CONTENTS

1

CONTENT-BASED IMAGE RETRIEVAL USING METADATA AND RELAXATION TECHNIQUES

W. W. Chu, C.-C. Hsu,
I. T. Ieong*, and R. K. Taira+

Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90024

** NCR Corporation*
100 N. Sepulveda Blvd.,
El Segundo, CA 90245

+ Department of Radiological Sciences
University of California, Los Angeles
Los Angeles, CA 90024

A chapter in *Managing Multimedia Data: Using Metadata to Integrate and Apply Digital Data,*
edited by Wolfgang Klas and Amit Sheth

ABSTRACT

In this chapter, a metadata approach is proposed for content-based image retrieval. A three-layered data model is introduced which consists of a Representation Layer (RL), a Semantic Layer (SL), and a Knowledge Layer (KL). Raw images and segmented objects (represented in contours) are represented in the RL. Hierarchical, spatial, temporal, and evolutionary semantics of the image content are captured in the SL. This layer mimics the user's conceptual view of the image content. *Shape model(s)* and *decomposition techniques* are introduced for extracting features for differently shaped image objects. Knowledge of image feature characteristics is presented in the KL. An instance-based conceptual clustering technique is used to cluster conceptually similar image objects and features. A hierarchical knowledge representation, called *type abstraction hierarchy,* is used to represent the clustering of similar images based on a common set of features. A knowledge-based spatial evolutionary query language, KEQL, is developed to provide spatial and evolutionary constructs to query the image content with conceptual terms and approximate predicates. The knowledge base is used to relax query conditions when no answer or no satisfactory answer is returned.

Further, the knowledge base is used to interpret the high level conceptual terms and approximate predicates.

Using an object-oriented database, the proposed data model and query processing methodology have been implemented for a medical image database system at UCLA to demonstrate the validity of the metadata approach for retrieving medical images by features and content.

1 INTRODUCTION

Advances in medical imaging systems have revolutionized the acquisition of human images, providing views of cross sections and physiological states using a variety of modalities, including X-rays, computed tomography (CT), magnetic resonance (MR), etc. Currently, a medical picture archival and communication system (PACS) infrastructure is operational at UCLA to provide efficient archival, retrieval, communication, and display of the large repository of digital medical images [20, 39]. The digital image data is acquired automatically by PACS which contains two optical disk library units, each with a capacity of one terabyte. At present, image retrieval in these systems is based on file names or artificial keys such as *patient hospital identification numbers*. There is a need to retrieve the images by content. For example, in therapy treatment planning, the therapist is often interested in retrieving historical cases which demonstrate diagnostic image features (object shape, size, location, extent, clustering patterns, etc.) similar to a current study of interest.

The knowledge of the spatial content of a medical image is important in surgical or radiation therapy of brain tumors since the location of a tumor has profound implications for a therapeutic decision. The evolutionary behavior of a tumor is important when studying the treatment-response effects to various therapies. Also, the temporal characteristics of image features help describe the behavior of disease processes. For example, what is the normal growth rate of tumors for a given classification of patients? How does this growth rate change with various regiments of drug/radiation therapy?

Several approaches have been proposed to retrieve images in pictorial databases. Features of image objects are often used to index the images. In such systems, retrieval is based on the indexed features [17]. The information retrieval approach [34, 35] transforms the image and the query into signatures. The query is answered by matching the signatures. In addition, there are approaches which store boundaries of image objects using minimum bounding box representations [36, 18]. Since a bounding box is merely an approximation of the boundary of the original image object, representing the image object with such structures may not allow the user to retrieve all the answers that satisfy the query specification. Image objects can also be represented in point sets [31]. Spatial relationships among the image objects are dynamically determined through pixel-level manipulation during query processing. The orthogonal relationships among objects can also be encoded into a 2D string [5]. As a result, user queries concerning the orthogonal relations of the objects can be answered.

In addition, there are attempts to answer approximate queries concerning spatial objects. QBIC [29] uses object features such as *area* and *circularity* to retrieve similarly shaped objects. VIMS [1] retrieves similar images by relaxing feature values of the image based on the standard deviation of the features.

