I^2*Cnet*: Content-Based Similarity Search in Geographically Distributed Repositories of Medical Images

Stelios C. Orphanoudakis, Catherine E. Chronaki, and Despina Vamvaka

Abstract

The retrieval of images by content is rapidly gaining acceptance as an important function of image database systems. This paper discusses the architecture of I^2Cnet, a network of servers which provide content-based query services through a WWW browser. In I^2Cnet, algorithms for the representation, storage, and retrieval of medical images based on different descriptions of image content are implemented using description types. AttributeMatch, a description type supported by I^2Cnet, aims to capture the knowledge of medical experts in queries by using a similarity criterion which can be tailored to user preferences. We present results showing the query response time of AttributeMatch, obtained with image classes of various sizes, and the degree of similarity of retrieved images to the query image under different similarity criteria.

Key Words: content-based image retrieval, medical image databases, visual information management, teleradiology, regional health care network, WWW information services

1. Introduction

Current research on visual information systems and multimedia databases raises a number of important issues, including the need for query methods which support retrieval of images and video by content [1-3].

At the same time, the rapid growth of popularity enjoyed by the World Wide Web during the last few years, due to its visual nature and information retrieval capabilities, has directed research efforts towards the development of systems that provide network-transparent information services based on pictorial content [4]. In this vast, dynamic information infrastructure, the development of medical information systems with advanced browsing and navigation capabilities and a visual query language supporting content-based similarity queries will play an increasingly important role in medical training, research, and clinical decision making.

The architecture of I^2*Cnet* (Image Indexing by Content network), discussed in section 3, addresses these issues by providing content-based retrieval as an added value service in a regional health care network. The main elements of the I^2*Cnet* architecture are: I^2*C* clients, I^2*C* server brokers, and I^2*C* servers. I^2*C* clients use a standard WWW browser to request I^2*C* services and to submit content-based similarity queries. I^2*C* service brokers activate software agents to update the profile of available services and to provide support for network-transparent queries. I^2*C* servers maintain databases of image content descriptions and interact with the health care network to retrieve additional information on selected images and respond to queries which involve image content and other electronic patient record data. I^2*Cnet* extends the functionality of I^2*C*, an existing information system which can serve as a browser for images and image descriptions, an editor of image content descriptions, and a processor of content-based queries (see section 2).

In I^2*Cnet*, description types encapsulate methods for the representation, storage, and retrieval of medical images based on their pictorial content. As a case study of a description type supported by I^2*Cnet*, section 4 describes *AttributeMatch*, an attributebased description type which attempts to capture and express the knowledge of medical experts by allowing them to construct a similarity criterion which emphasizes clinically relevant features of image content. Thus, medical images may be matched to the query image based on different types of similarity, ranging from strict structural similarity to more elastic clinical similarity. The derivation of an image description requires that the user specify the Regions Of Interest (ROI) for which a standard set of geometric properties are computed, and a subset of these ROIs for which texture descriptors are also computed. As part of the query, the user specifies the relative significance of ROIs and ROI features. These significance factors are used in the similarity criterion to focus the attention of the search engine on selected features of the ROIs. At present, we do not use imaging modality parameters or quantitative tissue characteristic properties, although their inclusion would be a straightforward extension of our current approach. A possible problem with including such properties is the need for calibration, so that they can be compared.

Performance results, which include response time and the relative similarity of images retrieved using different similarity criteria in *AttributeMatch*, are presented in section 4.3. Section 5 places our work in the context of medical image databases and visual information management systems, discussing related work on the content-based management of image and video data. Finally, conclusions, the status of the project, and future plans are presented in section 6.

2. I^2C : Image Indexing by Content

2.1 Overview

I^2C [5] is an image management system which has been developed as part of an ongoing project for the design, implementation, and evaluation of approaches to image content representation and content-based similarity retrieval strategies. Different approaches are implemented and investigated using *description types*. Furthermore, a user-defined image class hierarchy helps the user direct queries to the appropriate image set, thus increasing the efficiency of retrievals through search space reduction. Machine learning techniques, which would identify the class an image belongs to and automatically direct the query to it, are also being investigated.

The architecture of I^2C has been developed based on the principles of generality, modularity, extensibility, open architecture, and conformance to standards. Its open architecture permits the on-line incorporation of new algorithms to the I^2C toolbox. System conformance to standards, e.g. DICOM, facilitates the interoperability of I^2C with PACS and other medical information systems, thus providing an added value service in the medical application domain.

Users interact with I^2C through a graphical user interface (see Fig. 1) to an extensible toolbox of image processing and editing tools to generate, archive, match, and browse through images and their content descriptions effectively and efficiently. The I^2C database engine maintains I^2C persistent objects: algorithms, description types, image classes. The use of object-oriented techniques in the I^2C database engine facilitates flexible programming of associations among image classes and algorithms, while the use of low level database access primitives achieves low response time. Thus, the I^2C database engine provides both efficiency and flexible programming.

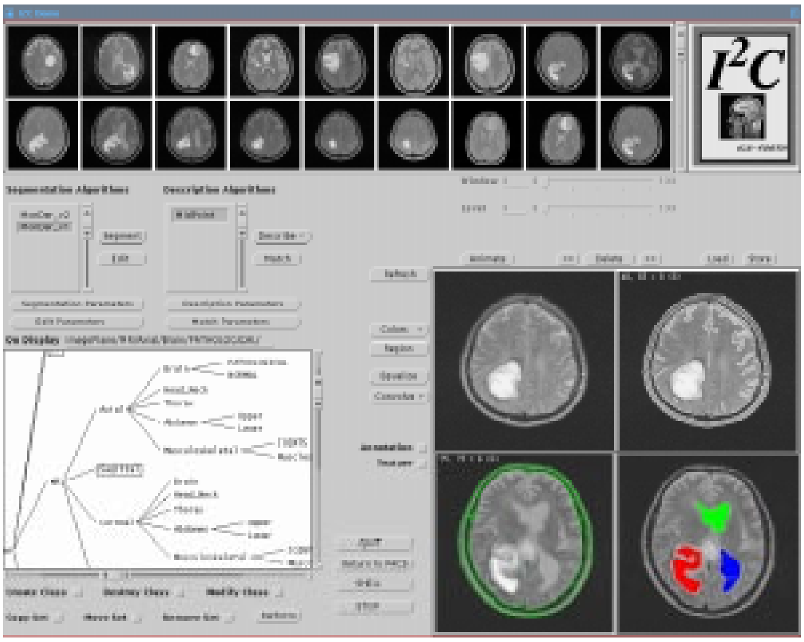


Figure 1: I^2C user interface: image & description browser (top), algorithm workbench (middle left), class editor/browser (bottom left), display window (bottom right)

2.2 Description Types

Description types are used in I^2C to encapsulate information relevant to an image content representation method and a content-based retrieval strategy. The functional components of a description type are: the *description generator*, the *description matcher*, and the *description manager*. The description generator is involved in the extraction of image features and the generation of an image description as a set of persistent objects. The description manager maintains the repositories of the description type. It is responsible for the insertion, deletion, and modification of the objects that comprise such content descriptions, with concurrent update of the indices involved.

The I^2C database engine provides persistence primitives that can be used by the description manager, if the repositories of the description type are stored in EXODUS [6], the native storage manager of I^2C. Alternatively, the description manager may employ an arbitrary persistence method (or methods), possibly exploiting data distribution and parallel search strategies. The description matcher processes content-based queries addressed to the description type and identifies images similar to the query image. Moreover, if the state of the description matcher is maintained in the persistent store between executions of the program, the description type is able to handle incremental query modifications.

2.3 Image Class Hierarchy

All actions and processes supported by I^2C are based on the organization of images and image-related information into classes representing different imaging modalities and parts of the anatomy. Thus, algorithms used to obtain descriptions of image content can be tuned for a specific class of images (e.g. MR brain images). Algorithms tuned for a specific class are likely to yield far better results for images in the current class than general-purpose algorithms, since they can exploit knowledge regarding the characteristics of images in the class. Furthermore, the efficiency of retrievals can increase by directing content-based queries to appropriate classes.

