
Visual Information Retrieval Technology

A Virage Perspective

Revision 4

Dr. Amarnath Gupta
Amarnath@virage.com

Virage Inc.
177 Bovet Road
Suite 520
San Mateo, CA 94402
(415) 573-3210
Fax: 573-3211

info@virage.com
<http://www.virage.com>

© 1995–97 by Virage, Inc.
Distribute freely without modifications.

The authors wish to thank the entire Virage team for creating the vision of the technology and for giving it form.

Overview

Motivation

A significant event in the world of information systems in the past few years is the development of multimedia information systems. A multimedia information system goes beyond traditional database systems to incorporate various modes of non-textual digital data, such as digitized images and videos, in addition to textual information. It allows a user the same (or better) ease of use and flexibility of storage and access as traditional database systems. Today, thanks to an ever-increasing number of application areas like stock photography, medical imaging, digital video production, document imaging and so forth, gigabytes of image and video information are being produced every day. The need to handle this information has resulted in new technological requirements and challenges:

- Image and video data are much more voluminous than text, and need supporting technology for rapid and efficient storage and retrieval.
- There are several different modes in which a user would search for, view, and use images and videos.
- Even if multimedia information resides on different computers or locations, it should easily be available to the user.

Thus, representation, storage, retrieval, visualization and distribution of multimedia information is now a central theme both in the academic community and industry alike. At present, there is no technology available in the market that has the capability to manage this information. This white paper presents technology to meet this urgent need, produced by Virage, Inc., a pioneering company in Visual Information Management.

Virage, Inc. was formed in April of 1994 by Professor Ramesh Jain, Director of the Visual Computing Laboratory at the University of California, San Diego. The company's core technical team has done extensive academic research in multimedia information system technology and has developed a new model for such systems, called the Visual Information Management System (VIMSYS) model. Unlike traditional database systems, this model recognizes that most users prefer to search image and video information by what the image or video actually contains, rather than by keywords or descriptions associated with the visual information. This requires an information system very different from a traditional DBMS. In a traditional DBMS, an image is treated as a file name, or the raw image data exists as a binary large object (BLOB). The limitation is clear: a file name or the raw image data is useful for displaying the image, but not for describing it. In fact, textual descriptors such as a set of keywords are also inadequate to describe an image, simply because the same image might be described in different ways by different people. The only proper method by which the user can get access to the content of the image is by using image-analysis technology to extract the content from an image or video. Once extracted, the content represents most of what the user needs in order to organize, search, and locate necessary visual information.

This breakthrough concept of content extraction alleviates several technological problems. The foremost benefit is that it gives a user the power to retrieve visual information by asking a query like "Show me the pictures that look like this one." The system satisfies the query by comparing the content of the query picture with that of all target pictures in the database. This is called Query By Pictorial Example (QBPE), and is a simple form of content-based retrieval, a new paradigm in database management systems. An important concept in content-based retrieval is to determine how similar two pictures are to one another. The notion of similarity (versus exact matching as in database systems) is appropriate for visual information because multiple pictures of the same scene will not necessarily "match," although they are identical in content. In the paradigm of content-based retrieval, pictures are not simply matched, but are ranked in order of their similarity to the query picture. Another benefit is that content extraction results in very high information compression. The content of an image file may be expressed in as little as 1Kb or 2Kb, regardless of the original image size. As an image is inserted into a Virage database, the system extracts the content in terms of generic image properties such as its color, texture, shape and composition, and uses this information for all subsequent database operations. The original image is not accessed except for display. An additional strength of content extraction is that it allows the use of distributed database techniques for information storage and exchange across different computers without any platform dependence and without overloading the network bandwidth. Naturally, the VIMSYS model also supports textual attributes as would all standard databases.

The Virage Model of Visual Information

Following the aforementioned data model for visual information, Virage technology admits four layers of information abstraction: the raw image (the Image Representation Layer), the processed image (the Image Object Layer), the user's features of interest (called the Domain Object Layer) and the user's events of interest for videos (the Domain Event Layer). The top three layers form the content of the image or video.

