuild the similarity pyramid based on the weighting, which best discriminates the relevance set from a set of randomly selected images. The “Wormhole” button will rebuild the similarity pyramid based on the wormhole distance described in the paper.

After pruning, search, browser control, rebuild, optimize, and wormhole will work on the subset database until user use the “Full” button to go back the original database. The “Full” button will switch the environment to the original database.

The last button is “exit” which will close the main program “active.exe”.

The bottom part is reserved for debugging message. The first one is the initial size of database. The second one is message or the filename of selected image.

3.3 Package Content

The “work” subdirectory contains:

browser.tcl:	the tcl/tk user interface of the image database system
active.exe:	the real program which accept input from browser.tcl
getfile.tcl:	the tcl/tk subroutine for adding images to database
isimage.exe:	to decide if file is image, used in getfile.tcl
db_folderSIMPLE	the database files automatic generated by browser.tcl
extract.ppm:	temporary file for display, generated by active.exe.
list_of_images:	temporary file for adding images, generated by active.exe.

4. Function Description

4.1 Add Image

Use “Add Image” command to browser CDROM / photo directory and to add a list of images to database.

System extracts feature after user selecting a list of images.

After extracting feature, new images will be added to the similarity pyramid for browsing.

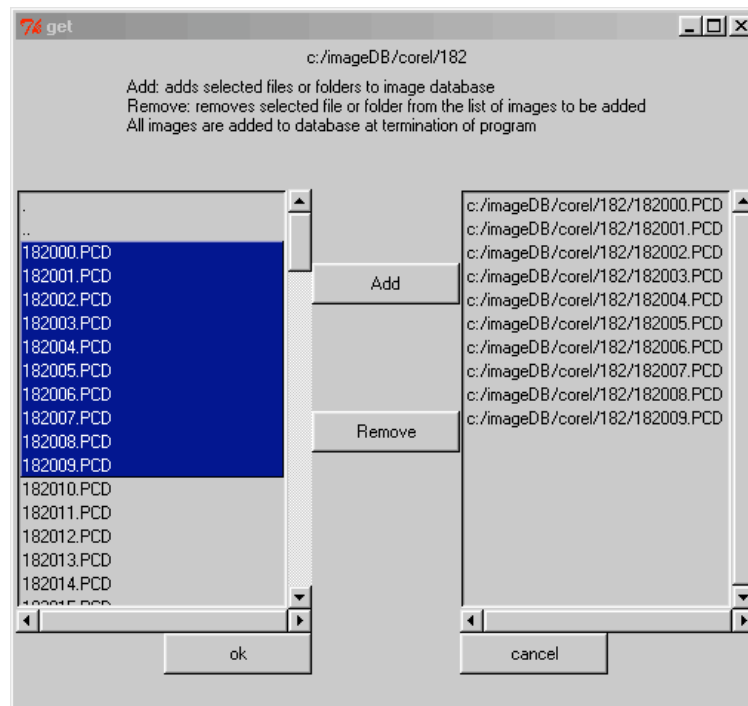


Figure 4: Interface for adding new images

4.2 Query by Search

The main stream of image database research is query-by-example approach. The problem occurs when user needs to find the best K matches to a query image. In this system, users can click on an image, which is most similar to the desired image. After the system returns a set of similar images, users can use these images to do another query.



Query Image



4 best matches

4.3 Query by Browsing

However, there are some limits about the query-by-example approach. To use this approach, user is asked to give an example image. Moreover, the search cycle usually starts with an example image. After the system returns a set of similar images, most likely user will use these images to do another query. Unfortunately, most likely it will return a similar set of images, therefore, becomes trapped among a small group of undesired images. To break this trap, browsing environment can provide an alternative approach.

Following is an example of how we can navigate through a similarity pyramid. Initially, we see a global view of the database. Starting from top level, we choose an image of interest to zoom in to see more similar images at the next finer level of the pyramid. We can continue zooming into smaller clusters until we find an image of interest. In case of entering into a group of not quite similar images, we can pan across the pyramid at any level to see other type of images. Or we can zoom out to make panning more efficiently.