Images are assigned to classes using primary criteria, such as imaging modality, part of the anatomy, orientation, plane of cut, etc. and secondary ones. Secondary criteria of class membership may be derived from the clinical interpretation of image content or determined by a machine learning algorithm based on quantitative image features. At system installation, an initial image class hierarchy is established in collaboration with health care professionals. Additional image classes or subclasses can be created at a later time, so that images for which specific description types have become available can be assigned to a separate class.

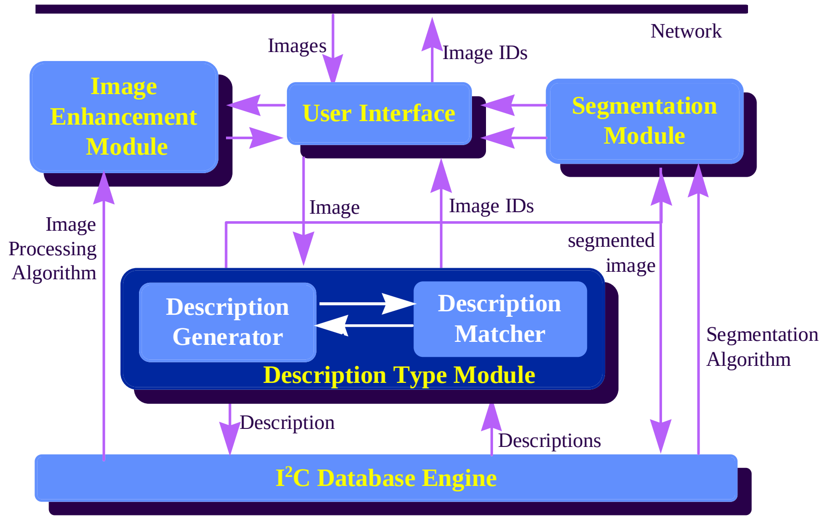


Figure 2: I^2C Architecture

2.4 Architecture

The architecture of I^2C is object-oriented and adheres to the principles of extensibility and modularity. As shown in Fig. 2, the system consists of a small set of communicating modules: the *image enhancement module*, the *segmentation module*, the *database engine*, and the *description type module*. All modules exchange information through the I^2C core, which handles messages and links I^2C to image archives through information gateways. I^2C has been designed to be independent of specific notions and representations of image content, and does not assume a specific image similarity criterion. Image features, which have been computed using tools of I^2C, are encapsulated in the intermediate image description object and subsequently communicated to description types. New features and alternative algorithms for their computation can be supported through appropriate extensions to the intermediate image description object.

Currently, at a system administration level, specific programs facilitate the integration of new image enhancement and segmentation algorithms, as well as description types. Dynamic associations of algorithms and classes are also programmed at this level. Thus, a user of I^2C has the option of creating a personal image description repository and developing his own variety of image description types. This can be accomplished either by tuning the algorithms provided by I^2C and their similarity criterion, or by developing new description types and using I^2C as a query front end and a description browser.

2.5 Query Formulation

Two types of query are currently supported by I^2C: *query by example* and *query by sketch*. In each case, the user specifies the image class and a description type based on which the query will be served. In general, query by example may be automatic or interactive. In automatic query by example, the user requests the retrieval of images similar to the image in the main display window (lower right quadrant in Fig. 1), an image description is generated automatically, and a set of similar images is returned. Alternatively, in interactive query by example, the user guides the image content description process. In query by sketch, the user enters the contour editor environment and sketches the boundaries of anatomical objects, which will form the basis for the description and the retrieval of images by content.

The I^2C contour editor environment, shown in Fig. 3, occupies the area of the display window drawing area) and class browser (display of ROI features). This environment aids the user in the creation of image content descriptions and the composition of queries. To achieve this goal, the user may combine the results of image segmentation algorithms to create, store, and retrieve anatomical contours. Texture descriptors and significance factors may also be associated with ROIs contained within particular contours.

2.6 Database Engine

The I^2C database engine manages the persistent I^2C objects which keep information on image classes, algorithms, description types, images, intermediate image description objects, and their associations. The association of an image class to a segmentation algorithm indicates that the specific algorithm has been tuned for that class. A specialized object-oriented interface abstracts the details of the adopted persistence mechanism and allows the dynamic creation and maintenance of associations. Moreover, the I^2C database engine provides primitives for the management of object clusters and indices, which can be used as the building blocks in the development of data repositories for description types.

The I^2C database engine has been implemented as a client process on top of the EXODUS storage object manager [6]. EXODUS is an extensible database system indented to simplify the development of high performance application-specific database systems. According to the OO7 benchmark, EXODUS rates highly among existing OODBMS [7]. The EXODUS storage manager client library is linked to the I^2C database engine module and provides a procedural interface to primitives for atomicity, indexing, versioning, and management of large untyped objects. The EXODUS client module maintains its own buffer pool on the local machine, where it caches, prefetches, and updates pages containing objects and indices, instead of simply forwarding requests to the EXODUS server. The resulting object-oriented database engine, being a dedicated application, is quite efficient in comparison to a full fledged object-oriented DBMS.

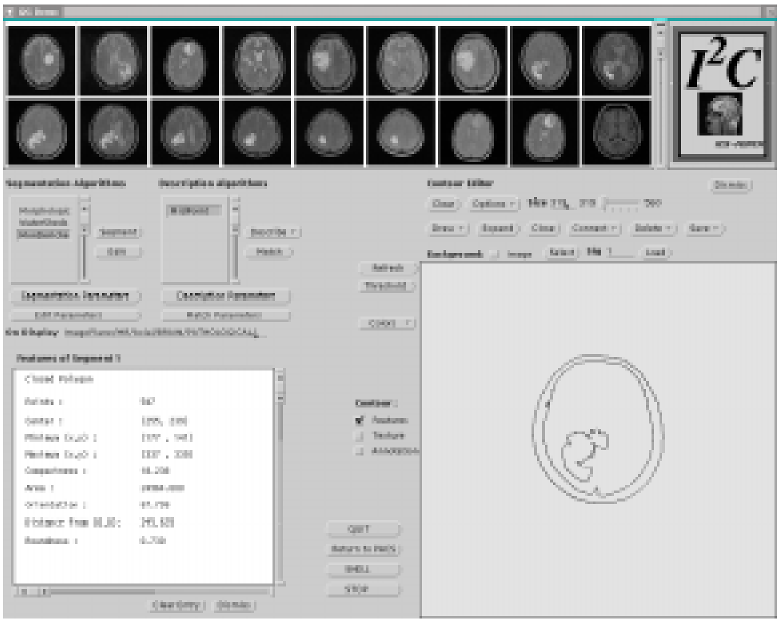


Figure 3: Contour editor environment of I^2C. This environment is used to edit the results of segmentation algorithms (bottom right), to draw ROIs in query by sketch (bottom right), and view the values of computed ROI features (bottom left).

2.7 Discussion

The I^2C prototype has been operational for the last three years. During this time, its toolbox has been enriched with new feature extraction tools and algorithms for image enhancement and segmentation. Various description types have been designed, implemented, and evaluated for different classes of medical images. Furthermore, I^2C has been integrated with TelePACS [8], a home grown Picture Archiving and Communication System, which has been installed in three large hospitals and is currently undergoing clinical evaluation. Preliminary results indicate that I^2C can be an effective added-value service in PACS and telemedicine environments, particularly as a clinical decision-support tool and a tool for medical training and clinical research. It is also a useful platform for the implementation and evaluation of algorithms for image content description and content-based image retrieval strategies.

