1993

Retrieval from an image knowledge base

James Janicki

Rochester Institute of Technology

Computer Science Department

Retrieval from an

Image Knowledge base

by
James H. Janicki
16 Hadley Court
Pittsford, NY 14534

A thesis, submitted to

The Faculty of the Computer Science Department

in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science.

February 23, 1993

Approved by:

Professor Walter A. Wolf

Professor Peter G. Anderson

Professor John A. Biles

Title of Thesis:

Retrieval from an Image Knowledge base

I, James H. Janicki, hereby deny permission to the Wallace Memorial Library of Rochester Institute of Technology (RIT), to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

February, 23, 1993

TABLE OF CONTENTS

1. ABSTRACT

With advances in computer technology, images and image databases are becoming increasingly important. Retrievals of images in current image database systems have been designed using keyword searches. These carefully designed and handcrafted systems are very efficient given the application domain they are built for. Unfortunately, they are not adaptable to other domains, not expandable for other uses of the existing information and are not very forgiving to their users. The appearance of full-text search provides for a more general search given textual documents. However, pictorial images contain a vast amount of information that is difficult to catalogue in a general way. Further this classification needs to be dynamic providing for flexible searching capability. The searching should allow for more than a preprogrammed set of search parameters, as exact searches make the image database quite useless for a search that was not designed into the original database. Further the incorporation of knowledge along with the images is difficult.

Development of an image knowledge base along with content-based retrieval techniques is the focus of this thesis. Using an artificial intelligence technique called case-based reasoning, images can be retrieved with a degree of flexibility. Each image would be classified by user entered attributes about the image called descriptors. These descriptors would also have a "degree-of-importance" parameter. This parameter would indicate the relative importance or certainty of that descriptor. These descriptors are collected as the "case" for the image and stored in "frames" Each image can vary as to the amount of attribute information they contain.

Retrieval of an image from the knowledge base begins with the entry of new descriptors for the desired image. Along with the descriptors are the degree-of-importance parameter. The degree-of-importance would indicate the requirement for the desired image to match that descriptor. Again, a variable number of descriptors can be entered. After all criteria are entered, the system will search for cases that have any level of matching. The system will use the degree-of-importance both in the knowledge base about the candidate image(s) and the degree-of-importance on the search criteria to order the images. The ordering process will use weighted summations to present a relatively small list of candidate images.

To demonstrate and validate the concepts outlined, a prototype of the system has been developed. This prototype includes the primary architectural components of a potentially real product. Architectural areas addressed are: the storage of the knowledge, storage and access to a large number of high-resolution images, means of searching or interrogating the knowledge base, and the actual display of images. The prototype is called the "Smart Photo Album" It is an electronic filing system for 35mm pictures taken by the average photographer on up to the photo-journalist. It allows for multiple ways of indexing the

pictures of any subject matter. Retrieval from the knowledge base provides relative matches to the given search criteria. Although this application is relatively simple, the basis of the system can be easily extended to include a more sophisticated knowledge base and reasoning process as, for example, would be used for a medical diagnostic application in the field of dermatology.

1.1. Acknowledgements

I would like to acknowledge the support received from the RIT thesis committee, in particular the energy and enthusiasm received from Walter Wolf. I would also like to thank my family for being understanding and supportive of this effort.

1.2. Classification Codes

Using the computing classification codes by the ACM, the primary classification code is H.3.3 - Information Search & Retrieval. Secondary codes are: I.2.4. I.4.0. and H.3.1.

2. Introduction and Background

The economic climate in today's world forces businesses to rely on computers for their competitive advantage. With the ever increasing capabilities and continued decreasing cost of today's PC, they are commonplace in the business community. PCs contain high-resolution displays, large storage capacities and fast processors. They provide sophisticated applications that non-programmers require to run their business. Current PCs allow for displaying of graphics information in addition to text. Graphics and images play an increasing role in the communication of business related information. The expression "a picture is worth a thousand words" indicates how images represent a vast amount of information that is extremely difficult to describe in textual form. The trend in computer technology is continually focused on graphics and images by uses such as

- graphical user interfaces (i.e., X-Windows, Microsoft Windows);
- the incorporation of graphics into standard documents via sophisticated word processors and desktop publishers;
- draw, paint, image and pixel manipulation software;
- presentation and business graphics software; and
- the increasing popularity of multimedia and full motion video.

PCs display graphical information and images on high-resolution color monitors. A minimum of 512 X 480 pixel resolution is required to closely examine a still image[1] Current raster devices (VGA, SuperVGA, XGA) exceed this minimum and have resolutions that approach the 1 million pixel range (1280 X 1024 pixels). Each color pixel can be represented by 8, 9, 16 or 24 bits; the most common being RGB 5:5:5 for 16 bits and RGB 8:8:8 for 24 bits. Typical color monitor sizes start at 14 inches and

proceed to as large as 21 inches. This equipment provides for a completely new interface and different applications to its users.

Further advances in mass storage technology have developed disk drives with hundreds of MBytes of storage (approaching GByte range) in a relatively small package at a very affordable price. This increase in storage has made non-textual forms of information available more readily to users (i.e., 1 MByte of information can provide a single VGA-resolution image or approximately 6 seconds of audio). A common use for optical storage has been the recording of audio information on CD-ROM. This optical format provides for 30 to 60 minutes of digitized music that is very portable and highly reliable. In a PC, a CD-ROM can contain a combination of text, binary, graphical, image, and audio information. Use of compression techniques can further extend the useful storage capacity. Use of mechanical jukeboxes provide for an array of CD-ROM's with automatic access. Thus, optical storage technology provides the convenient mechanism and vast storage required to create an image knowledge base.

An emerging technology/product using CD-ROM for DCI is PhotoCD[2] This technology provides for the accurate recording of over one hundred 35mm photographic pictures on a standard CD-ROM (typical capacity of 650 MBytes). The CD-ROM contains several resolutions for each image starting from a low resolution of 64 x 96 pixels for icons to its highest resolution of 2048 x 3072 pixels. Images are compressed to further increase the amount of information stored on the CD-ROM. The PhotoCD is readable by any CD-ROM XA drive. It allows any user to create a custom CD-ROM with their unique pictures by using a 35mm camera. Thus, PhotoCD can provide a large area of imaging application to the PC desktop. However, given its recent market introduction, the current software for the product is limited to a simple PhotoCD Access Developer Toolkit. This toolkit provides file system type access to images on the PhotoCD. The software capability to "index" the images is not currently available. For consumer purposes, using a "contact sheet" (containing picons and image numbers that is printed and included in the jewel case when the images are recorded to the CD-ROM) allows for retrieval of the desired image. The user simply provides the desired image number printed on the contact sheet for a higher resolution image to be displayed.

Given current processing speeds and monitor capability on the PC, displaying several picons is feasible. With these picons, the user can "browse" to determine which images are of interest and select or mark those desiring a more detailed study. Those marked images (hopefully a smaller set) can be displayed in a higher resolution one image at a time. The display time for these high resolution images will be dependent on the hardware and software involved and the resolution desired on the display device (which could be different than the stored resolution). After further examination, the user can determine which image most closely matches their needs. Given a large set of images, this process can be very time consuming. This would make it unacceptable for today's business application. Thus, a more convenient

and accurate means of electronically indexing of these images is required. The most sophisticated form of retrieval would be from an image knowledge base.

2.1. Problem Statement

Although browsing through images using a PC and a PhotoCD can be fun and interesting for a consumer, browsing is not a technique that is useful in the business community given a large database of images. Further, picons cannot provide enough information for a reasonable selection process since they are image dependent (i.e., large panoramic view versus a close-up of a specific object, bright outdoor lighting versus night time). Further, there is no support information available to aide the selection process. Having some form of index information would allow for the images to be organized. This organization would provide for a quick search through the image database to provide a prompt response of a single or small set of images given some specific set of criteria. This prompt retrieval is critical in a business application. This is evidenced in the RightPages Image-Based Electronic Library[3] where displaying of front covers of recent journals is inadequate to determine the journal articles of interest. The system uses the table of contents from each journal along with a user's profile to highlight journal articles of interest to the user. Thus, it is essential that a set of index information or knowledge needs to be incorporated with each image in the database.

To incorporate information in the image database, a designer would determine the appropriate set of attributes to be recorded. This information would be application dependent. It would be organized by the type of retrievals expecting to be preformed. Traditional methods would suggest that this textual information would be stored in a relational database with an application dependent set of columns or tuples representing the attributes. The end-user of the system would go through a data entry process for each image in the database. The data entry activity would require a set of forms or screens that would allow the user to enter the specific type of information desired given the application. The forms would also perform some level of validation on the entered data (i.e., were digits entered for a numeric field, was a valid date entered for a date field) and potentially a conversion before storing the information in the database (i.e., converting an ASCII string representing the entered date such as "November 11, 1992" into an integer value). The net effect is that we have two databases in the system, one for the images and one for the textual information that describes the images.

Difficulties in building the textual database include the designer's ability to incorporate or even appreciate all of the attributes necessary for the application domain of interest. It would include the programmer's ability in developing sufficient audit and validation routines to insure data is in the proper format. The data entry person would require training on how the system was built and operates. They would also need application domain knowledge to be able to interpret the image and accurately type the data in the proper

field. Potential areas of concern include: Can the designer anticipate all the necessary fields required in the textual database to satisfy all desired retrievals? What is the programmer's ability to eliminate all data entry errors through comprehensive validation? What is the data entry person's ability to properly key in the needed information without mistake? How to determine how closely a picture's attributes match the desired selection criteria? Even after we have built this textual database and attempted to minimize potential problems, still other questions are raised such as: Can the system be extended to accommodate new retrievals? Can the retrievals provide the images desired given partial information for retrieval? Can the appropriate images be retrieved given some relatively small amount of data entry error? What is the probability of having the desired image not be retrieved given the search parameters specified?