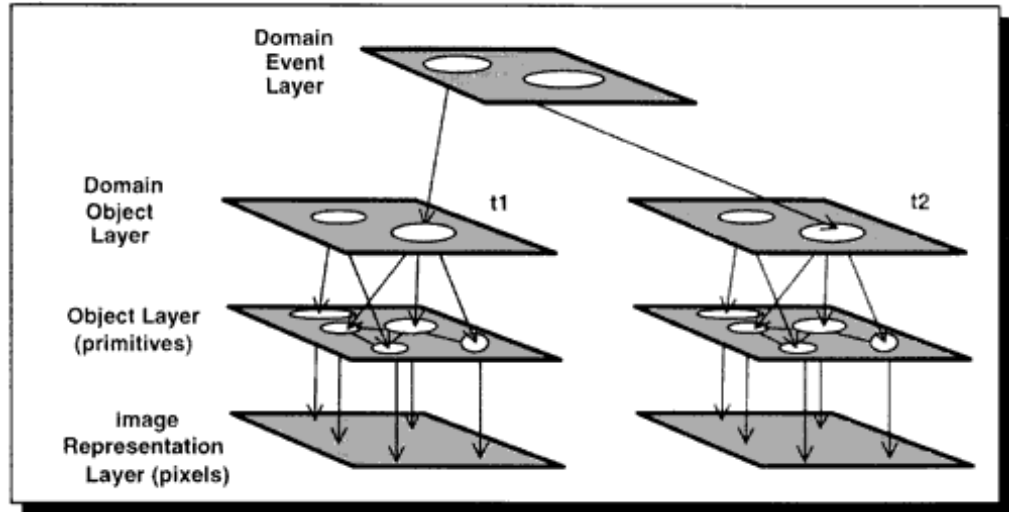


Fig. 1. The VIMSYS Model

Primitives

Image objects are computable image properties that can be localized in the spatial domain (arrangement of color), the frequency domain (sharp edge fragments), or by statistical methods (random texture). Virage calls these computed features primitives. Primitives are either global, computed over an entire image, or local, computed over smaller regions of the image. For each generic image property such as color, texture, and shape, a number of primitives are computed.

Distance Metrics

Since primitives are extracted by different computational processes, they belong to different topological spaces, each having different distance metrics defined for them. Computationally, these metrics are designed to be robust to small perturbations in the input data. Because the abstracted image primitives are defined in topological spaces, searching for similarity in any image property corresponds to finding a (partial) rank order of distances between a query primitive and other primitives in that same space. Also, since the space of image properties is essentially multidimensional, several different primitives are necessary to express the content of an image. This implies that individual distance metrics need to be combined into a composite metric using a method of weighted contributions.

Primitive Weighting

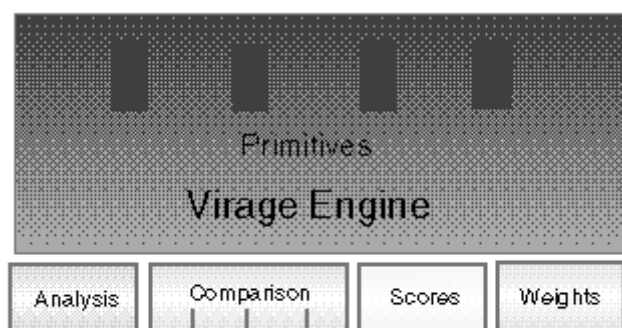
Our experience is that the overall similarity between two images lies literally "in the eye of the beholder." In other words, the perceptual distance between images is not computable in terms of topological metrics. The same user will also change his or her interpretation of similarity depending on the task at hand. To express this subjective element, Virage provides visual tools to the user to control which relative combinations of individual distances satisfies his or her needs. As the user changes the relative importance of primitives by adjusting a set of weighting factors (at query time), the Virage system dynamically recomputes the overall similarity between a single query image and the full set of target images.

Domain Events

The VIMSYS event model relates to time-dependent media such as digital video, as well as image sequences such as mamograms of the same individual over time. Time-dependent features are things such as object motion, object discontinuities, scene breaks or cuts, and (particularly in the case of video) editing features such as dissolves, fades, and wipes.

The information model described above is central to the architecture of the Virage technology. All other aspects such as the keywords associated with images, the exact nature of data management and so forth are somewhat secondary and depend on the application environments in which the technology is used. In the following section, the software aspect of this core technology is explained. This is followed by an explanation of the different environments in which the core model is embedded.

The VIR Image Engine



Virage technology is built around a core module called the Virage Engine and operates at the Image Object Level of the Virage model. There are three main functional parts of the Engine: Image Analysis, Image Comparison, and Management. These are invoked by an application developer. Typically, an application developer accesses them during image insertion, image query, and image requery (a query with the same image but with a different set of weighting factors). The following section describes the function of each unit, and how the application developer uses the Virage Application Programmer's Interface (API) to exchange information with the VIR Image Engine. The full capabilities of the Engine are decomposed into two API

sets: the base Engine, and the Extensible VIR Image Engine. The base Engine provides a fixed set of primitives (color, texture, structure, etc.) while the Extensible Engine provides a set of mechanisms for defining and installing new primitives (discussed in detail later).