I^2C has been designed to be interoperable with image archives which conform to appropriate standards. Therefore, in addition to its use as an added-value service in a PACS environment, it can be interfaced to an image archive and used as a standalone system to manage private image collections. However, the cooperation among I^2C systems is an aspect of its functionality that was not initially considered. Therefore, the design of *I^2Cnet* as a network of I^2C servers brought to light several limitations of the I^2C architecture and built-in functionalities. First, query formulation, processing, and optimization were heavily dependent on NFS running in the local network. Second, the I^2C user interface was closely coupled to the rest of the system and no attempt was made to minimize the transfer of information between the database engine and the user interface. Third, in a distributed environment, the construction of the I^2C database server as an EXODUS client process is less flexible than extending the database server itself, what SHORE [9] calls a “value added” server.

To overcome these limitations, I^2Cnet has been designed based on a loosely-coupled architecture of I^2C clients and servers, as presented below.

3. *I^2Cnet:* A network of I^2C servers

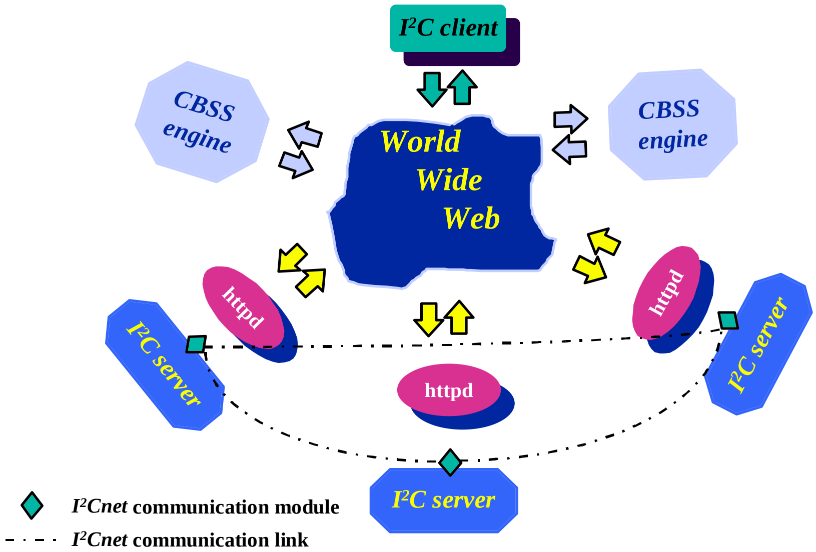


Figure 4: I^2Cnet: A network of I^2C servers designed to operate over the World Wide Web and to provide content-based similarity retrieval services to the regional health care network. Other content-based similarity search engines (CBSS) available in the World Wide Web are accessible through a unified interface.

3.1 Overview

With the evolution of the information superhighway and the involvement of our group in the development of a regional health care network [10], the existence of communicating image repositories that evolve in a decentralized fashion establishes content-based similarity retrieval as an added-value service.

I^2Cnet has been designed as a network of image description servers based on I^2C and operating over the World Wide Web (see Fig. 4). Through a WWW browser, I^2C clients will enjoy reliable and network-transparent content-based access to distributed image repositories. A major advantage of this approach is that other services with a similar scope, including third party content-based similarity retrieval services like those of QBIC [11], can be accessible through a unified framework offered by HTML. The combination of standard HTML and browser programming offers a generalized interface to heterogeneous types of data, and provides advanced, platform independent, client-server interaction.

The I^2Cnet approach is based on multiple levels of service quality. Minimally, a user with a typical web browser can interact with the system. However, several add-ons like the I^2C service broker allow for more sophisticated interaction with the network, by introducing a certain level of network transparency and intelligence in this interaction. In general, two types of user interaction with I^2Cnet have been considered:

- *Network-transparent* interaction, in which the user formulates a request without any concern as to which server or servers will process it.

- *Server-specific* interaction, in which the user specifies how the request should be handled.

Although typical interaction of a user with I^2*Cnet* involves a content-based query which is addressed to multiple I^2*C* servers, I^2*Cnet* offers additional services such as image posting, i.e. making an image available to I^2*C* servers for description, or requesting the execution of an image processing algorithm. To support such interactions, I^2*C* servers should share information even if they operate and evolve independently. This shared information refers to image classes, description types, and processing algorithms supported by each server. The set of services offered by each server changes dynamically over time. New I^2*C* services become available and old ones cease to exist. Furthermore, based on user preferences, some services are of higher quality and reliability than others. I^2*C* service brokers are active entities in I^2*Cnet* which maintain profiles of use and service availability and adapt their behavior based on user feedback.

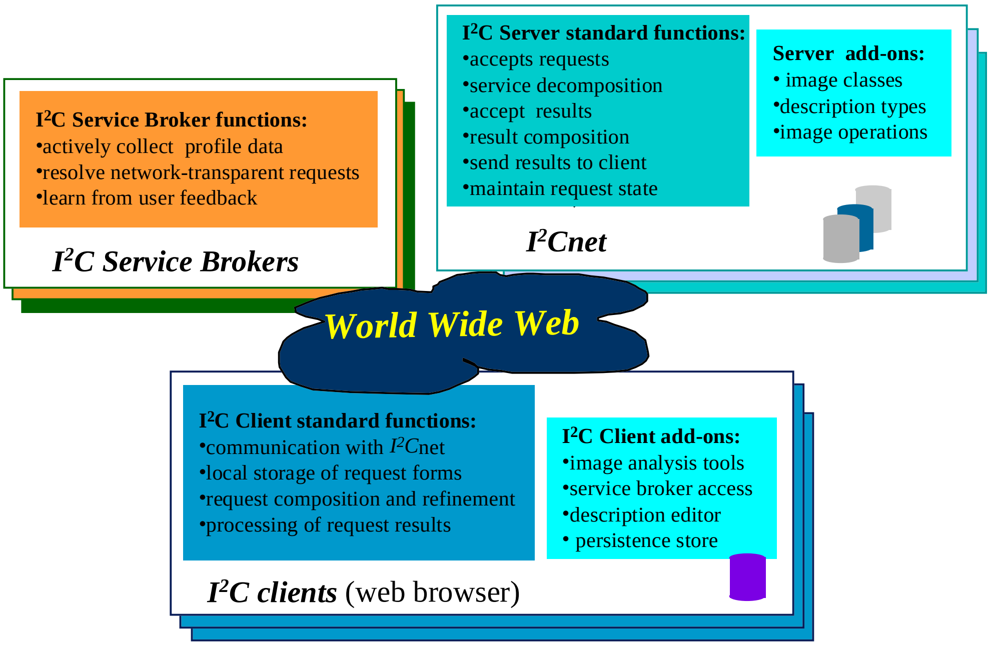


Figure 5: I^2*Cnet* architecture.

3.2 Elements of the I^2*Cnet* architecture

The design of the I^2*Cnet* architecture brought out several new issues and challenges. Peer-to-peer server communication, agent based computing, query formulation and incremental query modification, query decomposition and result composition, integration of heterogeneous systems over a regional network to achieve quality of service and security, are only a few of them. Fig. 5 shows the basic elements of the I^2*Cnet* architecture and their functions: I^2*C* servers, I^2*C* clients, and I^2*C* service brokers.