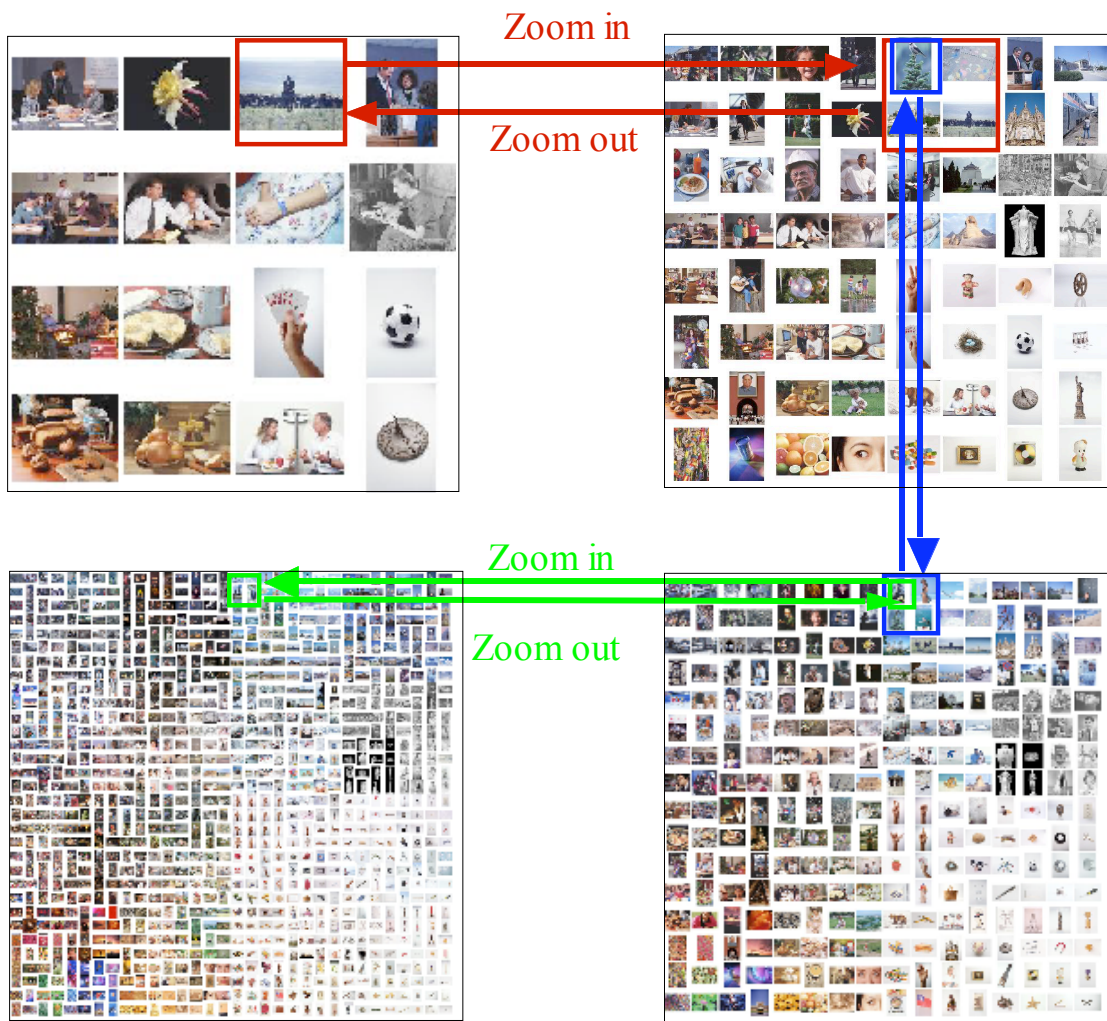


Figure 5: Example of navigating through similarity pyramid

4.4 Pruning

The objective of pruning is to remove images from the database that are not likely to be of interest to the user while retaining all or most potentially desirable images. This is useful since the reduced size of the pruned database makes it easier and more effective to browse.

While traditional query methods attempt to find images that are likely to be of interest, pruning retains all images, which are of possible interest, but at the expense of retaining many questionable images. Intuitively, pruning attempts to achieve high recall, but at the expense of precision; whereas traditional queries tend to emphasize precision over recall. The reduced precision of pruning is acceptable because the similarity pyramid structure allows the user to efficiently browse the resulting images.

4.5 Reorganizing

In addition to pruning, relevance feedback can be used to reorganize the database so the images of interest are grouped together. As with pruning, this reorganization makes the browsing process more efficient. Since the desired images often form disjoint clusters that are spread through the feature space, they will often be spread into disjoint groupings in the similarity pyramid. The reorganization step helps to move these desired images near one another in the similarity pyramid structure. While reorganization may be performed at any point in the browsing process, it is typically performed after pruning.

In this system, we employ two different methods for reorganizing the similarity pyramid. Both methods depend on computation of a new dissimilarity metric based on the images in the relevance set. One method uses linear transformation to optimize the organization of the similarity pyramid. The other method uses nonlinear transformations of the dissimilarity metric to nonlinearly warp the feature space so that images in or near the relevance set are near each other. This distorted distance function, which we call wormhole distance is then used to rebuild the similarity pyramid so that images similar to the relevance set are grouped together.