To minimize all of the concerns raised in the traditional approach, a number of system requirements are desired. First, the incorporation of knowledge into the image database should be as general as possible. It should only be as domain or application specific as necessary to provide for efficient retrieval. It should minimize the need for a rigid structure and should maximize the user's ability to specify that knowledge (avoid limiting the attributes of the image to a small predefined set). Further, the knowledge should be understandable to all users and not just the designer, the data entry person or the domain expert. The structure of the knowledge is important in being able to provide for an efficient retrieval. The knowledge should be easily extended to include additional attributes of each image. The system should perform relative searches rather than exact matches. The system should act as a whole where the images and knowledge are closely integrated.

The goal of this thesis was to develop an image knowledge base that would grow as the number of images increased. Each image would have a set of descriptors describing its content and/or use as it is added. These descriptors would be relatively simple sentences making it easy to add images. These descriptors provide for factual information about the contents of the image as well as subjective information. The descriptors would contain relationships. In addition to the descriptors, the user can specify a degree-of-importance with each one differentiating major from minor focus.

The retrieval of an image from the knowledge base would require input of a set of descriptors with their corresponding degree-of-importance in this search. The set of descriptors would be compared against the knowledge base to determine the relative matches. The set of images matching any of the search descriptors would be further analyzed by the number of descriptors that match, the degree-of-importance associated with each search description and the degree-of-importance with each image in the matching set. The retrieval would provide an ordered list of images based upon each image's ability to match the specified search criteria. The results of the search would be presented in a manner that would allow for access to the full-resolution image, a picon, the textual knowledge, and the explanation within the

reasoning process that determined the order. It could be possible to extend the results to include some small amount of audio information.

Inherent in any large knowledge base is the requirement for an extensive amount of data entry. In knowledge bases developed by an expert, it is usually the expert or developer that performs the data entry activity rather than the end-user. Since the Smart Photo Album is so general an application, it would be impossible to find an expert that would satisfy all consumers. Also the pictures incorporated into the image knowledge base are user dependent. Thus the end-user of the Smart Photo Album will be both the data entry person and the retrieval person of the knowledge base.

2.2. Previous Work

2.2.1. Image Databases

Previous work in image databases has focused on both graphical and image information. Graphical information is typically represented as vector information (i.e., CAD drawings). Images are represented in raster format where each pixel is represented by a value; all pixels that make up the image are fully represented. A number of approaches will be presented that indicate the manner in which image databases are currently being developed. Some of them deal with vector data which is much farther removed from the current scope. However, a brief look at their architecture will be examined for any parallels.

In graphical databases, retrieval is typically based on image content. Given application domain knowledge, objects within the image are recognized using different object interpretations, degree of recognition, and their position in the image space. Images can be composed of complex objects[4] These complex objects are decomposed into simpler graphical shapes that can be more easily assimilated. Classification is limited to a well defined set of basic objects. The main disadvantage of these type of graphically oriented databases is the need to translate the raster information of the image into vector information. If we are unable to translate the raster information into a recognizable object (defined by a set of vectors), then the classification process fails.

Another retrieval approach for graphical databases has focused on shape similarity retrieval[5]. An object is represented in terms of its local structural features. Each structural feature is represented as a point in a multidimensional space. Indexing is performed using a multidimensional point access method. The technique can handle occluded and touching objects.

Other research in image databases deals with computer vision and machine intelligence where the focus is on understanding the actual content of the images. With this type of understanding, a robot could

navigate through a room with a number of obstacles within the room. Another common application is the analysis of aerial photographs desiring to identify specific objects such as roads, houses, etc. SIGMA[6] is such a system. The architecture consists of four main modules: the "Geometric Reasoning Expert" uses scene domain knowledge for spatial reasoning among objects, the "Model Selection Expert" which reasons about appearances of objects in the image based on knowledge about the mapping, the "Low-Level Vision Expert" which performs image segmentation using image domain knowledge and the "Question and Answer Module" The first three modules are analysis modules to construct the description of the scene. The last module realizes the interactive information retrieval facility to examine the content of the constructed description.

Unfortunately, the decomposition and recognition process for any of these systems is limited to specific objects within its domain. These objects cannot be generalized. Given the Smart Photo Album, it would be impossible to create such a domain knowledge. Further, the query process entails the user describing the shape of the object and/or location information within the image. This process would only make sense in our application if the user had remembered the approximate content of the desired image. Also, it would be very difficult to provide a user-friendly query language (it would require the user to draw the shape desired in order to begin searching). At best this approach might provide a finer level of search after a gross search was performed with some other mechanism.

Another approach that has been primarily used in the art-history field is Iconclass[7] Each image is given an alphanumeric code that represents its position into a large, well-defined, hierarchical catalogue. This code represents the subject of an image and/or the elements in the image (depending on depth). As an example, an image of a mountain landscape would have the following classification (only showing a limited part of the classification tree):

1. Supernatural, God, Religion

2. Nature > 21 The four elements

22 Natural phenomena

23 Time

24 The heavens

25 Earth > A. Maps, Atlases

:

H. Landscapes > 25H1. (Landscapes)

:

25H11. Mountains

3. Human Beings
4. Society, Civilization, Culture
5. Abstract Ideas and Concepts
6. History
7. Bible
8. Non-Classical Myths, Tales and Legends
9. Classical Mythology and Religion

Although this scheme is quite efficient in the art-history domain, it requires all users to be intimately familiar with the large set of classes to perform a successful retrieval. Further, providing a single classification can be limiting and error prone. This would be the same situation as trying to use the Library of Congress or Dewey Decimal Systems found in libraries. The librarian understands the notation but users require a catalogue to be able to determine the number and location of a desired book.

Retrieval of images from an image database using keyword search is quite common. A number of chosen words are used to describe the image, typically using terminology in the application's domain. These words are hierarchically populated into a relational database that provide for quick retrieval (refer to previous section). However, images contain a vast amount of information that is difficult to capture in a relatively small number of database index fields. These indexes tend to be static and rigid making anything but exact searches possible. Exact searches make the image database quite useless for a search that was not designed into the original database.

The shortcomings of keyword searches have led to full-text searches or content based retrieval systems[8]. Here images contain text that is recognized by an OCR device. The text is used for indexing purposes without the need for a user to perform any data entry (other than initial OCR regions). Retrieval is performed by the user specifying a set of words describing the images to be retrieved. The OCRed content

is searched using exact word matches for each search word and proximity ranges between each word. No predefined organization of the text is required.

2.2.2. Case-Based Reasoning

Rule-based expert systems have been a major focus in artificial intelligence research and development. The basic unit of knowledge is the rule which constitutes a conditional test and resulting action pair (typically modeled by "if <condition> then <action>). Shortcomings of rule-based expert systems focus on knowledge acquisition, lack of memory, and lack of robustness. To build an expert system, an expert needs to be identified who can articulate his knowledge and experiences in such a fashion as to impart it within the system (either by himself or with the aide of a programmer). It soon becomes an arduous task to implement hundreds of rules that appear to the expert as common knowledge (which might be difficult to explain or so commonplace that the expert forgets that he performs them). Lack of memory implies that the expert system would perform the exact same set of operations given the same problem in succession; it did not learn or remember from the first problem it solved. Expert systems lack robustness in not being able to respond to a problem if none of the rules apply. Worse would be an inappropriate response given lack of rules or domain knowledge.

Case-based systems are an alternative to rule-based systems. The knowledge in CBR is found in the "case" or example which represents the complete knowledge and experience of a real-world problem. The system performs reasoning by comparing a problem being input to the set of cases to find the closest match. Applications that have found success using CBR are the help desk and legal reasoning[9] in domains such as tax law. CBR systems tend to be easier to build and maintain than rule-based systems.

To demonstrate the importance being given to CBR, a significant undertaking was the large scale implementation by the NEC Corporation[10]. A corporate-wide case-based system called SQUAD (for the Software Quality Control Advisor) was developed to support their software quality control activities. The data acquisition process started without a completely defined case format given the lack of domain experts who have a well-organized idea of all the factors involved. Goals for the system included: an easy way to share experiences among 150,000 employees; a flexible and robust system that could accommodate the changing nature of a corporate structure; and a system that can deal with missing and inaccurate information which arise in real-world installation. The net effect of SQUAD are quality and productivity improvements over a hundred million dollars a year.

A case-based teaching system called CreANIMate uses story telling to help students learn about animals (physical features and survival)[11]. Multimedia is incorporated into the system by use of video clips depicting animals in the wild. The system through question and answer dialogue provides suggestions to

the student about why animals have certain features and offers a video clip for evidence. Reminding heuristics are employed to retrieve stories to achieve specific pedagogical objectives. Specific strategies include: example remindings - to provide examples, similarity-based remindings - to assist in generalization, and expectation violation remindings - to challenge the student's expectations.

Retrieval from CBR systems fall into three categories: inductive, nearest neighbor and template retrievals[12] Inductive retrieval involves traversing a hierarchical knowledge structure in the form of a decision tree looking for a leaf node providing a similar case to the user. The inductive method requires a predefined set of indexes of relevant features to build the decision tree. Without the index, the hierarchical structure cannot be created.

The nearest neighbor retrieval performs a feature-by-feature comparison of the input case against all cases in the library. Using a weighted vector a feature score is generated with the highest score being the closest match. Typically a score of one hundred (100) indicates a perfect match where a score of zero (0) indicates no match at all. Each feature will effect the overall score of the case, based on its weight.

Template retrievals are used for directed searches of a case library when you know exactly what you want. A template is defined for matching on specific characteristics or keywords as in an SQL query of a relational database.

2.3. Theoretical and Conceptual Development

The knowledge base being built has specific tradeoffs. Ideally the system should provide the most expressive means to the developer to provide descriptors of the images. This implies no limitations on the vocabulary and sentence structures that can be entered by the user. However, this creates potentially an intractable inference problem or at best a poor performing system. Thus, an alternate implementation would be to allow the developer to enter statements in a general, structured representation language, which the system would translate into a restricted, internal language or representation that allows for efficient inference. If an exact translation is impossible; a best approximation will be used to speed up inference without giving up correctness or completeness.