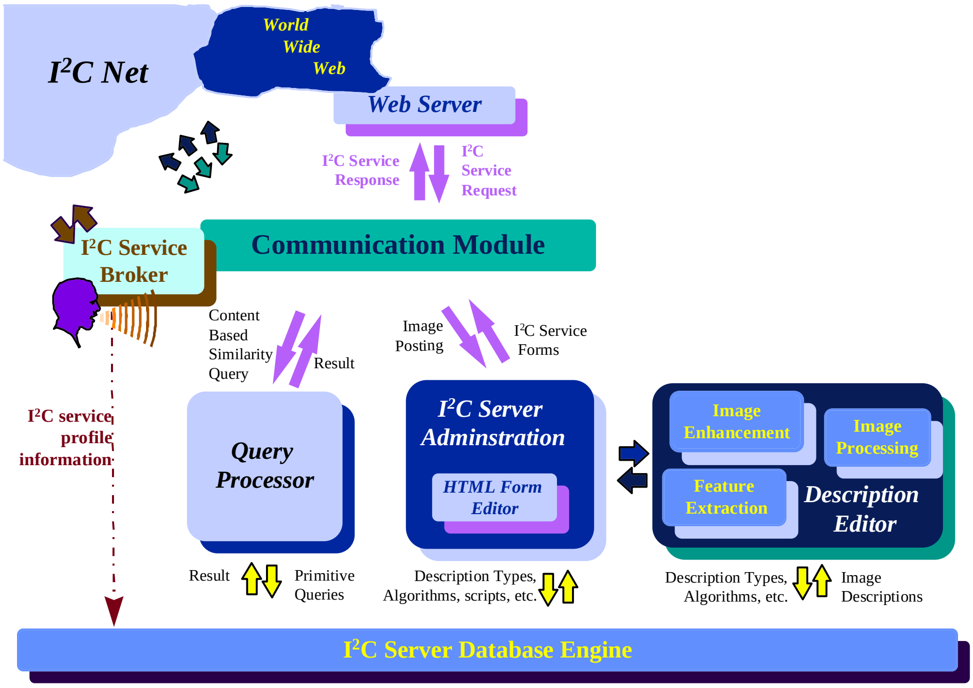


Figure 6: *I²C servers* maintain autonomous repositories of image descriptions and cooperate with service brokers to respond to network-transparent queries.

3.2.1 I²C servers

I²C servers are linked to image repositories, manage image content descriptions, cooperate to provide federated *I²C* services, and interoperate with the health care network. The functional modules of the server architecture shown in Fig. 6, are the *description editor*, the *administration front end*, the *communication module*, and the *query processor*.

The description editor evolved out of the contour editor environment of *I²C*. In *I²Cnet*, the description editor is part of the *I²C* server architecture, and an optional add-on to *I²C* clients. Using the description editor, users are able to generate image content descriptions by extracting image features with a wide variety of image processing tools. In addition, the description editor may be used to compose queries by sketch. An important feature of the description editor is its extensibility; it provides a plug-in interface for new image enhancement, segmentation, and feature extraction algorithms.

In *I²C*, database administration was carried out through a collection of application programs. Administration of the *I²C* database involves the maintenance of associations between image classes, algorithms, and description types, the replenishment of description databases with new images, the introduction of new algorithms, description types, and classes, and the scheduling of time-consuming off-line jobs such as segmenting all images in a class or locating all pairs of most similar images in an image class. In *I²Cnet*, the scope of the administration module is broader. Each description type supported by the server has a home page with appropriate links that guides the user through the process of submitting a query. The maintenance of these pages is performed by the administration module. In addition, messages regarding the posting of images arrive from the network and are logged for the administration personnel to review.

The communication module facilitates the connection of the server to the local web server, other I^2C server communication modules, and I^2C service brokers. The local web server forwards to the communication module service requests from I^2C clients throughout the network. The interconnection of communication modules is necessary when I^2C servers cooperate to service requests that involve multiple servers. Finally, communication with service brokers involves the resolution of network-transparent queries. The service broker translates the network-transparent query into a server-specific one. The communication module maintains the state of pending queries, service records of past requests, and usage statistics.

The query processor handles service requests addressed to the specific server. A service request is decomposed, its parts are executed, the results are composed, and a service record is prepared. In case of a content-based query, the retrieval record includes a set of imageIDs, hyperlinks, and scores reflecting confidence in the result. ImageIDs point to thumbnail miniatures of images. Hyperlinks point to the original image in the regional network. The above information is inserted into an HTML template form and the resulting page is returned to the communication module.

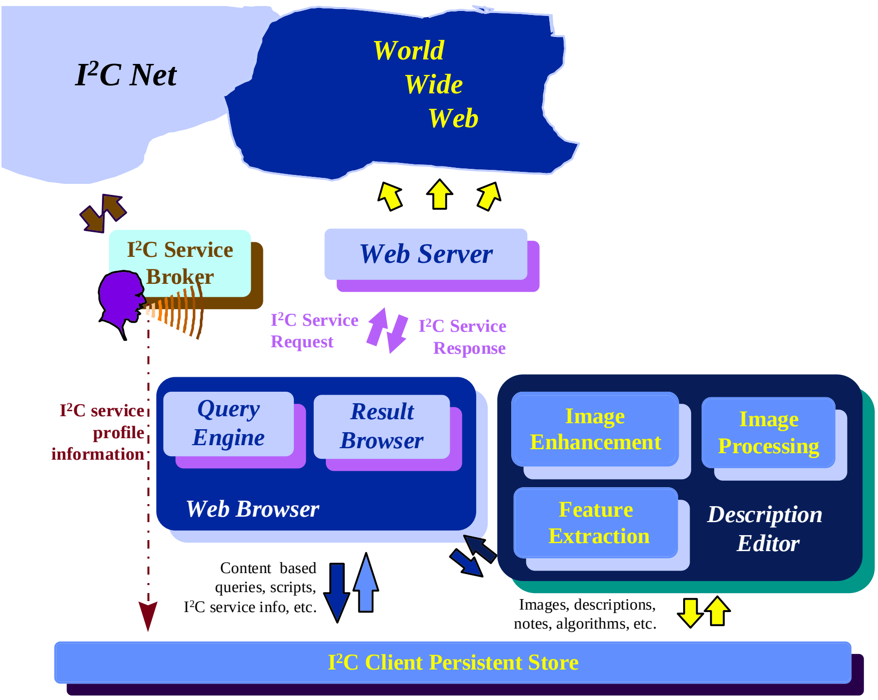


Figure 7: A minimal I^2C client is a typical web browser. Optional add-ons like the *Description Editor*, and the *Service Broker* allow for more sophisticated interaction with *I^2Cnet.*

3.2.2 I^2C clients

The I^2C client architecture shown in Fig. 7, presents the functional components of the client, some of which are part of a typical web browser. A web browser serves both as a *query engine* and a *result browser*, since the user is able to browse through services provided by *I^2Cnet* in a host *I^2C* server and submit server-specific or network-transparent queries.

For advanced user interaction with *I^2Cnet*, several optional add-ons can be considered as part of the *I^2C* client architecture: the *description editor*, a *service broker*, and a *client side persistent store*. The user may use the description editor to enhance the appearance of the query image, analyze its features, and create its description based on guidelines provided by the home pages of description types (on-line manual). An additional functionality which could be considered, is that of creating synthetic images, using shape and texture catalogs, as well as other tools of the description editor, and submitting them as an enhanced query by sketch. A personalized service broker helps the user to maintain query pages which are tailored to his (her) preferences. The client side persistent store is used by the description editor to store information on algorithms, and texture and shape collections, and by the service broker to maintain profile information.

3.2.3 I^2C Service Broker

The administration of *I^2Cnet* requires the maintenance of a service profile for each *I^2C* server. This profile includes loosely consistent information on *I^2C* service availability: e.g. image classes, processing algorithms, and description types. The image class hierarchy provides a common frame of reference for *I^2C* services provided by different servers. The main component of the image class hierarchy is present on all servers hosting default image classes, processing algorithms, and description types. Additional image classes, description types, and processing algorithms may be developed locally and be exported as *I^2C* services. This results in a dynamically evolving and to some degree heterogeneous environment.

In this environment, *I^2C* service brokers maintain loose consistency and provide network-transparency. The *I^2C* service broker activates agents to actively explore the network, collecting information on the available servers and services. These agents query *I^2C* servers for updates in the set of available services, statistics of use, and user feedback on provided services. Specifically, the service broker maintains a directory of available services throughout *I^2Cnet*, *server profiles* which reflect overall server performance, *user profiles* which reflect user preferences and information on the client platform, and *service profiles* which reflect availability and quality of particular *I^2C* services.

In the case of network-transparent services, the user specifies the requested service (*what*), and the broker employs profile information to resolve the query (*how*). Stored profiles on servers, users, and services are used to select the most appropriate set of servers and description types, and translate the network-transparent query into a server-specific one. To improve the overall quality of the provided services, the user is prompted to provide feedback on the quality of the services provided. This information is taken into account in the profiles maintained by the service broker and affects its future behavior with respect to the way network-transparent queries are resolved.