Image Analysis

The Image Analysis functions perform several preprocessing operations, such as smoothing and contrast enhancement, to make the image ready for different primitive-extraction routines. Each primitive-extraction routine takes a preprocessed image, and, depending on the properties of the image, computes a specific set of data for that primitive. A vector of the computed primitive data is stored in a proprietary data structure. The application simply hands the Engine a raw image buffer, and the Engine returns a pointer to a set of data containing the extracted primitive data.

The application is then responsible for storing and managing the data in a persistent fashion. The Engine operates in a "stateless" fashion (akin to Web Serving), which means it has no knowledge of how the image data is organized and stored, or how the results of queries are managed. There is no transaction management at the Engine API level. This property means that system developers and integrators need not worry about conflicts between the Virage Engine and other application components such as databases, client-server middleware, etc.

Comparisons

There are several ways to compare images using the API. Each method involves computing one or more similarity distances for a pair of primitive vectors. The computation of the similarity distance is performed in two steps. First, for each primitive such as "structure" or "global color," a similarity distance is computed. These are then combined with weights by a judiciously cho-

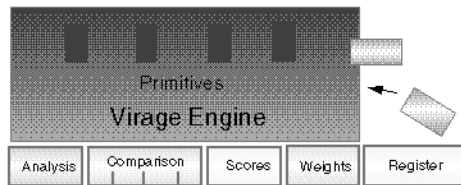
sen distance function that forms a final score that is used to rank results by similarity. Of course, the definition of “similarity” at this point is determined by the set of weights used. Applications may also synthesize a property weighting (such as “composition”) by intelligently applying weights during comparisons. If “composition” is weighted low, then global primitives should be emphasized; if it is weighted high, then local primitives should be emphasized.

In order to get the result of an image comparison, the application supplies the precomputed primitive vectors from two images, together with a set of weights. The system fills in the score data structure and returns a pointer to the caller. The score structure from a comparison can be used to efficiently recompute a new score for a different set of weights (but the same query image, in other words, a re-query). An additional API call will take a score structure and a weights structure and recompute a final score (ranking) without needing to recompute the individual primitive similarities. A third API call is an extension of the first, in that the user also supplies a threshold value for the score. Any image having a distance greater than this value is considered non-qualifying, which can result in significant performance gains since it will probably not be necessary to compute similarity for all primitives.

Management

There are several supporting functions that fall in the category of “management.” These include initialization, allocation and de-allocation of weights and scores structures, and management of primitive vector data.

The Extensible VIR Image Engine



The purpose of the Extensible Engine is to provide to the application developer the flexibility of creating and adding custom-made primitives to the system. There are three important aspects of this type of development. The first is the notion of a schema of primitives, which constitute the overall visual matching mechanism for the application being developed. Applications using the Extended Engine may specify a schema of primitives that is tailored to a specific application; for example, grayscale images don't require color primitives. The second is the definition of custom primitives and incorporating them into a schema. The third is a set of image-processing support tools to assist in easily developing new primitives.

In order to define a new primitive, the developer supplies custom functions to:

- compute the data associated with the extracted features for the primitive
- compute the distance between two sets of feature data previously extracted
- perform a byte swap of the feature data for endian management
- print the values of the primitive for debugging purposes

These functions are then registered with the system and associated with a primitive ID tag. From there, it can be incorporated into any schema definition by referencing the ID tag just like a built-in primitive.

Graphical User Interfaces

In addition to the programmer's interface, Virage also has a fully operational set of GUI tools necessary to develop a complete application. These includes facilities for image insertion, image query, weight-adjustment tools for requery, dialog boxes for creating field names for meta-data, inclusion of keywords, and support for several popular image file formats.

Query Canvas

The Query Canvas is a specific user-interface mechanism that is an enhancement to the query-specification environment. It consists of a bitmap editor in which the user can sketch a picture with drawing tools and color it using a color-selection palette. In addition, the user can drag and drop an image from an existing collection onto the canvas and modify it using the same drawing tools. Once an image has been created, it can be submitted as a query to the system. This tool saves the user significant initial browsing time in those cases where he or she already has an idea of what the target images should look like. Since the query canvas allows modification of images, it encompasses the functionality of the "query-by-sketch" paradigm.