The image knowledge base was built from a set of descriptors and importance factors. The developer provides a set of simple sentences as descriptors for each image. Each sentence is of varying length. Sample sentences are: "Girl wearing red hat", "Sailing race on sunny day", and "Vacation in Colorado". The set of sentences will be of varying number. The maximum allowable number of sentences will be a relatively large number with an absolute upper-bound. Along with each descriptor is a degree-of-importance. This value will be a simple selection of three values: high, medium, and low.

The system parses each sentence or descriptor provided by the developer. Using a context-free grammar, the parser determines if the sentence is in proper form. A dictionary or lexicon is used to perform a lookup for each word in the sentence. The dictionary validates proper spelling of words. It also provides the type of word (i.e., verb, noun, adjective). By passing the type of word back to the parser, the parser can continue with the rest of the sentence to validate the sentence structure. If the sentence is in proper form, it is accepted and stored in the frame. If a word is unrecognizable, the system informs the data entry person of the word that could be misspelled or not found in the dictionary. If the sentence has an invalid structure, then the parser would so inform the data entry person.

2.3.1. Frames

All of the information within the image knowledge base is stored in frames. Each image has its own frame. Within the frame are four major slots (see Figure 1). Each slot contains a category of information about the image. The first slot contains image location information which includes fields such as CD-ROM volume name, image filename, maker of CD-ROM, version of product used by maker and most importantly a system-wide unique object-id.

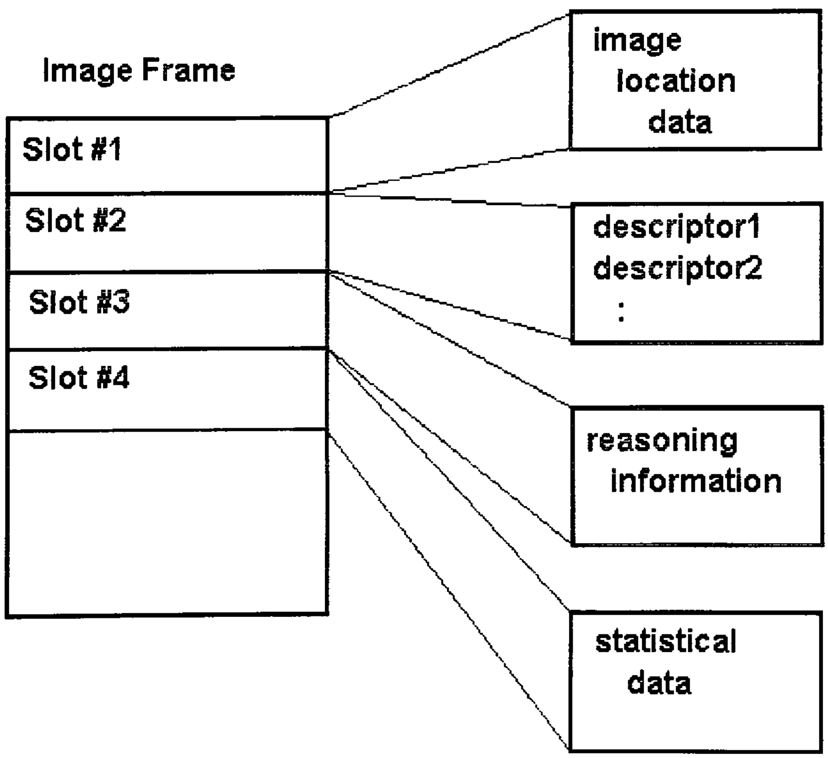

Figure 1 - Image Frames

In the second slot is information provided by the developer about the image. There are a set of descriptors along with the corresponding degree-of-importance. These descriptors are translated into an internal format to minimize storage and maximize retrieval performance. For the sake of the prototype, the original descriptor as well as the internal format are maintained. However, the real product could eliminate the original text if storage becomes an issue. Also, there exist some static fields such as name of picture, date of picture, author of picture, and picture location.

The reasoning slot contains additional information supporting the domain knowledge provided by the field expert within the system. This slot includes: questions to ask the user to further clarify a search, provide references to other cases, provide additional information about this case such as conclusions to draw or recommendations to follow. This slot has no applicability in the prototype application of the Smart Photo Album. However, for example, it would be vital to a medical diagnostic knowledge base providing the name of the diagnosis, pertinent information about the case and potential treatments for cure.

In the statistical data slot will be information recorded by the system as operations are performed. Fields will include: date and time the image was entered into system; date and time of last retrieval; and date and time of last modification. The statistical information will be transparent to the end-user of the system. It will have limited value within the prototype; however, it could be used for such things as caching to improve system performance.

2.3.2. Architecture

Figure 2 outlines the major architectural components of the prototype. Architectural components include: the knowledge base, the control mechanism, and the inference mechanism. The knowledge base contains the image frames, their indexes, the dictionary and the thesaurus. The control mechanism contains the allowable sentence structures, weights for each value of the degree-of-importance factor, confidence factors, and search heuristics. The inference mechanism contains the modules to support searching of cases, ordering of candidate images, and adding of information to the knowledge base.

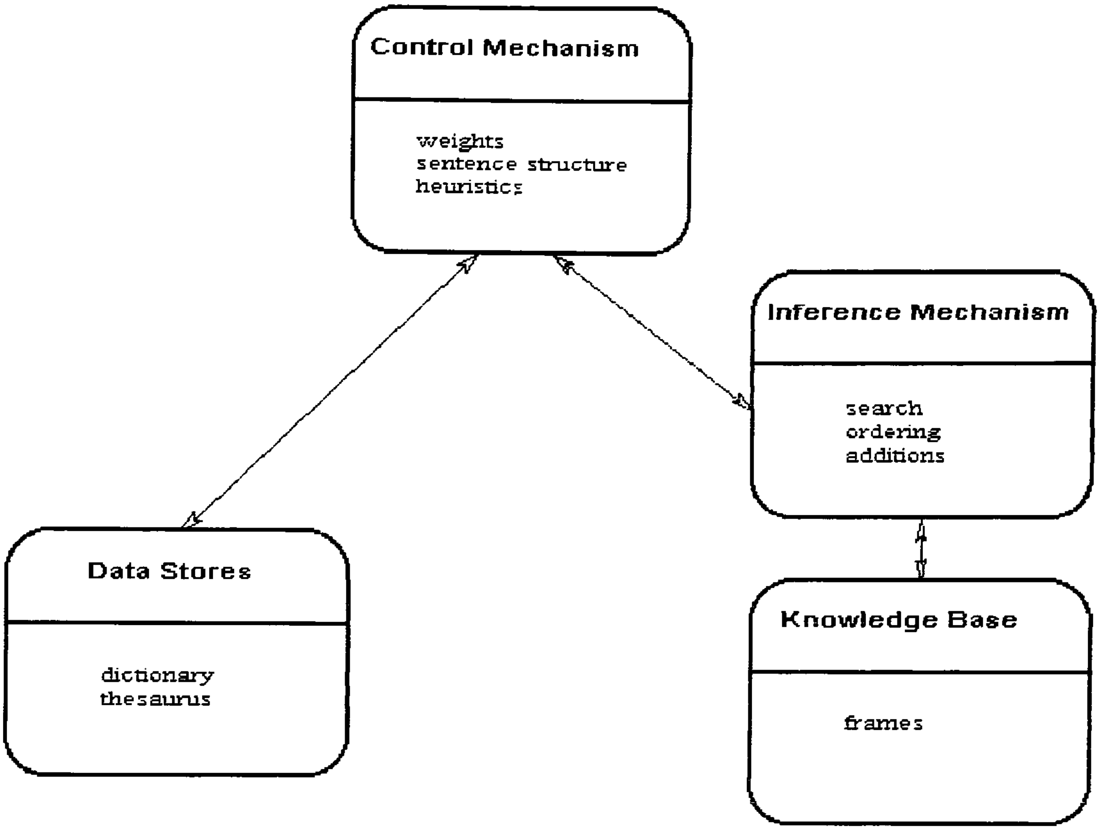

Figure 2 - Organization of the knowledge base

For the prototype, it is assumed that there will be only one knowledge base or case base.

2.3.3. Ordering

Given the sparse domain of knowledge, it would be difficult to organize ahead-of-time the information that we will be receiving from the user[13] This would only be possible if the system was built by a domain expert. Even if adequate data existed, knowledge about the mix of queries to be made are required to create a structure. Thus to eliminate bias in trying to identify a structure, a general and dynamic approach is required.

The image frames are hierarchically ordered using a set of indexes. This hierarchical ordering will be used to reduce retrieval time. Each index contains the list of images in the knowledge base that contain a given descriptor word. Each descriptor word that has at least one image in the knowledge base will have an index. Initially a small image knowledge base will have a very sparse set of indexes. However, as the knowledge base grows, the indexes will begin to expand. This expansion will theoretically be along similar image attributes given that each data entry person tends to use a consistent set of attributes across

all pictures. Also the subject matter for a given knowledge base should have similarities even if this is at a very general level.

The ordering does not provide a true hierarchical ordering given the multiple descriptors and words in each descriptor for an image. The ordering only accounts for the first phase of the search which is to drastically reduce the number of images in the knowledge base to a small set that can be more carefully analyzed.

2.3.4. Search

The search process is performed as a phased process using several stages of processing. Before the search process begins, the descriptors entered as search criteria are validated using the same parser used during data entry. Given a set of valid descriptors, these will be decomposed into a complete list of words to search on. Each index relating to a word in the list will be searched providing a list of images that match a word in one of the descriptors. The individual image lists will be combined into a single list eliminating all redundant images. However, the information about the images in the list will be expanded to include the number of occurrences that image has in the set of image lists. This single list of images completes the first stage of processing which provides all potential candidates in the image knowledge base that have some match to the search criteria.

The second stage of processing begins the analysis of the candidates. The analysis process attempts to rank the candidate images by similarity to the search criteria. The single list of images is sorted based upon number of occurrences. Thus, the images with the most number of matching descriptor words are potentially the most highly ranked image (excluding any degree-of-importance factors taken into consideration at this time). Each candidate is then compared and a weighted sum is calculated for each matching attribute. When calculating the similarity between the query information and a candidate image, the method ignores attributes considered to be irrelevant[10] (i.e. additional descriptors that are part of the image but do not match the search criteria will not be used).