To update profile information, *I^2C* brokers activate software agents. Software agents are programs capable of autonomous goal-oriented behavior in a heterogeneous computing environment [12]. They are currently used to actively gather and supply various types of information available on the Internet [13,14].

3.3 User Interface Issues

The user interface of an I^2C client is a typical Web browser like Mosaic, Netscape, etc. The user submits content-based queries by interacting with web pages (effectively filling HTML forms), and then browses the retrieved images.

Each description type supports a set of query types which correspond to web pages. We expect that the same web page will be compatible with multiple description types. These pages may be of arbitrary complexity, containing HTML forms with embedded script language and/or applets. For example, an advanced description type may allow the specification of a query in natural language. Currently, such advanced query forms require the communication of appropriate tools (e.g. voice recorder) with the browser. Since the whole purpose of WWW is to provide uniform access to heterogeneous sources of information, the only viable solution is to extend editing tools in a way that allows the transfer of data to and from the Web browser.

Currently, the description editor of I^2C is being modified to enable its cooperation with public domain Web browsers. Thus, a user formulating a service request to I^2*Cnet* will be able to drag and drop an image, a sketch, or even a voice recording into an HTML form. Furthermore, the same interface will enable those I^2C clients which are connected to the health care network to interoperate with dedicated medical information systems. For example, a doctor confronting a difficult case may look up images similar to the images in an electronic patient record by dragging them into his (her) favorite query page.

3.4 Communication among I^2C servers

When the user wishes to interact with I^2*Cnet* for the purpose of requesting an I^2C service, a connection is opened to a web server, local to the host I^2C server of the query. Then, the web server forwards the service request to the communication module of the host server. The communication module is responsible for decomposing the request, sending the subqueries to the communication modules of the I^2C servers involved, maintaining the state of the query, receiving for the results, and composing the response to the user.

Communication among I^2C servers is implemented through a message passing library based on ONC RPC [15]. Distributed object management approaches like CORBA [16], which provide a much higher level of abstraction, encapsulation, and flexibility through an interface language are investigated as the means of integrating I^2*Cnet* with the health care network. However, CORBA introduces additional overhead in the communication among I^2C servers which share the same interface without necessarily supporting the same services, which may be difficult to justify.

3.5 Query Language

The query types supported by I^2C will also be available in I^2*Cnet*, in the HTML framework. As far as specialized query types are concerned, it is the responsibility of the I^2C server that offers the corresponding query services, to make the appropriate web pages available. However, beyond these primitive query types, service requests that identify servers and services require the specification and development of an embedded query language. Thus, complex requests entailing multiple servers and description types can be expressed through logic operators and map functions.

A query response set consists of the images retrieved by description types at particular I^2C servers. The user may follow an image link to view image related data in the health care network. In addition, each query response is labeled as to the I^2C server and description type that produced it. The user is prompted to rate the relevance and overall quality of the specific response. User feedback is recorded in the service record and affects the translation of network-transparent queries. This permits I^2Cnet to evolve intelligently, since fast servers with rich data description repositories and accurate description types are likely to be used more frequently in network-transparent queries, while slow servers with unreliable description types become obsolete.

3.6 Discussion

In a regional health care network with multiple heterogeneous medical information systems, important issues are those of portability, security [17,18], and interoperability. The separation of the I^2C client (user interface) from the I^2C server has reduced portability problems to a great extent. I^2C was bound to XWindows and the SunOs operating system. The WWW interface of the I^2C *client* has made the portability of a client a much easier task, by disengaging the implementation platform of the user interface from that of the database engine.

Within a hospital PACS environment authorized users may access a patient record without violating security aspects. Thus, the use of I^2Cnet as a clinical support tool, provides authorized users with the capability to access patient records linked to retrieved images. In remote accesses, without proper user authorization, the approach taken in I^2Cnet is that of anonymity [19]. A user lacking authorization to access certain databases of patient records may still retrieve certain segments of a patient's record without knowing who that patient is.

The interoperability of I^2Cnet and the health care network, effectively introducing content-based similarity retrieval as an added value service, will be achieved through the use of the Patient Meta-Record (PMR) which integrates all patient related information in the regional network [20]. The PMR provides an authorized professional with the means to locate all health centers a patient has visited and access segments of the local patient records. Linking I^2Cnet to the archives of the regional network will allow an I^2C user to navigate the associated PMRs, starting from images retrieved as a response to a content-based query. At the technical level, the I^2C server database stores image identifiers which point to the source image archive. The combination of these identifiers and appropriate standards of information exchange, that facilitate information gateways among heterogeneous systems and state of the art caching techniques, form the basis for a regional network offering integrated services.

4. *AttributeMatch*: Content-based Retrieval of Medical Images

4.1 Image Content Description

The processes involved in the content-based management of images in I^2C are encapsulated in description types. In this section, we present *AttributeMatch*, one of the description types currently supported by I^2C. The representation of image content in *AttributeMatch* consists of geometric properties and texture descriptors of selected ROIs. Such features, differentiated with respect to their relative clinical significance, play an important role in comparisons of medical images routinely carried out by human experts. Furthermore, the similarity criteria applied by human experts are often quite subjective. The similarity criterion of *AttributeMatch* enables users to bias the search for similar images towards their own subjective notion of similarity by specifying the relative significance of ROIs and their computed features.

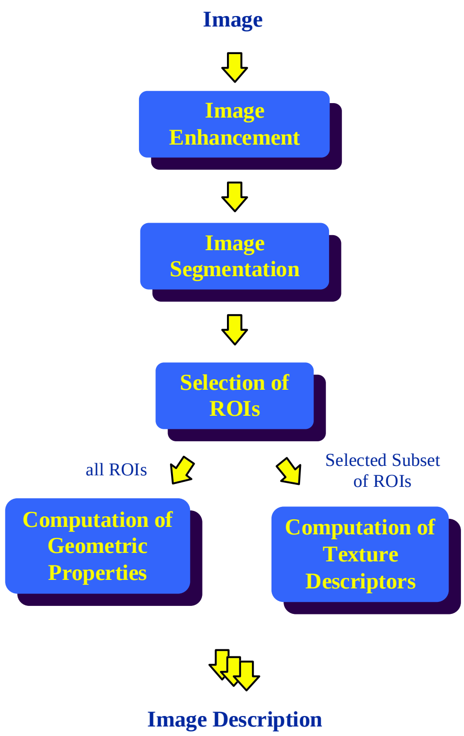


Figure 8: Image description generation process in *AttributeMatch*.

To extract descriptions of image content, the description editor of I^2C [5], used in interactive mode, permits editing and combining the results of different image processing modules. These include image enhancement, segmentation, and editing of segmentation results to obtain ROIs (see Fig. 8). The size of an image description and the speed of execution of a retrieval process clearly depend on the number of ROIs retained as being clinically important. For each ROI retained, the following set of features is computed: location, shape, size, and, only for a selected subset of these ROIs, a set of texture descriptors. Shape and size properties include roundness, compactness, area, and orientation. A number of different texture descriptors are computed based on gray-tone cooccurence matrices associated with a ROI (e.g. maximum probability, angular second moment, contrast, inverse difference moment, entropy, correlation, variance, cluster shade, diagonal moment, k statistics, and fractal signature) [21,22]. In the future, extensive experimentation and a more rigorous feature selection process will permit us to identify a reduced set of texture descriptors which constitute an adequate representation of image content.

4.2 Content-based Retrieval

In *AttributeMatch*, a content-based query consists of a description of image content I_Q, the relative significance factors w_r of ROIs, the relative significance factors w_f of ROI features, and the degree of dissimilarity or distance D_d of any region from a dummy region. Dummy regions are introduced when the images to be compared do not have the same number of ROIs. The default similarity criterion assigns equal significance to all ROIs and ROI features. The matching algorithm retrieves from the database images whose descriptions match the query description under the specific similarity criterion.