Existing approaches to image retrieval often focus on a specific application domain and have limited query capabilities[17, 5, 34, 29]. Layered model has been proposed to represent the *raw features* and *qualitative features* of video images [?]. Biological objects *evolve*, *fuse*, *split*, and change their spatial orientation over time. Few approaches support temporal spatial query answering [31] or temporal evolutionary query answering [11], and none support spatial evolutionary query answering. This is due to the lack of appropriate metadata that captures the spatial and evolutionary nature of image content for supporting image retrieval.

To remedy such shortcomings, in Section 2 we propose the use of metadata to capture image content and knowledge for supporting spatial evolutionary query answering on image content. The metadata includes:

1. *Raw-image-level metadata* for providing the characteristics of the raw images and pixel-level description of the image content (e.g., contours of interested image objects)

2. *Spatial, shape, temporal and evolutionary metadata* for modeling the spatial, shape, temporal, and evolutionary characteristics of the objects in the images

3. *Knowledge-level metadata* for representing the knowledge of spatial, shape, temporal and evolutionary characteristics of the image content

Section 3 describes our feature extraction of images for querying. Section 4 presents the modeling of medical image content with semantic-level metadata.

A knowledge-based spatial evolutionary query language (KEQL) for expressing user queries is introduced in Section 5, and knowledge-level metadata for intelligent interpretation of image features and content is discussed in Section 6. The knowledge-based query processing is provided in Section 7, and finally the system implementation is provided in Section 8.

2 A MULTI-LAYERED METADATA MODEL FOR REPRESENTING IMAGE CONTENT

A three layered model is used to organize the image metadata which consists of the Representation Layer (RL), the Semantic Layer (SL), and the Knowledge Layer (KL) (see Figure 1).

Raw images and contours of the objects in the images are stored in the RL. Related images are organized and stored in various stacks for efficient access [6]. The RL includes two types of metadata: 1) characteristics of the acquired raw images and 2) contours of query-able objects in the images. The characteristics of raw images (e.g., pixel width, dimensionality of images, the height and angle that a brain MR image is taken, number of slices in an MR image set, the spacing between two MR image slices, the machine and technician acquiring the images, etc.) are contained in the header of the (medical) image.

The SL stores metadata for object features and spatial relationships among objects. Techniques are developed to describe the shape of objects and their spatial relationships (e.g., *Joined, Disjoined*, etc.) and evolutionary relation-

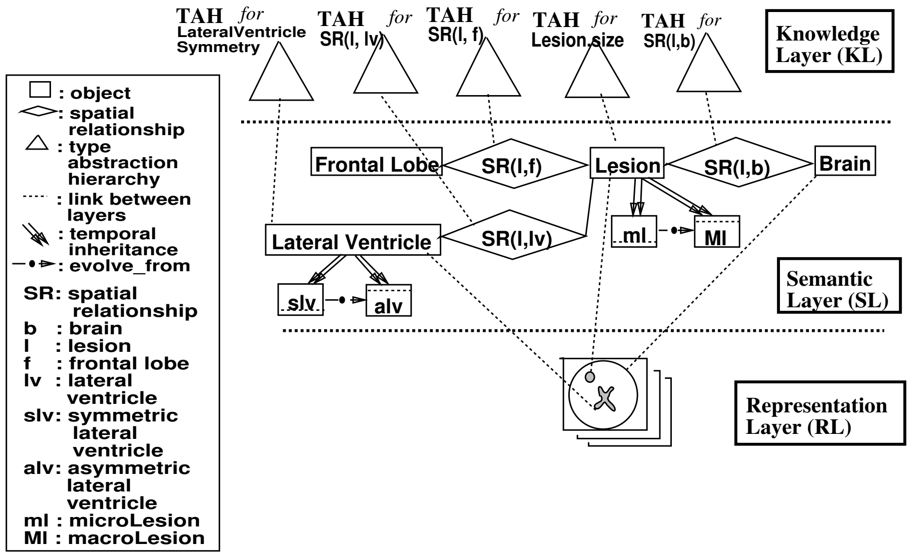


Figure 1 Modeling of metadata for image features and content

ships (e.g., *Evolve, Split, Fuse*). An object-oriented approach is used to model the features extracted from the contours of interested objects. The model allows us to express spatial, temporal, and evolutionary objects. The metadata about these query-able objects and features are domain-dependent and application dependent.