Light Table

As a workgroup support tool for visual information retrieval among desktop publishers, Virage has incorporated a sharable workspace called a Light Table. During a query session, selected output from a query result can be placed in a Light Table. Once in the Light Table, they can be arranged in groups for inspection under various background illuminations, and can be annotated for opinion sharing in the group. Light Tables can be sent around by e-mail (MIME) or put on a floppy disk.

Applications

The VIR Image Engine directly implements the Visual Information Model previously described and acts as the hub around which all specific applications are constructed. The Engine serves as a central visual information retrieval service that fits into a wide range of products and applications. The Engine has been designed to allow easy development of both horizontal and vertical applications.

Vertical Applications

Because the facility of content-based image retrieval is generic, there is a large potential for developing the Virage Technology in several vertical application areas, such as:

- digital studio
- document management for offices
- digital libraries
- electronic publishing
- face matching for law enforcement agencies
- radiological information systems
- environmental image analysis
- on-line shopping
- trademark searching
- Internet publishing and searching
- remotely sensed image management for defense

To explain why the Virage Technology is a central element in these applications, let us consider some application possibilities in detail.

Environmental Imaging

Environmental scientists deal with a very large number of images. Agencies such as NASA produce numerous satellite images containing environmental information. As a specific example, the San Diego Bay Environmental Data Repository is geared towards an . . .

“. . . understanding of the complex physical, biological and chemical processes at work in the Bay . . . it is possible to correlate these different kinds of data in both space and time and to present the data in a visual form resulting in a more complete picture of what is and what is not known about the Bay. . . . This is the kind of information that is required to assist decision makers in allocating scarce resources in more effective and informative monitoring programs by sharing data, eliminating redundant monitoring and reallocating resources to more useful and effective purposes. Another key component of this work is to provide all of these data and resultant analyses to the public-at-large . . . through the World-Wide-Web of the Internet.”¹

For such applications, the methods are applicable to any geographic area in the world. Many of the datasets for environmental information are in the form of directly captured or computer-rendered images, which depict natural (mostly geological) processes, their spatial distribution, and time progression of measurands. It is a common practice for environmental scientists to search for similar conditions around the globe, which amounts to searching for similar images.

Medical

A significant amount of effort is being spent in nation-wide health care programs for early detection of cancer. Image comparison is one of the fundamental methods for detecting suspicious regions in a medical image. Specifically, consider a cancer-screening center where a large number of fine needle aspiration cytology (FNAC) tests are conducted daily for breast cancer. We can envision a system that uses Virage’s image-similarity techniques to provide an intelligent screening aid for the practicing cytologist. After the slide is prepared, it is scanned by a camera-equipped microscope at different levels of magnification. At each magnification level, the slide is compared to a database of other slides (or an existing pre-annotated atlas) at the same magnification, and similarity is computed in terms of cell density, number of nuclei, shapes of nuclei, and number of dividing cells. Suspicious regions of the slide are presented to the cytologist for closer inspection. If nothing suspicious is found, the system might suggest skipping the next higher level of magnification. The cytologist could always override the suggestion, but in general, it would save the cytologist the tedium of scanning through the entire slide, and thus increase his or her productivity.

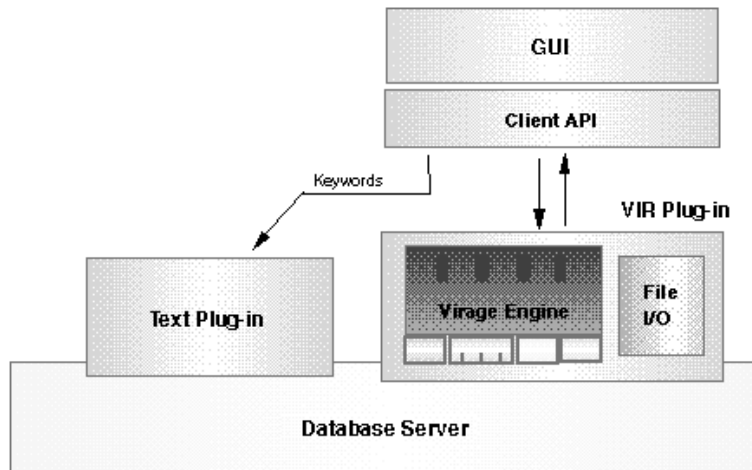
Multimedia

Digital libraries of videos are becoming common due to the large number of sports, news, and entertainment videos produced daily. Searching capabilities for a video library should allow queries such as “show other videos having sequences like this one.” If the query sequence has a car chase in it, the system should retrieve all videos with similar scenes and make them available to the user for replay. The basic technology to achieve this relies on detection of edit points (cuts, fade-ins, and dissolve), camera movements (pan and zoom), and characterizing a segmented sub-sequence in terms of its motion properties. Also needed is a smooth integration with a database system containing textual information (such as the cast, director, and shooting locations), and other library facilities for which software products already exist.