Two types of query are supported by *AttributeMatch*. Their difference lies in the type of user input and the level of user interaction in the generation of a description of image content:

- *Query by Example*: Query by example is *interactive*. Specifically, users are able to interact with the generation of the image description, following the process outlined in section 4.1: first, the query image is segmented, a number of key ROIs are selected, and their features are computed. In this query type, the input is a description of the query image content, and the significance factors of ROIs and their features.
- *Query by Sketch:* In this case, the input is a sketch of relevant anatomical ROIs and the significance factors of these ROIs and their features. The user draws this sketch in the contour editor environment. To derive the image description, the geometric properties of all sketched ROIs are computed.

4.2.1 Matching Descriptions

The input to the matching algorithm consists of an image description I_Q, the significance factors w_r of ROIs, the significance factors w_f of ROI features, and D_d. Based on w_f, w_r, and D_d, the similarity of I_Q to each description in the database is computed and the images whose descriptions achieve the highest degree of similarity to the input image description are reported back to the user.

Let R_i, $i=1,\ldots,l$, be the set of ROIs that belong to the query image description I_Q and R_j', $j=1,\ldots,k$, the set of ROIs that belong to an image description I_{DB} stored in the database. If $k < l$, the matching algorithm adds $|l-k|$ dummy regions to the description with the fewer ROIs. First, the distance $D_R(R_i, R'_j)$ of every pair of regions R_i, R'_j in the two descriptions is computed:

$$D_R(R_i, R'_j) = \frac{1}{\sum_{f \in A} w_f} \sum_{f \in A} w_f D_f(R_i, R'_j)$$

where A is the set of features in the description, w_f is the user-defined significance factor for feature f, and D_f is a distance function which computes the normalized distance of regions R_i, and R'_j with respect to feature f. Normalization is achieved through division by $D_{max}(f)$, the maximum computed distance in f for all regions stored in the database. In the current implementation of *AttributeMatch*, $D_f(R_i, R'_j)$ is a normalized Euclidean distance metric. If the value of a feature is not available, $D_f(R_i, R'_j)$ is set equal to $D_{max}(f)$. This may occur, if feature f is a texture descriptor. Note that the denominator is a normalization factor that maps the distance of two regions to the interval [0,1].

Having computed the distance between each pair of regions (R_i, R'_j), the distance of the two descriptions I_Q, I_{DB} is computed as:

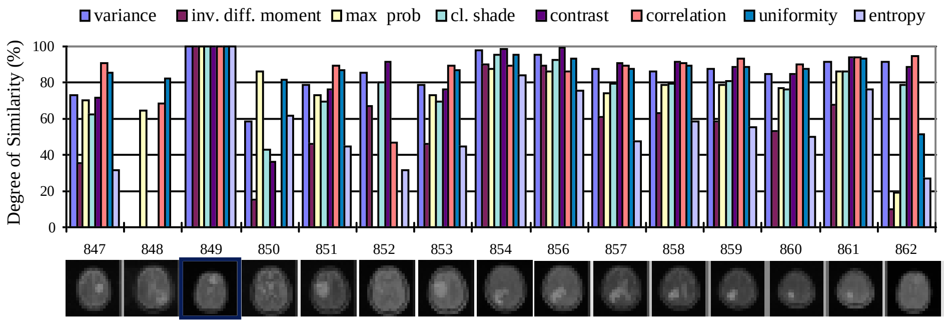
$$D_I(I_Q, I_{DB}) = \frac{1}{\sum_{r=1}^{l} w_r} \sum_{r=1}^{l} w_r D_R(R_r, R'_r)$$

where R_r and $R'_r = map(R_r)$, $r=1,\ldots,l$, are corresponding regions in the two descriptions and w_r is the significance factor of region R_r. The objective is to find the mapping function *map()* that minimizes $D_I(I_{DB}, I_Q)$. This step can be modeled as an *assignment* problem which is solved using a variant of the Hungarian Method [23] in $O(l^4)$ time, ie. polynomial time of order 4 with respect to l, for $l>k$.

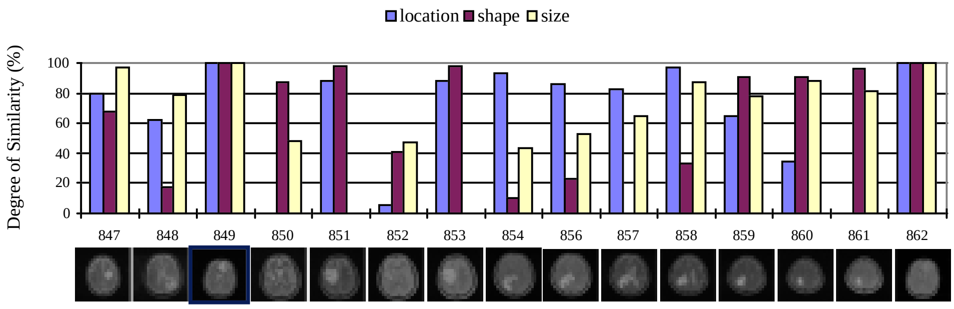
4.3 Performance

4.3.1 Adapting the Similarity Criterion

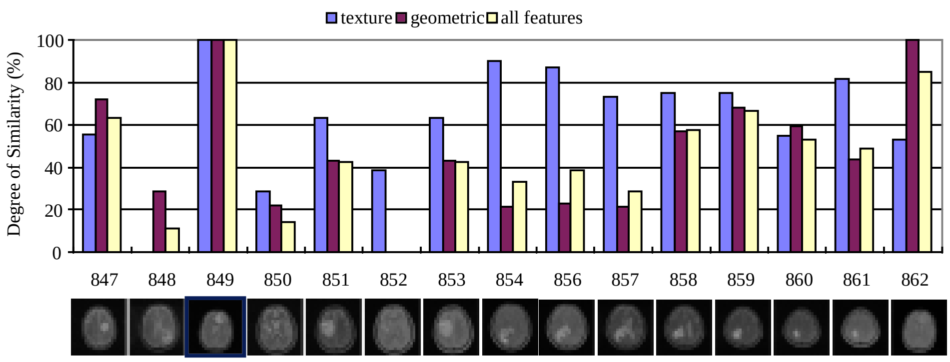
In the evaluation of *AttributeMatch*, experiments involving different similarity criteria were carried out, as follows: The descriptions of all images in an image class of MR Head images were generated using *AttributeMatch*. A query image was randomly selected from the class and its description was compared against all descriptions in the class. Since the query image description already exists in the database of descriptions, irrespective of the similarity criterion it always scored higher than any other image.



(a) Similarity based on Texture Descriptors



(b) Similarity based on Geometric Properties



(c) Similarity based on Different Types of Features

Figure 9: Query Response in *AttributeMatch:* A sample subset of images in a certain class has been selected and the degree of similarity between these images and the query image (849) has been computed based on different features.

Fig. 9 shows the similarity of a fixed image (no. 849) to a sample of the images in the class, using different similarity criteria. Note that changing the similarity criterion affects the degree of similarity between images in the class.

Fig. 9(a) shows the discrimination ability of different texture descriptors. Each bar shows the similarity to image no. 849 under a different texture descriptor. We observe that, although image 862 is structurally very similar to 849 (the query image), it has a much different texture according to most texture descriptors, except correlation. We also observe that image 849 is similar to images 854 and 856 according to most texture descriptors.

Fig. 9(b) shows image similarity to image no. 849, when only geometric properties are taken into account. We observe that image 862 is now identical to 849, and very similar to image no. 847.

Finally, Fig. 9(c) shows image similarity taking into account different types of properties: geometric, texture, and both combined. Images 862, 859, and 847 appear to be the most similar when all properties are taken into account.