The knowledge layer (KL) stores the knowledge-level metadata for supporting approximate and conceptual query answering on image content. An instance-based conceptual clustering technique, MDISC [9], is used to cluster image objects based on user-specified conditions (e.g., shape descriptors and spatial relationships). The clustering of image objects is represented in a hierarchical knowledge structure, called Type Abstraction Hierarchy (TAH) [9], for supporting query answering involving similar-to predicates, spatial semantic operators (e.g., *Nearby* and *Far_Away*), and conceptual terms (e.g., *Large* and *Small*). These knowledge structures provides domain-dependent ontologies for approximate and conceptual terms used in querying image content. These metadata also help restrict the vocabularies in querying spatial, temporal, and evolutionary object features.

3 FEATURE EXTRACTION

3.1 Object Segmentation

Traditional image segmentation approaches are broadly categorized as based on intensity discrimination [3] or edge detection [2].

The intensity approach assumes that a fundamental relationship exists between the pixel intensity and the physical substrate. Theoretically, imaging instruments are measuring specific physical properties such as tissue relaxation time, electron density, or atomic number. However, numerous studies have shown that factors such as field inhomogeneity, instrumentation noise, and partial volume effects all contribute to pixel intensity fluctuation and create uncertainty in discrimination [3]. Thus, simple global thresholding methods usually fail due to inherent image noise [33].

Current edge-detection-based work shows some degrees of success [26], but in general requires user interaction. Most edge-detection systems cannot discriminate between true object edges, noise edges, and edges originating from, for example, a second overlapping structure. A fundamental understanding of the nature of the basic signals and hence, edge characteristics, is somewhat lacking.

Some success has been gained by the use of multi-scale edge analysis [25]. The overall strategy is that noise-free edges can be obtained from a coarse scale, while a true edge location is obtained from a fine scale. However, analyzing the multi-scale behavior of edges is generally done by using heuristic rules. Visual interpretation plays an important part in correlating edges in different scales as well as in making decisions on determining a true or false edge. A lack of rigorous means for performing multi-scale edge analysis as well as parsing has seriously limited the effectiveness of these kinds of methods for automated segmentation. Most of the successful semi-automatic image segmentation methods used in image management systems fall into this category, such as active contouring [13] and liquid transforms [29] where the coarse contour or seed point is manually provided.

Most recent work in segmentation of anatomic objects involves model-guided knowledge-based methods. The important features of the model-guided, knowledge-based image segmentation approaches include [24]:

1. Physical models of each target image object (e.g., brain, lateral ventricle, cranium) are maintained within an attached knowledge-base which can be maintain:

    - Compositional models: These models identify what substructures are part of a larger superstructure. These compositional models allow the segmentation system to first localize larger superstructures in the image and leverage on important landmarks of these superstructures to localize more specific target substructures. (e.g., localizing the brain cranium constrains the spatial searching of lateral ventricle).

- Grey-level models: These models provide statistical information regarding the expected gray-level distribution of various tissue components (e.g., brain gray matter, brain white matter, cerebral spinal fluid, bone, background, common artifacts) within a medical image superstructure (e.g., brain and hand).
- Shape models: A shape contour analyzer decomposes complex object contours into more fundamental lines and curve segments. It is used by the system to verify that contoured objects conform to the general expected shape of the target object.

2. Object-oriented organization allows the system to attach specific processing methods, relationships, rules and data for each target object (e.g., brain objects, tumor objects, and hand objects).
3. Object models are used to identify important landmarks on object contours [16].
4. Models are used to verify the expected shape of extracted contours and localize any errors signifying gross difference between object shape and extracted contour shape.