Weights are assigned to the degree-of-importance factors for each candidate image. A higher degree-of-importance would be a stronger indication of that attribute within the image. Additionally weights are assigned to the degree-of-importance factor on the search criteria. The higher the factor, the stronger a requirement for a particular attribute given other search attributes. Each candidate is viewed separately for a total match against all search criteria information. The results of the search is an ordered list starting with the highest weighted sum. The user is able to view the information (actual image and attribute information) about the resulting list.

Another stage of the search would be for searches that produce zero results. Here chaining will be used to expand the list of search candidates.[13] A thesaurus will be used to find alternate words to those given as search criteria. These alternate words will then be used to redo the search. Hopefully using the alternate words at least one match can be found within the image knowledge base.

If a search still produces a zero result even after chaining, then the system will have an opportunity to perform failure-driven learning[11]. Since the user had an expectation that failed to be visualized, the user should be able to adjust the cases within the knowledge base. This could be in the form of modifying a particular or set of cases. It could also be by expanding the thesaurus. It would be most beneficial if the system could determine what modifications might be necessary to achieve the user's expectations. Although this type of learning would be desirable it is beyond the scope of the prototype.

3. System Description

The focus of this thesis is the implementation of a prototype called "Smart Photo Album" This prototype embodies the design of a general image knowledge base using the PhotoCD product on the IBM PC platform. The knowledge about the images is entered by a developer using a context-free grammar. A dictionary will be used to validate the words entered. The end-user will enter queries in the same manner. The system uses case-based reasoning as its searching technique to retrieve images from its knowledge base. The search provides the closest approximation to the user entered search criteria or query information in an ordered fashion. Although the prototype is not a fully functional product, enough design effort has been incorporated into it to understand what the entire product would look like.

3.1. Functional Specification

The interaction between the system and all outside sources can be viewed from the Context Diagram as shown in Figure 3. The outside sources to the system are the Developer and the User. All other components, like the PhotoCD, are incorporated within the system. Note that although the Context Diagram shows a Developer and a User, this can easily be the same person and mostly will be in the case of the Smart Photo Album. In situations where the system would be used in the development of an expert system (i.e., all required knowledge was already incorporated into the PhotoCD and software by a domain expert), the Developer would be the expert performing the development and the User would be the recipient and executer of the expert system.

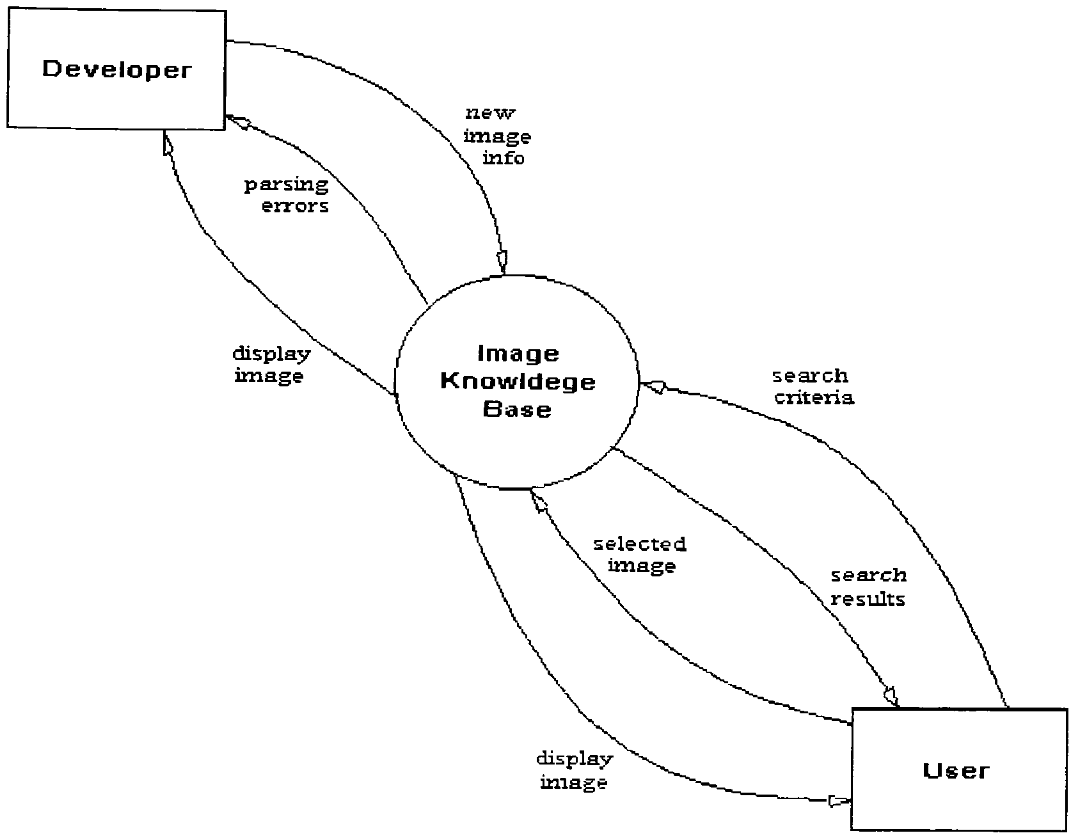

Figure 3 - Context Diagram

The breakdown of the context diagram can be found in Figure 4. The system has two main functions or components. It also has three data stores. The first function is the ability to add images to the knowledge base. This will require describing the image to the system by providing descriptors about the image. The addition of the images to the CD-ROM will be distant in time from the data entry time. Certainly a good photographer will keep notes as pictures are taken about the specifics of each picture. These notes are invaluable during data entry. However for the casual photographer, the number of pictures taken is not large, allowing for the user to reconstruct the circumstances of the picture as the user is looking at the contact sheet in the jewel case. This mental reconstruction would take the place of any missing notes. Displaying the image on the screen during data entry is also possible.

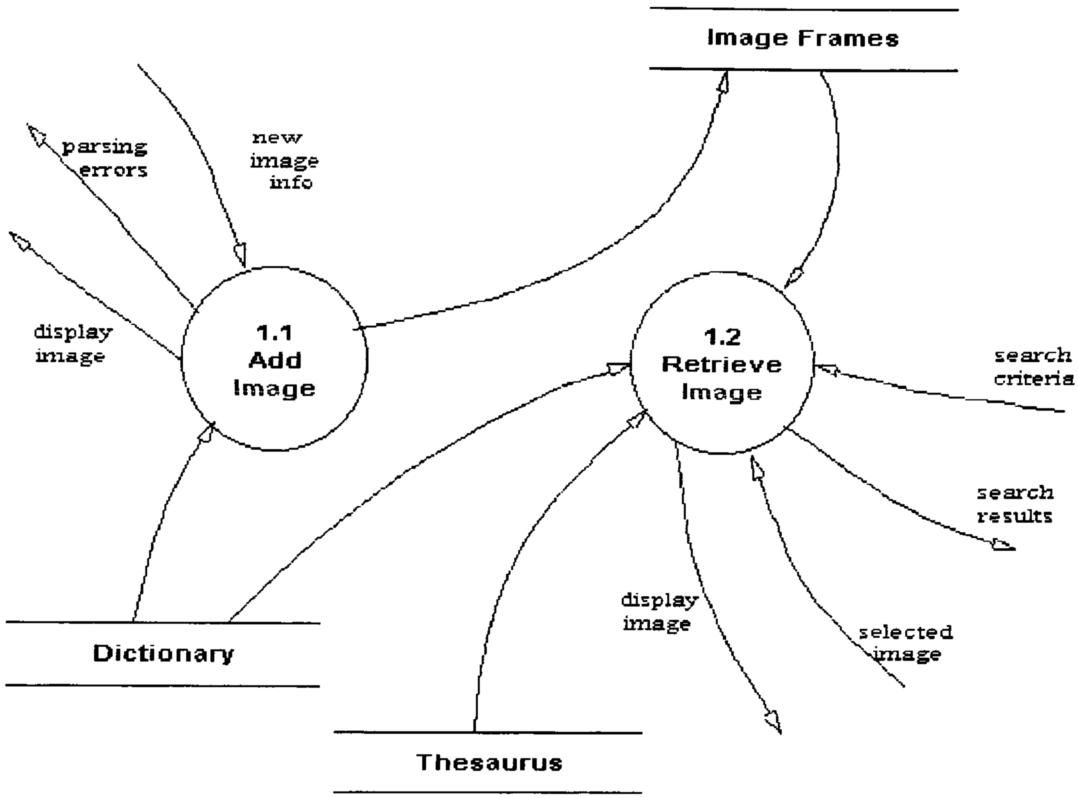

Figure 4 - Data Flow Diagram 1.0

The most common sequence of events is for the user to take a roll of pictures. Then the user will have the pictures developed and added to the latest PhotoCD. If this is the first roll of film ever entered on a PhotoCD, or if the last development process filled the existing PhotoCD, then a new PhotoCD will be created by the film developer. Next the user will load the PhotoCD into the drive with the Smart Photo Album application active. By activating the "add" function, the system will validate for the last known image and look for additions on the PhotoCD. The system can then prompt the user for information about each new additional image.

The data entry activity adds to the Image Frames data store. The Dictionary data store is used in the validation of entered words as part of the descriptors. The Dictionary is used in a read-only fashion during data entry.

The second function would be the retrieval of an image by specifying search criteria. The search criteria is entered as a set of descriptors. The Dictionary data store is used in the validation as before. The retrieval can be an iterative process providing more or less criteria depending upon the number of images found in the search. The retrieved images can then be displayed. A Thesaurus data store could also be

used for word lookup in case the initial retrieval failed to produce any results. The Thesaurus data store is used in a read-only fashion during retrieval.