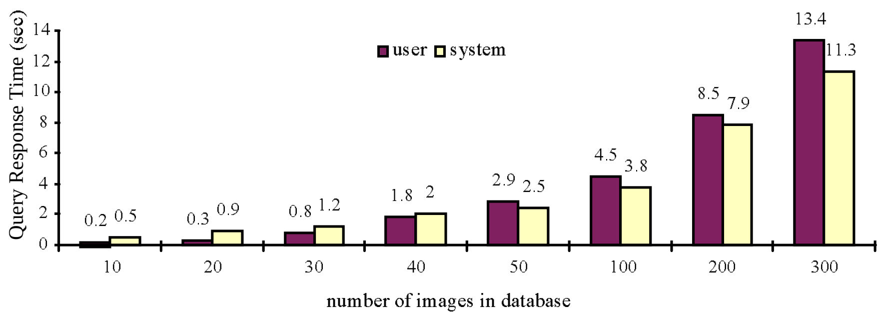


Figure 10: Query Response Time, as a function of search space, achieved by *AttributeMatch.*

4.3.2 *Efficiency of retrievals*

We have experimented with *AttributeMatch* queries in image classes of different sizes, and the average query response time appears in Fig. 10. The response time of various queries has been measured using the *UNIX* time command. Measured time has two components: operating system overhead (system time), and computation time (user time). Operating system overhead is fairly constant, while computation time increases with the number of images in the class. The actual response time observed by the user is the sum of system time and user time, plus the time overhead introduced by other processes in the system (time sharing).

A general similarity criterion forces the matching algorithm to compare the description of the query image with every image description in the database. We are currently experimenting with a heuristic search space reduction technique. The goal of this technique is to reduce the number of database descriptions considered in the search, without compromising retrieval accuracy.

5. Related Work

In conventional media databases, image retrieval is accomplished using associated textual information. Most commercial medical image databases allow diagnostic image retrieval based on individual fields in the medical record of a patient or information contained in the image record. Such conventional retrieval methods are no longer adequate due to the extensive use of multimedia in health care, the need to utilize as much of the information present in each medium as possible, and the requirement to fuse information from different sources. Therefore, the use of sophisticated image and video retrieval mechanisms in modern media databases is advocated [3,24-26].

A number of approaches to the content-based management of images have been proposed and implemented in research prototypes and commercial systems. Most systems support the combination of textual annotations and visual information for efficient browsing and navigation in image databases. In [25], the conceptual content of images is captured in annotations which are then indexed using the Semantic Indexing System [27] for the purpose of browsing and retrieval. The fusion of information from newspaper photographs and their captions is exploited in Piction [28], a system used to retrieve photographs based on their pictorial content. Tanimoto [29] introduced the iconic index, that is the use of picture icons as picture indices. Chang et.al. [24] propose protocols of use for goal oriented image prefetching. In I^2C, we deal mostly with visual content. Queries regarding patient data and visual content involve information stored by I^2C servers and Image Management and Communication Systems (IMACS). Such queries are decomposed and processed separately, thus leading to possibly more than one set of query results.

The problem of compressing images while retaining content information has been addressed in Photobook, a browsing and database search tool developed at the MIT Media Lab [30,31]. Photobook supports query by textual analogies as well as query by example. Images are represented in a compact way through semantics-preserving compression. The system measures image features such as brightness, edges, and texture, and selectively applies the Karhunen-Loeve or Wold transform to obtain a compact description of image features. When detailed relations between objects are important, the Karhunen-Loeve transform is used. Alternatively, the Wold transform [32] is used when describing textural properties such as orientation, randomness, and periodicity.

The QBIC (Query by Image Content) project at IBM, Almaden, also explores content-based retrieval methods [11]. It deals with the problem of content-based management for still images and video and it provides tools for the interactive and semi-automatic description of image content. QBIC offers the option of query by example, by sketch, by color, and by texture pattern. The representation of image content is based on attributes such as texture, shape, and color, as well as on user-defined attributes. QBIC addresses content-based similarity retrieval in video databases by regarding a video clip as a sequence of related image frames, from which one may select representative ones. Motion fields are then used to relate neighboring frames in the retrieval process. QBIC technology has been combined with traditional database search in a commercial product, the Ultimedia Manager [33].

In the Virage Engine based on the VIMSYS model [34], image properties computed over the entire image (global) or smaller regions of the image (local), are extracted by different computational processes, each having a different distance metric. Individual distance metrics are combined into a composite metric by the user, who adjusts a set of weighting factors and changes the current interpretation of similarity according to the task at hand. Image features considered include texture, color, and color composition. Virage technology has been integrated in the Illustra object-relational database [35].

The QBIC and Virage approaches are similar to I^2C. All three facilitate the addition of custom-made components. In I^2C, custom components are description types, segmentation algorithms, image processing algorithms, and properties. In QBIC, image content may be represented by user-defined attributes. To define a new primitive in the Virage Engine [36], extraction, distance, print, and marshaling functions need to supplied by the developer. In I^2C, the relation between content description and matching is loose, i.e. the same primitive may be used with a different distance function in different description types, while in Virage "description types" may be implemented as a *schema* of native primitives in the form of different applications.

CANDID, a system developed at the Los Alamos National Laboratory, supports query by image example [37]. A global signature is computed for every image in the database. The signature is derived from various image features such as texture, shape, color. A distance between probability density functions of feature vectors is used to compare image signatures.

Tagare et. all [38], have developed a query-by-pictorial-example system which can retrieve similar MRI images of the heart. For efficient retrieval of similar tomographic images, Voronoi diagrams are used to represent the spatial arrangement of anatomical parts. The modal matching approach has been proposed as a way to describe anatomical objects in terms of their generalized symmetries, as defined by their vibration or deformation modes [39,40].

Providing access to image databases through a WWW front end has received considerable attention [41]. Several research projects maintain on-line demonstrations of content-based retrieval through a standard WWW interface [42,43]. DOIA is a medical image database system that provides a WWW front-end. DOIA provides a unified global access structure to distributed dermatology images. For image retrieval in DOIA a predefined set of keys (common terminology) is used. It provides database search according to diagnosis, localization, and appearance of skin lesions [43].

Sclaroff [4] has proposed the development of a WWW image search engine that crawls the web collecting information about the images it finds, computes the appropriate image decompositions and indices, and stores the extracted information. Since such a system should employ automatic image content representation, an arsenal of image decompositions and discriminants is precomputed. At search time, the users may select a weighted subset of these decompositions to be used for computing image similarity measurements. Our work in I^2*Cnet* has the same flavor, in that it addresses a new service to the regional health care network through a WWW interface, providing transparent access to multiple autonomous, geographically distributed repositories of medical images.

6. Conclusions

In this paper, we presented the architecture of I^2*Cnet*, a network of content-based similarity search engines, which has been designed to provide the medical domain with an added-value information service using a World Wide Web interface.

I^2*Cnet* extends the functionality of I^2C, an existing standalone system for the management of images based on their content, by providing a richer set of query types and uniform access to other engines, which exist or will someday exist on the Internet. The I^2C service broker activates software agents to actively maintain profile information in I^2*Cnet*, in order to provide continuing support for network-transparent queries. Cooperation among I^2C servers and description types will be transparent to the user, unless it is otherwise requested. As a case study, *AttributeMatch*, a description type currently supported by I^2C, has been presented with performance results from its application to a class of MR head images.

The implementation of I^2*Cnet* is now in progress and we expect that a new multimedia service will soon be available to the medical community through the Web. Preliminary results obtained with the stand alone I^2C system indicate that query response time scales well with image database size and that adequate retrieval accuracy can be achieved using clinically relevant similarity criteria.

Acknowledgments

The I^2C work was carried out in collaboration with KNOSSOS Technologies S.A. and was funded in part under project EurIPACS of the CEC AIM Programme.

Many people have contributed to this work. Costas Georgiadis implemented part of the *AttributeMatch* description type and Stavros Kostomanolakis developed part of the I^2C user interface and the database engine. Evangelos Markatos contributed to this paper with many useful suggestions. The design of the I^2*Cnet* architecture benefited from many fruitful discussions with all other members of the Medical Information Systems Laboratory at ICS-FORTH. Their contributions are greatly appreciated.