3.2 Extracting Shape Features Via Decomposition

Shape is an important property for retrieving images. Global shape measurements such as area and perimeter only provide a coarse description of shape. In QBIC[29], global shape descriptors such as *circularity* are used with area and perimeter value to describe the shape of image objects. Global shape descriptors such as *curveness*, *rectangularity*, *compactness*, *direction*, *elongatedness*, and *eccentricity* [38] provide certain approximate shape measures for contours of relatively simple shapes (*i.e.*, shape without irregular intrusions and/or protrusions). However, for contours of complex shapes (i.e., shapes with irregular intrusions and/or protrusions), these descriptors do not provide sufficient expressive power for describing their specific shape content [23]. Therefore, we propose a *landmark-based decomposition technique* to decompose complex-shaped contours into shape descriptive sub-structures based on the content-related landmark(s) in the images.

Landmarks are characteristic points in an image which are easily recognizable on all images and can be used to decompose *every* instance of the complex object into descriptive sub-structures (e.g., the tips and webs of a hand are the landmarks for decomposing a hand contour for detailed shape description of a hand). Landmark points in a contoured object can be obtained, for example, by matching model segments with corresponding data segments as described in [16] and [24].

With the landmark identified, the complex-shaped contours are decomposed into shape-descriptive sub-structures. The shape content of the complex-shaped object can be represented by the simpler shape and spatial relationships of the decomposed sub-structures. Attributes of the sub-structures can be common spatial characteristics (such as area, length, and width) or shape descriptors of the decomposed sub-structures. Capturing these attributes provides a more precise description on the shape of the image objects.

The lateral ventricle is used as an example object to illustrate landmark-based decomposition in studying the shape symmetry of the lateral ventricle. In medicine, the symmetry of the lateral ventricle is observed using the two tip notches of the brain contour as the landmarks. The brain contour is modeled using the *shape contour model* to automatically identify the two tips. Then the lateral ventricle can be decomposed into four substructures as shown in Figure 2 to studying the symmetry of lateral ventricle. We can express the symmetry aspect of the lateral ventricle via the relative shape among these decomposed contours. In Figure 2, the height and width of the four components of a lateral ventricle are used to construct a *shape feature* to describe the left to right symmetry of the lateral ventricle as (upperLRWidthRatio (w_{ul}/w_{ur}), upperLRHeightRatio (h_{ul}/h_{ur}), lowerLRWidthRatio (w_{ll}/w_{lr}), lowerLRHeightRatio (h_{ll}/h_{lr})). The *left to right symmetry* of the lateral ventricle is an important shape feature for detecting various brain diseases [15].

3.3 Extracting Spatial Relationship Features

The primary concerns for a quantative description of spatial relationships include:

- Containment relationships which describe the relative position and locations of contact between a pair of objects. The detailed containment characteristics are further quantatively captured by features such as ratio of overlap, ratio of area, the shortest distance between their contours, etc.
- Directional relationships describe the directional characteristics between objects. These directional features can be quantatively captured by fea-

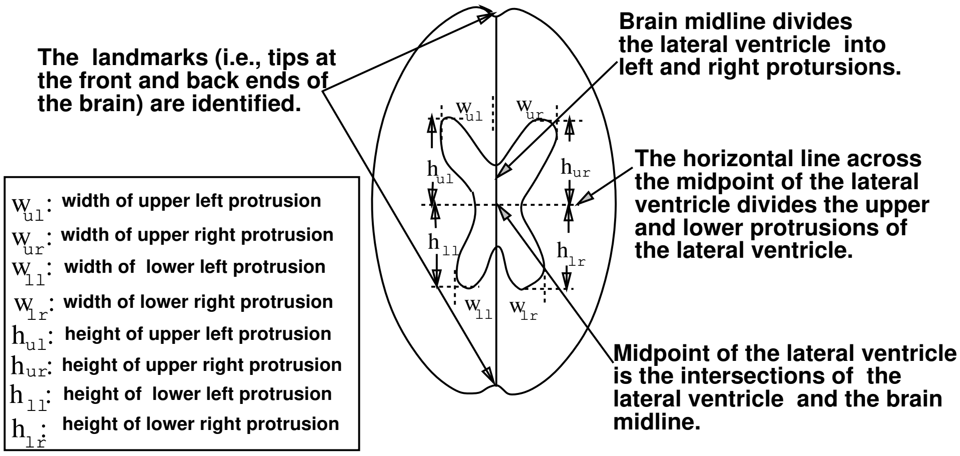


Figure 2 For more precise shape description, the shape model decomposes a lateral ventricle into four natural sub-structures: upper left substructure, upper right substructure, lower left substructure, and lower right substructure.

tures such as relative distance of centroid on *x axis*, *y axis*, and *z axis*, respectively.