3.1.1. Limitations and Restrictions

The prototype developed for this thesis is limited in scope and does not perform the following functions:

* natural language processing - The system is limited to simple sentence structures that are not rigorously validated against the English language.
* Only a subset of the English language will be incorporated and no foreign languages.
* The dictionary is limited in size; however it can be extended by the user.
* Although the data structures support it, the thesaurus has not been incorporated into the system. These is no means to enter this information. Also, it will not be available during a search even when a search fails to produce any results. Other creative ways for its use have not been explored.
* Multiple search criteria entered is ANDed together. Other boolean logic operators such as OR and NOT are not supported.
* The system supports a single CD-ROM Disc. Support for a set of CD-ROM's which would involve asking the user to load a specified CD-ROM in the drive (or by having a jukebox perform that action) is not incorporated.
* No support for a CD-ROM jukebox hardware. The system is built with a single CD-ROM.
* The system will not allow the user to modify the "control structure" information (i.e. weights assigned to the degree-of-importance, cutoff on maximum number of images as search results).
* There system utilizes only one knowledge base or case base at a time. Multiple knowledge bases activated and relationships between them are not supported.
* Failure-driven learning will not be performed as part of the prototype.

The Smart Photo Album prototype requires the user perform all data entry into the knowledge base. However, the system could be easily adapted to a knowledge base built by an expert by incorporating the addition and modification of images in the knowledge base into the development tools for the expert. The data entry capability would then be removed from the application presented to the end-user. Although this variation of the system could be developed, it is outside of the scope of this project.

3.1.2. User Inputs

As outlined in Figure 4, there are two input functions to perform. The first will be for the developer to enter all descriptors and degree-of-importance factors for each new image into the system. The second

will be for the end-user to enter search criteria. At any time, the user can also ask for a picture to be displayed from any type of list of pictures.

In addition, all users are required to respond to error prompts such as an incorrectly entered image descriptors. Along the same lines, all users will need to acknowledge informational messages.

User input is developed using a graphical user interface (GUI) for reduced training by the user, optimal user productivity and friendliness. The input follows standard conventions for the Microsoft Windows GUI as published by Microsoft. Input is provided by the user from the pointing device (i.e. mouse) and keyboard. Accelerator or function keys are utilized in commonly accessed functions along with appropriate defaults. All detailed input such as descriptors use modal dialog boxes.

3.1.3. Outputs

The system has two forms of output. The first is raster information in the form of varying resolutions of images and picons. The raster information is displayed in separate child windows. The second is in the form of text. All text is incorporated into the graphical user interface in the client area of child windows, dialog boxes and notify boxes. Key areas of notification to the user deal with errors or progress made. When descriptors are being entered, the system validates the words typed. If any words fail to be found within the dictionary or any sentence structures fails to be properly parsed, a notify box is presented requesting the problem be corrected before proceeding.

3.1.4. System Files

A number of data files exist in the prototype. They include:

1. dictionary or lexicon - contains all valid words that can be used as image descriptors for all images within the case
2. thesaurus - lookup for alternate words to those found in image descriptors to expand a given search (part of dictionary)
3. image knowledge base - image frames collectively stored within a file representing a given photo album
4. knowledge base index - a file matching descriptor words and images within the database for increased performance during searches

The dictionary and image descriptors are tightly integrated. Each word in the dictionary is given a unique integer value. This integer value is the internal representation of each descriptor word. Thus, a descriptor is internally stored as a sequence of integers where the sequences has specific meaning to the user. Storing integers instead of text strings reduces the disk space requirements.

A thesaurus of words can be used to either broaden or narrow the search as desired. The thesaurus would provide similar words in the dictionary to those that the user currently entered as search criteria to expand the search.

Although the prototype supports only a single active photo album (or image knowledge base), a user can create multiple photo albums. This does not imply that each PhotoCD will be its own photo album. A photo album is expected to have numerous PhotoCD's within the image knowledge base. Having multiple photo albums would allow the user to segregate the entire collection of PhotoCD's into groups. An example might be a photojournalist who wants to keep business related pictures separate from pictures dealing with their personal life.

3.2. System Specification

The system is developed using Microsoft Windows and a combination of C and C++ modules. The prototype follows all standard conventions provided by Microsoft for the Windows environment. Thus any user already familiar with other Microsoft Windows applications will have no difficulty in executing the Smart Photo Album prototype. The user interface portion of the system is event-driven. The application has a main menu (after a photo album has been activated). As part of each main menu item is a pulldown menu that lists all the actions or choices possible. Some menus have a secondary pulldown for a more detailed selection. Others require further input as indicated with an ellipsis. When a menu item is not applicable, its appearance is grayed in color and is disabled (not selectable by the user). The graying of certain actions prevents the user from making mistakes. The system has several menus. The current menu is dependent on the active window. For further details refer to the User's Manual section in the Appendix.

Some of the menu actions are performed without further input. Other actions provide a dialog box for gaining specific input from the user (these are recognized by an ellipsis). On the dialog boxes will be pushbuttons for completion or cancellation of the desired activity (typically an OK and CANCEL). This input is validated and stored away before the dialog box is removed from the screen when the user presses the "OK" button. Also there are combo-boxes for data entry items such as the degree-of-importance where a predefined list of items are available to select from.

Given the vast amount of information that the user might want to view simultaneously, the GUI was developed using MS-Windows Multiple Document Interface (MDI). MDI supports a variable number of child windows attached to the initial parent window. The child windows are displayed in the client area of the parent window. The child windows support the following items: a list of the picture on the CD-

ROM, a list of candidate pictures for each search, and for each image selected for display. Using MDI allows greater flexibility to the user including comparing the results of two searches.

3.2.1. System Organizational Chart

Given the event driven nature of the GUI, the entire system is event driven (see Figure 5). The two main functions outlined in Figure 4 (Add Image and Retrieve Image) are incorporated under the specific menu choices. When the "Add Picture" menu item is activated, the appropriate main-level routine is called to obtain the descriptors and degree-of-importance, perform the parsing, dictionary lookup, sentence validation, and storage of information into the case frame. When the "Find Picture" menu item is activated, the main-level routine is called to obtain the descriptors and degree-of-importance, parse the search criteria, activate search algorithm, and present results.

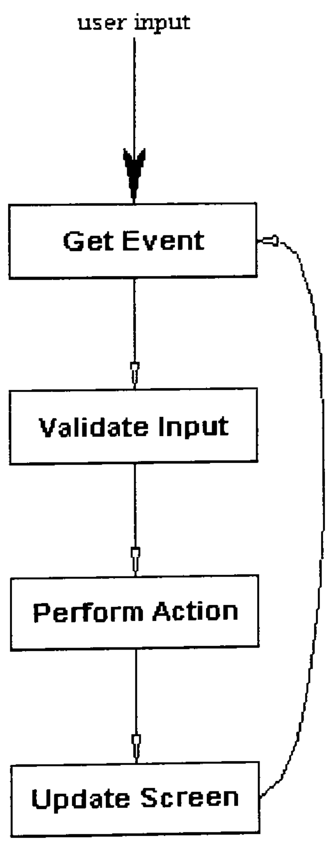

Figure 5 - Event Driven System

3.2.2. Data Structures

A number of data structures are implemented as part of the prototype. They support the collection of objects that are required to implement the system. They include:

1. varying length character strings (allocation of memory is limited to current string size but the string can be edited at any time creating a longer length string),
2. varying arrays of IDs,
3. varying array of pointers where the pointers provide an address to additional data structures in memory,
4. double-linked lists of objects, and
5. map of IDs to objects.

These structures are implemented using C++ classes for maximum reuse.

3.2.3. Equipment Configuration

The system is developed for the IBM PC and PC compatible market using an Intel 386 or better processor. The system requires a minimum of a 14 inch, color, VGA resolution monitor. The system needs a minimum of 4 MBytes of main memory to support display of images. Note that additional memory is recommended to increase display and search performance. A magnetic disk of a minimum of 80 MBytes is required to support software and database information. Most importantly is a CD-ROM XA disk drive that is compatible/certified for PhotoCD. Although the prototype is developed using a specific set of hardware, a real product produced from the prototype should be easily extendable to different hardware configurations (i.e., higher resolution monitor, larger screen monitor, faster display adapter, different CD-ROM drives, varying memory configuration). However, the prototype is not tested against any varying set of hardware.

3.2.4. Implementation Tools

The executing system requires MS-DOS 3.3 or higher and Microsoft Windows 3.X as part of its environment. The system also requires the necessary Windows drivers loaded for the specific hardware. In particular, a Windows driver for the CD-ROM compatible with the PhotoCD toolkit is mandatory. In addition would be the application software developed (i.e. Smart Photo Album) which will include the PhotoCD toolkit. For demonstration purposes, a sample PhotoCD is needed with visually identifiable pictures.

The development system requires the same environment as the executing system with the following additions:

1. Microsoft C/C++ compiler - version 7,
2. Microsoft Window's Software Development Kit (SDK) - V3.1,
3. standard development tools (i.e., editor and debugger) for the PC environment (i.e. Programmer's Workbench), and
4. the Eastman Kodak Photo CD Access Developer Toolkit for interfacing to the PhotoCD.

These software packages allow for the development of an executable and supporting dynamic link libraries (DLL), the combination of which make up the Smart Photo Album prototype.

4. Conclusions

The development of the Smart Photo Album was successful from a number of perspectives. The first is the completion of the stated objectives in the thesis pre-proposal. The second is the full implementation of a Microsoft Windows application using Microsoft standard conventions. The third is a successful object-oriented design and implementation using C++ as the programming language. This is significant given the author's limited prior experience in both object-oriented design and coding. The most significant accomplishment is the development of a search algorithm that uses CBR providing for "nearest neighbor' type searches of an image knowledge base.

The net effect of the development effort is 38 source code files for over 260KBytes in the following breakdown: 15 header files, 14 C++ files, 6 resource files (menus and icons), and 3 files supporting the building of the executable (make files). Included in the executable are numerous include and library files as part of the compiler and several toolkits used. Given the 2.5 months of development, this amounts to over 100KBytes of source code per month. The list of files are shown in the Appendix.

4.1. Problems Encountered and Solved

Certainly one of the biggest challenges in the project was the integration of the PhotoCD toolkit and use of a CD-ROM drive. Being a relatively new product on the market, there is limited support especially from hardware vendors offering multi-session drives. The second problem encountered was the large amount of code that needed to be generated to support all aspects of the system: case frames, opening and saving of database files, dictionary creation and access, development of a search algorithm, and the creation of the GUI. However, within the generated code is a large amount of flexibility and expansibility than can be visible from the user interface. This is largely due to the object-oriented methodology.