1. From the San Diego Bay Project home page at http://www.sdsc.edu/SDSC/Research/Comp_Bio/sdbay/sdbay.html

Application Integration

The Virage Engine naturally accommodates various types of databases and application frameworks. In these scenarios, the Virage Engine can be licensed to other system integrators and application developers to enrich their technology by including Virage's image-searching capabilities. Typical applications could be:



- Inserting the engine into the framework of popular database products such as Oracle, Informix, and other databases built to support multimedia. The engine can be integrated to accommodate both direct and indirect data-access models.
- An image-manipulation and processing tool like Photoshop from Adobe, Image Manager from Microsoft, or CorelDraw from Corel could use the Virage Engine for image search and management.
- An Internet crawler and search engine like WebCrawler, InfoSeek and Lycos can extend their capabilities with image finding in the network and thus help build a searchable image repository distributed over the network.

Each of these Virage-enriched applications have further growth potential in several vertical markets.

Future Directions

Traversing the path from the current state of visual information retrieval to full visual information management requires active research and development of new technology. The open architecture strategy latent in the design of the VIR Image Engine will allow it to interface with several kinds of software products. In this section, we briefly discuss some of the directions Virage is pursuing in achieving this goal.

Video and Multimedia Information

Virage has all the technological expertise to apply this technology to video information retrieval and expects to develop it fully in the near future. The focus of the video effort will be in providing basic video-manipulation tools for segmentation, storage, and retrieval based on scene breaks. Methods for fast preview, retrieval, timeline editing, and updating will be developed using temporal variation of low-level image properties, including motion characteristics of objects in the video. A key element of the technology is to derive index mechanisms and similarity metrics to capture the temporal change of visual features. We are exploring the market for applications of this technology, such as building video annotation systems, integration with digital video production systems, and providing an iterative query mechanism for video databases.

Domain Definition Mechanism

Recall the Domain Object Layer in the VIMSYS model, which referred to features meaningful to the users. In many vertical applications, such as medical imaging or facial image matching



for surveillance, the user has a well-defined model of the features that are of interest. These features are typically sub-objects present in the image (like an artery in an image of the cardiothoracic region), or they

can be features that relate to distributional properties of image features (for example, this shade of green with that texture designates a certain farming practice). In such applications, a visual information retrieval system needs a method to specify the domain objects in terms of features computed as image objects. To incorporate these specifications into the fold of visual information retrieval, Virage is planning to develop a set of parametrically specifiable primitives, a set of domain-specific primitives, and a domain-specific mechanism of constructing domain-specific objects using these primitives.

Query Specification Mechanisms

In cases where the user has a clearer specification of the visual attributes or arrangements to be retrieved, more expressive queries can be formed. Area range restrictions can be imposed on image regions (for example, all green regions should have an area between A1 and A2), or Boolean combination of spatial attributes can be mentioned (for example, the red circles could be here or here, but not there).

Equally important is the extension of our current Query Canvas as a more general re-query mechanism. In this extension we would like the user to edit a query for each feature, such as color, texture, and structure. We are exploring what kind of analysis tools need to be provided to the user to make this re-query mechanism most effective.

Virage also plans to have a more intelligent means of handling image browsing operations by using the user's viewing history. This will increase the user's search efficiency (mean time from random to focus) for very large image collections. We also expect to provide a query-time feature-definition facility for advanced users. With this facility, the user will be able to define a feature on the fly while making a query. This new feature will be computed by the system, used for the current query session and added to the system's built-in set of features for future use.

Conclusion

The visual sense of humans is a powerful system in terms of any quantifiable dimension you might choose: color discrimination, spatial resolution, temporal response, overall bandwidth, etc. It therefore makes sense that computers should interact with humans in a way that takes advantage of this power. Visual techniques for navigation and processing of information will undoubtedly be the foundation for information systems in the future. Virage's visual search technology is a major step in this direction.

Until recently, database and imaging technologies were separate islands serving different needs in separate application areas. Now they are each expanding in scope, merging and overlapping. The intersection of these technologies is creating a new category of applications. The Virage Engine represents a concrete and viable tool to begin constructing these new applications today. It has broad horizontal capabilities to address diverse needs in imaging and visual management. Its interface is flexible and extensible. It is the first of several steps to be taken by Virage as we deliver yet more powerful tools to application developers to enable entirely new application areas in visual information management.