References

1. Grosky, W.I. Multimedia information systems. IEEE Multimedia. 1(1):12-24; 1994.
2. Furht, B. Multimedia systems: an overview. IEEE Multimedia. 1(1):47-59; 1994.
3. Jain, R.; Pentland, A.P.; Petkovic, D. Workshop Report: NSF-ARPA Workshop on Visual Information Management Systems. Tech. Report, Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407; 1995.
4. Sclaroff, S. World Wide Web image search engines. Tech. Report TR-95-16, Image and Video Computing Group, Comp. Sci., Boston University, Boston, MA 02215; 1995.
5. Orphanoudakis, S.O.; Chronaki, C.E.; Kostomanolakis, S. I^2C: A system for the indexing, storage, and retrieval of medical images by content. J. Med. Inform. 19(2):109-122; 1994.
6. Carey, M.; DeWitt, D.; Frank, D.; Graefe, G.; Richardson, J.; Shekita, E.; Muralikrishna, M. The architecture of the EXODUS extensible DBMS. In: Dittrich, K.R.; Dayal, U.; Buchman, A.P., eds. On Object-Oriented Database Systems. Springer-Verlag; 1991:231-257.

7. Carey, M.; DeWitt, D.; Naughton, J. The OO7 benchmark. In: Proc. of the 1993 ACM-SIGMOD Conf. Washington, D.C; 1993.
8. Medical Information Systems Laboratory. TelePACS: A system for the acquisition, processing and communication of medical images. Int. Report, ICS-FORTH; 1995.
9. Carey, M.; DeWitt, D.; Franklin, M.; Hall, N.; McAuliffe, M.; Naughton, J.; Schuh, D.; Solomon, M.; Tan, C.K.; Tsatalos, O.; White, S.; Zwilling, M. Shoring up persistent applications. In: Proc. of the ACM-SIGMOD Conf. on the Management of Data. Minneapolis MN; May 1994.
10. Orphanoudakis, S.; Tsiknakis, M.; Chronaki, C.; Kostomanolakis, S.; Zikos, M.; Tsamardinos, Y. Development of an integrated image management and communication system on Crete. In: CAR'95. Berlin, June 21-24; Springer 1995:481-487.
11. Flickner, M.; Sawhney, H.; Niblack, W.; Ashley, J.; Huang, Q.; Dom, B.; Gorkani, M.; Hafner, J.; Lee, D.; Petkovic, D.; Steele, D.; Yanker, P. Query by Image and Video Content: The QBIC system. IEEE Computer. 28(9):23-32; 1995.
12. Etzioni. O.; Weld, D. Intelligent agents on the internet: fact, fiction, and forecast. IEEE Expert. 10(4):44-49; 1995.
13. Etzioni, O.; Weld, D. A Softbot-based interface to the internet. Comm. of the ACM. 37(7):72-76; 1994.
14. Hedberg, S.R. Intelligent agents: the first harvest of Softbots looks promising. IEEE Expert. 10(4):6-9; 1995.
15. Bloomer, J. Power programming with RPC. O' Reilly & Ass.; 1992.
16. Vinoski, S. Distributed object computing with CORBA. C++ Report Magazine, Jul./Aug. 1993.
17. Pfitzmann, A.; Pfitzmann, B. Technical aspects of data protection in health care informatics. In: Advances in Medical Informatics. IOS Press; 1992:368-386.
18. Bleumer, G. Security of decentralized health information systems. Int. J. of Biomed. Comp. 139-145; Feb. 1994.
19. Chaum, D. Security without identification: transaction systems to make big brother obsolete. Comm. of the ACM. 28(10):1030 -1044; 1985.
20. Orphanoudakis, S.; Tsiknakis, M.; Chronaki, C.; Kostomanolakis, S. The regional health telematic network of Crete. In: Proc. Health Telematics 95. Ischia Italy; Jul. 1995.
21. Proc. of The 8th Scandinavian Conf. on Image Analysis, VOL I; 1993.
22. Mandelbrot, B. The fractal geometry of nature. Freeman, San Francisco, Ca; 1982.
23. Papadimitriou, C.; Steiglitz, K. Combinatorial optimization: algorithms and complexity. Prentice Hall; 1982.
24. Chang, S.; Hou, T.Y.; Hsu, A. Smart image design for large image databases. J. Vis. Lang. and Comp. 323-342; 1992.
25. Constantopoulos, P.; Doerr, M. An approach to indexing annotated images. In: Third Int. Conf. on Hypermedia and Interactivity in Museums - ICHIM '95. San Diego, CA. Oct. 9-13; 1995.
26. Gudivada, V.; Raghavan, V. Content-based image retrieval systems. IEEE Computer. 28(9):18-22; 1995.
27. Constantopoulos, P.; Doerr, M. The Semantic Indexing System. Internal Report, Institute of Computer Science, Foundation for Research & Technology Hellas; 1995.
28. Srihari, R.K. Automatic indexing and content-based retrieval of captioned images. IEEE Computer. 28(9):49-56; 1995.

29. Tanimoto, S. An iconic/symbolic data structuring scheme. In: Pattern Recognition and Artificial Intelligence. New York, Academic; 1976:452-471.
30. Pentland, A.; Picard, R.W.; Sclaroff, S. Photobook: Tools for content-based manipulation of image databases. In: Proc. of Storage and Retrieval for Image and Video Databases II; Vol 2, 185, SPIE. Bellingham, Wash.; 1994:34-47.
31. Pentland, A.; Picard, R.; Davenport, G.; Haase, K. Video and image semantics: advanced tools for telecommunications. IEEE Multimedia. 73-75; 1995.
32. Liu, F.; Picard, R.W. Periodicity, directionality, and randomness: Wold transform features for image modeling and retrieval. Tech. Report 320, M.I.T. Media Lab. Perceptual Computing Section, 20 Ames St., Cambridge, Ma 02139; 1995.
33. Treat, H.; Ort, E.; Ho, J.; Vo, M.; Jang, J.; Hall, L.; Tung, F.; Petkovic, D. Searching images using Ultimedia manager. Res. report, IBM Santa Teresa Lab; 1993.
34. Gupta, A.; Weymouth, T.; Jain, R. Semantic queries with pictures: The VIMSYS model. In: Proc. of the 17th Int. Conf. on VLDB. Barcelona, Spain; 1991:69-79.
35. Stonebraker, M. Object-relational DBMS - the next wave. White paper, Illustra Information Technologies, Inc, Illustra Information Technologies, Inc., 1111 Broadway, Suite 2000, Oakland, CA 94607; 1995.
36. Gupta, G. Visual information retrieval technology - a Virage perspective. White paper, Virage Inc.; 1995.
37. Kelly, P.M.; Cannon, T.M. Experience with CANDID: Comparison algorithm for navigating digital image databases. In: SPIE Vol. 2368 Proc. of the 23rd AIPR Workshop on Image and Information Systems: Applications and Opportunities. Wash. DC, Oct. 12-14; 1994:64-75.
38. Tagare, H.D.; Vos, F.M.; Jaffe, C.C.; Duncan, J.S. Arrangement: a spatial relation between parts for evaluating similarity of tomographic sections. PAMI. 17(9):880-893; 1995.
39. Sclaroff, S.; Pentland, A.P. On modal modeling for medical images: Underconstrained shape description and data compression. In: IEEE Workshop on Biomedical Image Analysis. Seattle; Jun. 1994:70-79.
40. Sclaroff, S.; Pentland, A. Modal matching for correspondence and recognition. PAMI. 17(6); 1995.
41. Bittorf, A.; Krejci-Papa, N.C; Diepgen, T.L. Web image databases: Formats, sources, organization. In: Proc. 4th Int. World Wide Web Conf. Tutorial Notes; O'Reilly Ass.; 1995:141-161.
42. Virage Inc. Demonstration of visual information management software. http://www.virage.com/vir-res/demos; Virage Inc.; 1995.
43. Bittorf, A.; KrejciPapa, N.C.; Diepgen, T.L. Development of a dermatological image atlas with worldwide access for the continuing education of physicians. J. Telemed. & Telecare. 1:45-53; 1995.