4.2. Discrepancies and Shortcomings of the System

As in all significant development activities, modifications to the direction and schedule were required. One of the areas that was modified was the validation of input descriptors. The original intent was to build a finite state machine that would parse the input using a context-free grammar. Unfortunately, there was insufficient time to develop the finite state machine. This represents a limitation within the prototype. It requires the user type in meaningful sentence structures since the prototype cannot validate them beyond checking for each word in the dictionary. This would be a necessary feature for a real product.

Another limitation in the prototype is the "reasoning" the system performs if a search fails to produce any results. The original intent was for the system to inform the user that chaining will be invoked on a search (i.e., use of the thesaurus). The chaining would "try harder" to find a matching image given the search criteria. Theoretically this would minimize the searches producing zero results. By notifying the user, the user would have the option of cancelling the chaining. This would be important if the user felt that entering another set of descriptors would be more prudent instead of continuing to extend the search given the current set of parameters.

Despite these shortcomings, a larger number of optional areas were undertaken and completed in the development process. They include:

1. support for modification and deletion of existing descriptors and degree-of-importance,
2. allowing the user to extend the dictionary by adding words through the user interface,
3. development of varying image manipulation and display modes ("Rotate", "Size", and "Mirror"),
4. a GUI utilizing Windows MDI supporting multiple finds and images to be displayed simultaneously, and
5. other GUI features (i.e., horizontal and vertical scroll bars on the child text windows).

4.3. Future Development

A number of areas should be expanded to translate this prototype to a real product. Most of them center around adding robustness to the existing system. As already mentioned, the incorporation of a finite state machine for validating sentence structure is critical. Another key area is the support of multiple CD-ROM Discs or volumes.

Another area of interesting further development is in the search result reporting. From a user interface perspective, it is desirable to indicate the progress being made during a search. This is of increasing importance as the size of the image knowledge base grows. This could be simply an information message box indicating to the user the current number of images it found matching the search criteria as it is

performing the search. For a relatively small number of matching images, an alternate form for displaying the results could be developed. This could have the system display the picons with the matching percentage. The user can perform a close inspection on any of the images selected by expanding the picon to higher resolution. Further, if the number of images found was too large, the system should require the user to specify additional search criteria and terminate the search. Also, different weighting schemes could be developed that could provide more meaningful results given a large and potentially complex image knowledge base.

For performance reasons within a fairly large image knowledge base, the indexing would require further refinements. The current prototype indexes every word entered by the user. Although many words can be used in a descriptor, only nouns and verbs should be indexed. Attempting to index articles, adjectives, and other parts of speech would produce very little benefit. The adjectives provided in the search criteria could be used for further refinement of the ordering of candidate images.

Another interesting development would be the incorporation of a neural network into the system. The neural network could be used to produce similar images based upon the current picture being displayed. This would require the translation of the pixel information of the active image into something meaningful for the neural net. The result could be pictures of similar colors or proportions of color as the currently displayed image.

5. Bibliography

(1) Fox, Edward A., "Advances in Interactive Digital Multimedia Systems", *IEEE Computer*, October 1991, pg. 9

(2) "Kodak Photo CD Products - A Planning Guide for Developers", Eastman Kodak Company, Part No. DCI 200R, 1992

(3) Story, Guy A., O'Gorman, Lawrence, Fox, David, Schaper, Louise Levy, and Jagadish, H. V., "The RightPages Image-Based Electronic Library for Alerting and Browsing", *IEEE Computer*, September 1992, pg. 17

(4) Rabitti, Fausto and Savino, Pasquale, "Querying Semantic Image Databases" *Proceedings of the International Society for Optical Engineering*, Vol 1662, February 13-14, 1992, pg. 69

(5) Gary, James E. and Mehrotra, Rajiv, "Shape Similarity-based Retrieval in Image Database Systems", *Proceedings of the International Society for Optical Engineering*, Vol 1662, February 13-14, 1992, pg. 2

(6) Matsuyama, Takashi and Hwang, Vincent Shang-Shouq, "SIGMA - A Knowledge-Based Aerial Image Understanding System", New York, NY: Plenum Press, 1990

(7) E. Panofsky, "Iconography and Iconology: An Introduction to the Study of Renaissance Art", reprinted in *Meaning in the Visual Arts*, London, 1983, pg.51

(8) "PixTex/EFS", Excalibur Technologies Inc.

(9) Subar, David, "Case-Based Reasoning: a Familiar Story", *Computer Select*, Doc #49478

(10) Kitano, Hiroaki, Shibata, Akihiro, Shimazu, Hideo, Kajihara, Juichirou, and Sato, Atsumi, "Building Large-Scale and Corporate-Wide Case-Based Reasoning Systems: Integration of Organizational and Machine Executable Algorithms", *Proceedings of the Tenth National Conference On Artificial Intelligence*, July 12-16, 1992, pg. 843

(11) Edelson, Daniel C., "When Should a Cheetah Remind You of a Bat?", *Proceedings of the Tenth National Conference On Artificial Intelligence*, July 12-16, 1992, pg. 667

(12) "Functional Specifications for REMIND Development Version 1.0", Cognitive Systems Inc.

(13) Craddock, A. Julian, "Common Sense Retrieval", *Proceedings of the Tenth National Conference On Artificial Intelligence*, July 12-16, 1992, pg. 661

5.1. Books and Articles Read for Thesis

Barletta, Ralph, "An Introduction to Case-Based Reasoning", *AI Expert*, August 1991, pg. 43

Hibler, J.N. David, Leung, Clement H. C., Mannock, Keith L., and Mwara, Njagi, "A System for Content-based Storage and Retrieval in an Image Database", *Proceedings of the International Society for Optical Engineering - Image Storage and Retrieval Systems*, Vol 1662, February 13-14, 1992, pg. 80

Kolodner, Janet L., "Retrieving Events from a Case Memory: A Parallel Implementation", *Proceedings from Case-Based Reasoning Workshop*", May 1988, pg. 233

Krovvidy, Srinivas, Wee, William G. and Han, C. Y., "Case Based Reasoning Approach for Heuristic Search", *Proceedings of the International Society for Optical Engineering - Applications of Artificial Intelligence IX*, April 2-4, 1991, pg. 216

Luger, George F. and Stubblefield, William A., "ARTIFICIAL INTELLIGENCE and the Design of Expert Systems", Redwood City, CA: The Benjamin/Cummings Publishing Company, 1989

Rauch-Hindin, Wendy B., "A Guide to Commercial Artificial Intelligence", Englewood Cliffs, NJ: Prentice Hall, 1988

Slade, Stephen, "Case-Based Reasoning: A Research Paradigm", *AI Magazine*, 1991, pg. 42

Stottler, Richard H., "Case-Based Reasoning for Bid Preparation", *AI Expert*, March 1992, pg. 44

6. Glossary

animated graphics - Moving diagrams or cartoons. Often found in computer-based courseware. Animated graphics take up far less space than video images.

API - Application Programming Interface. Generic term for any language and format used by one program to help it communicate with another. Form of modularity providing interchangeability of components.

bitmap - representation of characters of graphics by individual color or black and white pixels. Pixels arranged in row and then column order. Each pixel is represented by some number of bits (black and white would be one bit).

bitmapped graphics - images which are created with sets of pixels. Also called raster graphics. Contrast with vector graphics.

cache - small portion of high-speed memory used for temporary storage of frequently used data. Reduces access time.

case - the set of information pertaining to a single experience or event desiring to be captured.

case sensitive - knows the difference between capital and lower case letters.

CBR - Case Based Reasoning. An artificial intelligence reasoning technique using existing cases as the source of knowledge.

CD - Compact Disc - a standard medium for optical storage of digital data in machine-readable form, accessible with a laser-based reader. CD's are approximately 4.75 inches in diameter and contain 650 Mbytes of data.

CD-ROM - Compact Disc Read Only Memory. A data storage system using Compact Discs as the medium.

CMYK - Cyan, Magenta, Yellow and Black. The four process colors used in color reproductions. Contrast to RGB as another color model.

compound document - file that has more than one element (i.e. text, graphics, voice, video) contained within.

compression - hardware or software technique for reducing the size of images so they occupy less storage space. Reversed by decompression.

contextual search - to locate a document stored in a system by search for text that appears in them rather than by searching for them by filename or other indexing parameters.

continuous tone - an image that has all the values of gray or color in it (i.e. a photograph).

DCI - Desktop Color Imaging. Using color images for business applications on your PC.

DPI - Dots Per Inch. A measurement of input and output device resolution and quality.

DVI - Digital Video Interactive. A compression technique for data, audio and video.

frame - a view of knowledge or memory as a static data structure used to represent and organize well-understood, stereotyped situations. Outlined in a paper by Minsky in 1975.

GByte - gigabyte. A thousand (1024) megabytes of data.

GUI - Graphical User Interface. Computer control system that allows the user to command the computer by "pointing-and-clicking" usually with a mouse at pictures of icons rather than type in commands.

high resolution - any image that is displayed in better quality be increasing the number of pixels per inch.

Huffman encoding - a lossless compression algorithm that replaces frequently occurring data strings with shorter codes.

hypermedia - a way of delivering information that provides multiple connected pathways through a body of information. Allows the user to jump easily from one topic to a related topic or supplementary material.

icon - a miniaturized image representing a program or device that is the basis of a graphical user interface. A smaller representation of a full image.

jukebox - a device that holds multiple optical discs and one or more disc drives, and can swap discs in and out of the drive as needed. Also called disc library or autochanger.

MByte - megabyte. A thousand (1024) kilobytes of data.

multimedia - combining more than one media for dissemination of information (i.e. text, audio, graphics, image, animation, full-motion video).

OCR - Optical Character Recognition. A device used to interpret ASCII characters from specific regions of raster information by using some form of pattern matching.

optical disc - a direct access storage device that is written and read by laser light.

PC - Personal Computer. Used to indicate an IBM of compatible personal computer.

PhotoCD - a CD-ROM based product produced by Eastman Kodak Co. that provides a high-resolution set of images transferred from 35mm film.

picons - a low resolution picture icons on a display or output device.

pixel - picture element or dot that are part of an image.

proximity search - a feature of full-text searching in which every occurrence of a word within a certain distance of another word is found.

RGB - Red, Green, Blue. The primary colors used by color monitor displays.

SQL - Structure Query Language. A standardized language used for accessing/querying a relational database.

SuperVGA - extension to IBM's VGA standard that provides 800 x 600 pixel resolution.

VGA - Video Graphics Array. Standard IBM video display standard providing medium resolution text and graphics at 640 x 480 pixels.

WORM - Write Once Read Many. optical storage device where data is permanently engraved in the media.

XGA - Extended Graphics Adapter. IBM's graphics standard that includes VGA and higher resolutions up to 1024 x 768 pixels interlaced.

Appendix A - User's Manual

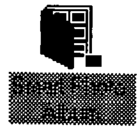

Figure A - Smart Photo Album icon

The Smart Photo Album application is loaded in any subdirectory of the user's choosing. The icon for the application, as shown in Figure A, is added to the Program Manager for easy access. The Smart Photo Album application is activated by double clicking on the icon. The activated application creates the parent window with standard Microsoft Windows controls such as window border, title bar, system menu (top left "-" button), maximize and minimize buttons (top right), and menu. The parent window with one menu pulldown is shown in Figure B.

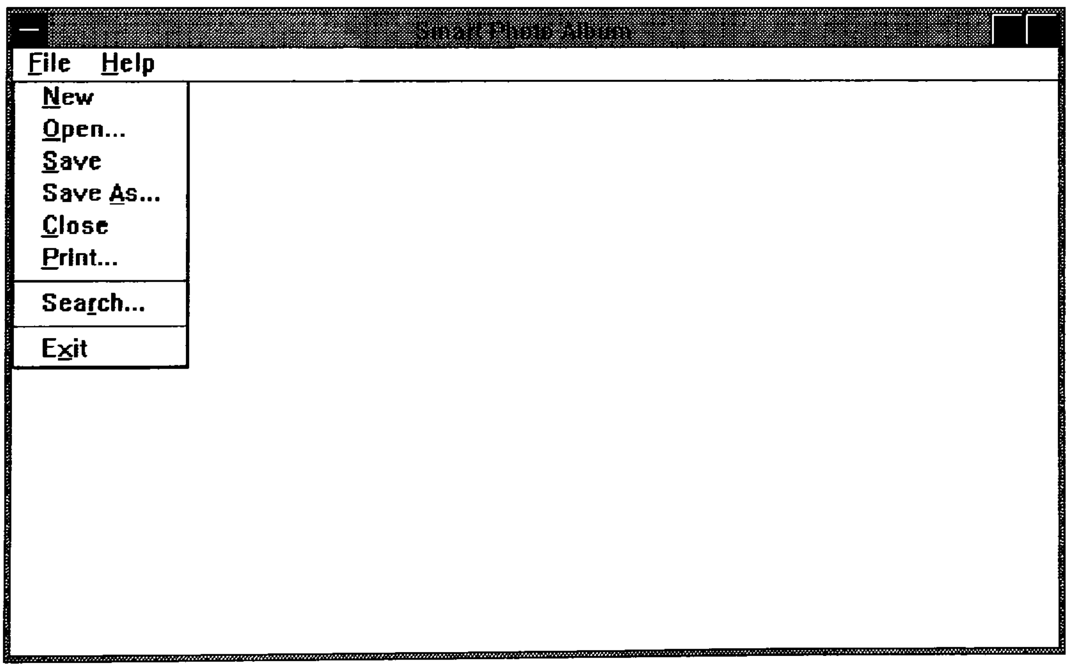

Figure B - Initial menu with "File" selected

The parent window begins with two pulldown menu selections available on the main menu bar, "File" and "Help". The "File" pulldown menu supports "New" for creating a new photo album and "Open" for accessing an already existing photo album. The "Open" choice brings up a file selection dialog box

allowing the user to select the disk drive, directory and filename of the existing photo album. All other menu choices are inactive except for "Exit" which allows the user to quit the application. Once the user has selected an existing photo album, the parent window changes as shown in Figure D.

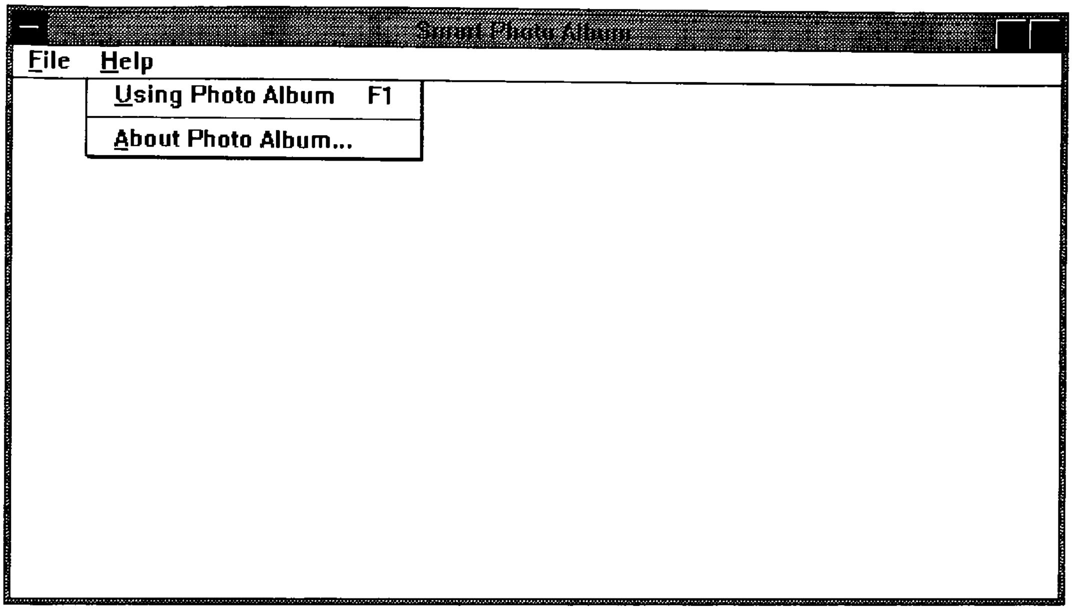

Figure C - Initial menu with "Help" selected

The "Help" pulldown menu as shown in Figure C, has two choices. "Using Photo Album" provides a dialog box with context-sensitive help. Note the use of the "F1" accelerator key for easy activation The "About Photo Album" is a quick way to determine the developer of the software package, the version number, and any other restrictions.

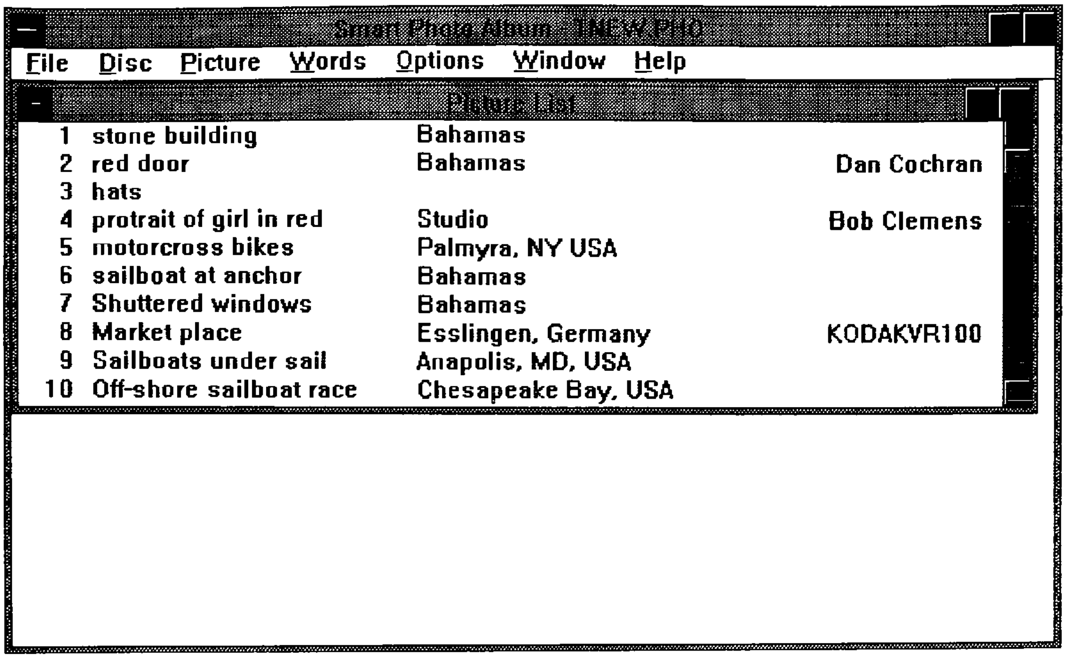

Figure D - Open photo album

With an open photo album, the main menu bar changes to include several additional selections: "Disc", "Picture", "Words", "Options", and "Window" These extend the "File" and "Help" selections where all pulldown menu choices have been activated (most importantly is "Save" which allows us to save any additions and modifications made to the open photo album). The other most significant change after the "Open", is the creation of a child window that displays the current contents of the image knowledge base. The child window has its own window border, title bar, system menu (top left "-" button), maximize and minimize buttons (top right), and scroll bars. Since the child window is displayed in the space allocated to the parent window's client area, the horizontal and vertical scroll bars are activated as need to view all text within the child window.

The "Disc" pulldown menu supports manipulation of several CD-ROMs. Since the prototype is limited to a single CD-ROM, the only menu choice is "Load" which verifies the CD-ROM drive is accessible and that the demonstration CD-ROM is loaded in the drive.

The "Picture" pulldown menu, as shown in Figure E, supports manipulation of the active photo album. Choices include: "Add", "Edit", and "Delete". Also available are "Display" for viewing an actual picture from the CD-ROM and "Find" which performs the CBR search. The "Display" selection creates another child window with the detailed pixels of the photograph in raster format within the client area using standard controls. The "Find" selection creates a child as seen in Figure F.

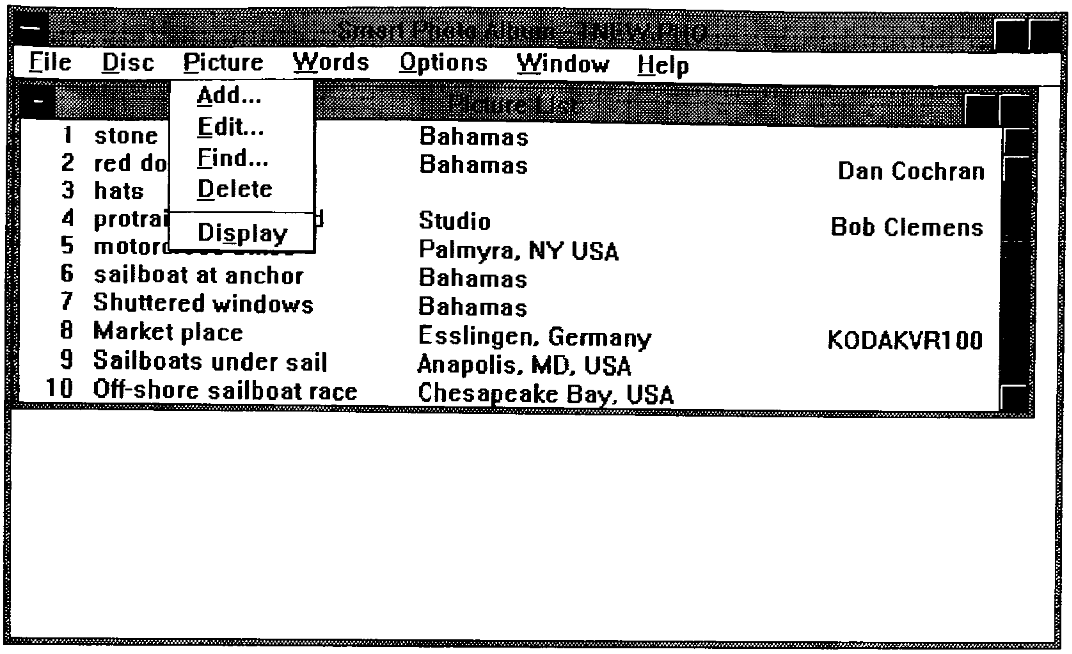

Figure E - "Picture" pulldown menu

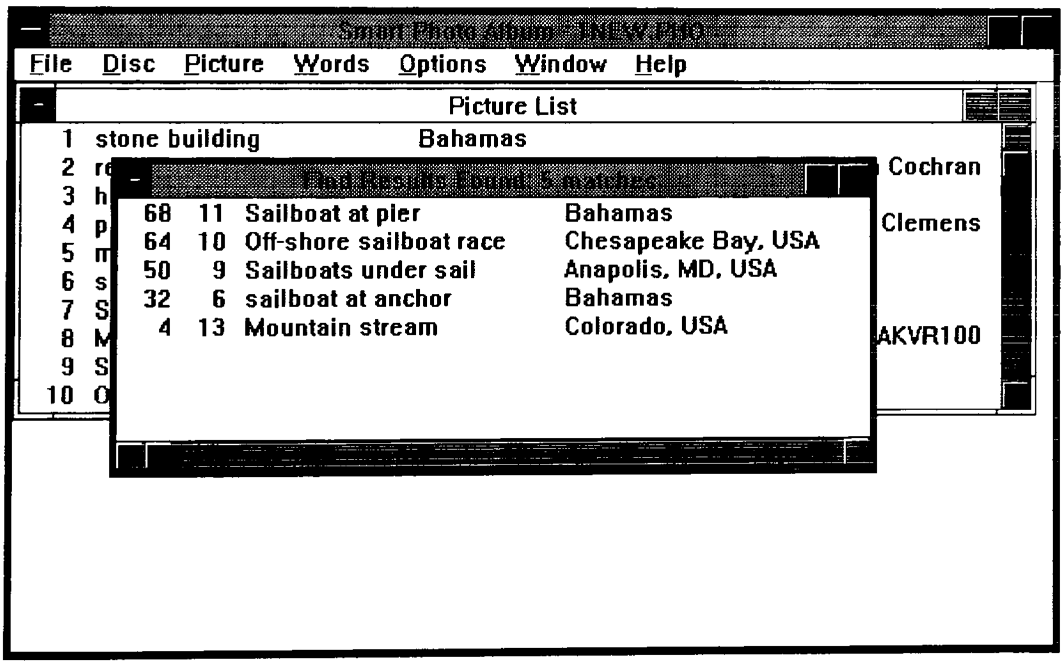

Figure F - "Find" results displayed

Under the "Words" pulldown is "Add" and "Synonym" These choices support the expansion of the dictionary and thesaurus. Note that "Synonym" has not been implemented within the prototype.

The "Option" pulldown menu supports user preferences as to the execution of the Smart Photo Album application.

The "Window" pulldown menu as shown in Figure G, supports manipulation of the child windows and the determination of the active or top child window. Choices include: "Cascade", "Tile", and "Arrange Icons". Note the availability of accelerator keys for "Cascade" and "Tile" In addition are choices for each open child window which is dependent on the number of active child windows. The top child window is check-marked for easy identification.

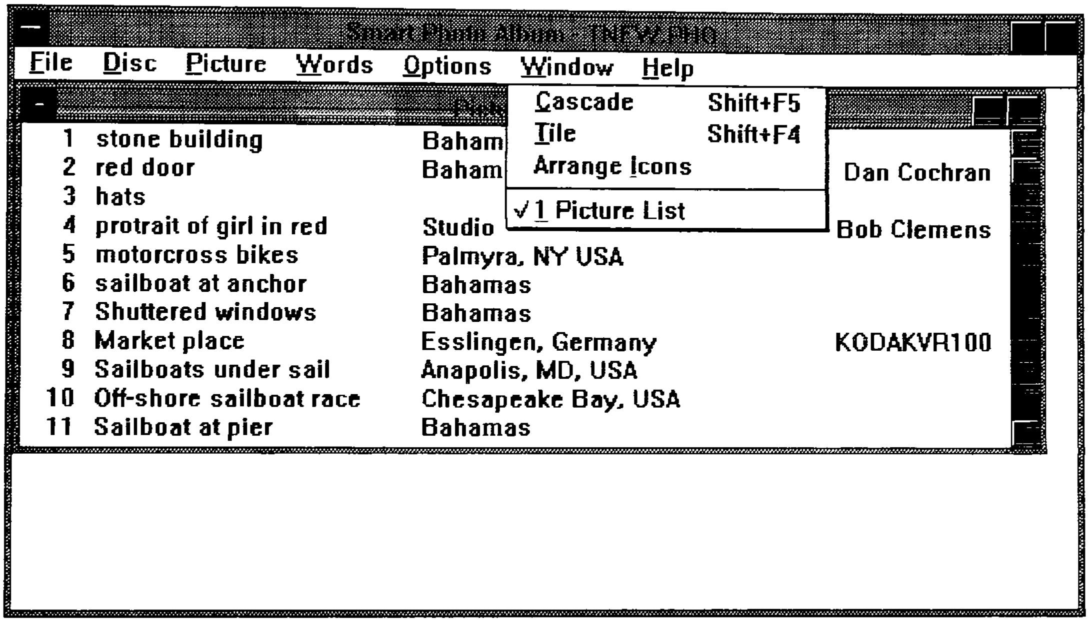

Figure G - "Window" pulldown menu

The "Find" results indicate similar information as the picture list from the "Open" with the addition of a matching value. This match value is the result of looking at all search descriptors against the descriptors of the picture at data entry time. The closest match is the highest score which is displayed first in the list.

Appendix B - Source Files

```
 Volume in drive C is RTE
 Directory of  C:\RIT\THESIS\CODE

DATABASE CPP     16540 02-08-93  10:53p
DATABASE H        2401 02-08-93   9:00p
DBINDEX  CPP     10338 01-20-93   8:21p
DBINDEX  H        2130 01-20-93   8:20p
DESCRIPT CPP     10634 01-27-93   8:30p
DESCRIPT H        3379 01-24-93   2:25p
DICTION  CPP     12498 02-08-93  10:50p
DICTION  H        3074 02-08-93  10:42p
FIND     ICO       766 01-31-93   3:59p
GENERAL  H        2465 02-08-93  10:02p
IDLIST   CPP      3016 02-04-93  10:54p
IDLIST   H         945 01-24-93   2:53p
IFRAME   CPP     22448 02-08-93   9:46p
IFRAME   H        4810 02-08-93   8:51p
PHOTO    MAK      7056 02-07-93   7:10p
PHOTO    STS      7303 02-08-93  11:11p
PHOTOAL  DEF       832 02-11-93  10:04p
PHOTOAL  DLG      5451 02-08-93  11:09p
PHOTOAL  ICO       766 12-19-92   3:04p
PHOTOAL  RC       7113 02-07-93  11:56a
PLIST    ICO       766 01-31-93   4:03p
RESOURCE H        2082 02-04-93  11:44p
SEARCH   CPP     12879 02-08-93   9:19p
SEARCH   H        3657 02-08-93   9:19p
SENTENCE CPP      7339 01-17-93  10:01p
SENTENCE H        1465 01-30-93   3:55p
WINBMAP  CPP     27557 02-14-93   8:30p
WINBMAP  H        4004 02-07-93   7:06p
WINDLG   CPP     10284 01-31-93   4:50p
WINDLG   H        2028 01-27-93   8:41p
WINFIND  CPP     17153 02-08-93  10:04p
WINFIND  H        2417 02-08-93   9:31p
WINMAIN  CPP     14523 02-08-93  10:27p
WINMAIN  H        2693 02-06-93   4:16p
WINPLIST CPP     19556 02-08-93  10:04p
WINPLIST H        2392 01-30-93  12:11p
WINUTIL  CPP      5905 02-07-93   2:32p
```