Based on anatomical knowledge and physician diagnosis, the spatial relationships of an object with respect to its surrounding objects can be derived; we shall call them the *primary spatial relationships* of the objects. We use primary spatial relationships to represent the semantics of the images in the data model. In addition, the relationships of abnormal objects discovered in diagnosis with respect to their adjacent objects are also kept in the system. Due to the relatively small number of abnormal objects found, we only need to maintain a small fraction of spatial relations in the system.

The number of spatial relations for n objects in an image is on the order of $O(n^2)$. Storing all the spatial relations not only increases the storage but also lengthens the time needed to search for the objects. To limit the size of storage of spatial features and relations, only those most frequently inquired about are stored in the system. There is a storage/response time tradeoff as to what spatial features and relations need to be stored.

4 MODELING OF OBJECTS IN MEDICAL IMAGES

The metadata in the semantic layer (SL), which models the temporal, spatial, and evolutionary aspects of the image objects, is detailed in the following sections.

4.1 Representing Image Objects

Objects and their contours in the images are identified by segmentation software. Features and content representing object characteristics are extracted and represented in the data model. Similar image objects are classified in the same class. For example, the object class defined for the micro-lesion in our Gemstone implementation is:

```
Object subclass 'Microlesion'
    instVarNames: #['diameter',
        'volume', 'eventTime', 'contourPointer', 'imagePointer']
    constraints: #[ #diameter, Number],
        #[ #volume, Number],
        #[ #eventTime, DateTime],
```

```
        #[ #contourPointer, Contour],
        #[ #imagePointer, Image].
```

`Microlesion` is a subclass of `Object`. The `instVarNames` defines the attributes: `diameter, volume, eventTime, contourPointer`, and `imagePointer` for object `Micro-lesion`. The `constraints` defines the type of each attribute. For example, `volume` and `diameter` are of type `Number`.

`MicroLesionSet` is an object class in which each instance is a `Microlesion`, as shown in the following:

```
Set subclass 'MicroLesionSet'
    constraints: MicroLesion.
```

If an image in the database contains a micro-lesion, then there is an object instance in `MicroLesionSet` corresponding to that micro-lesion which is represented by its features. An object class is created for each type of identifiable object in all the images. Thus, such data structures provide a framework and guidelines to extract features from all images. The object instance maintains links that point to its contours and images. The collection of all feature values constructs a *feature database*. These pre-computed feature values in the feature database enable the conceptual feature clustering and also speed up the image retrieval by feature and content.

Our data model extends the object-oriented model and introduces new constructs to describe the spatial and evolutionary behavior of objects.

object feature	**conceptual terms**
lesion.size	*small, medium, large*
lesion.roundness	*circular, non_circular*
lesion.surface	*smooth, coarse*
lateral_ventricle.left_to_right_symmetry	*upper_protrusion_pressed_to_the_right, upper_protrusion_pressed_to_the_left*, ...
...	...

Table 1 A *shape feature description table* for the brain

4.2 Modeling Shape

The proposed landmark-based decomposition technique provides an effective quantitative shape description when the image objects have limited numbers of shape components. The decomposed objects provides sufficient image content to retrieve specifically or similarly shaped image objects. Further, with image classification algorithms, such as MDISC [9], conceptual terms can be defined to describe a shape feature. The *shape feature description table* (Table 1) lists the conceptual terms for describing the shape features. Thus, users can use conceptual terms for describing a specific shape in a query: for example, *"retrieving images with lateral ventricles whose upper protrusion are pressed to the right"* (see *Query 3* in Section 5 and the MTAH defining *upper protrusion pressed to the right* in Figure 3).

4.3 Modeling Spatial Relationships

Modeling spatial relationships merely by simple semantic constructs such as *separated* and *connected* is insufficient to express real-life spatial relationships (as illustrated in Figure 4). Additional parameters are needed to describe

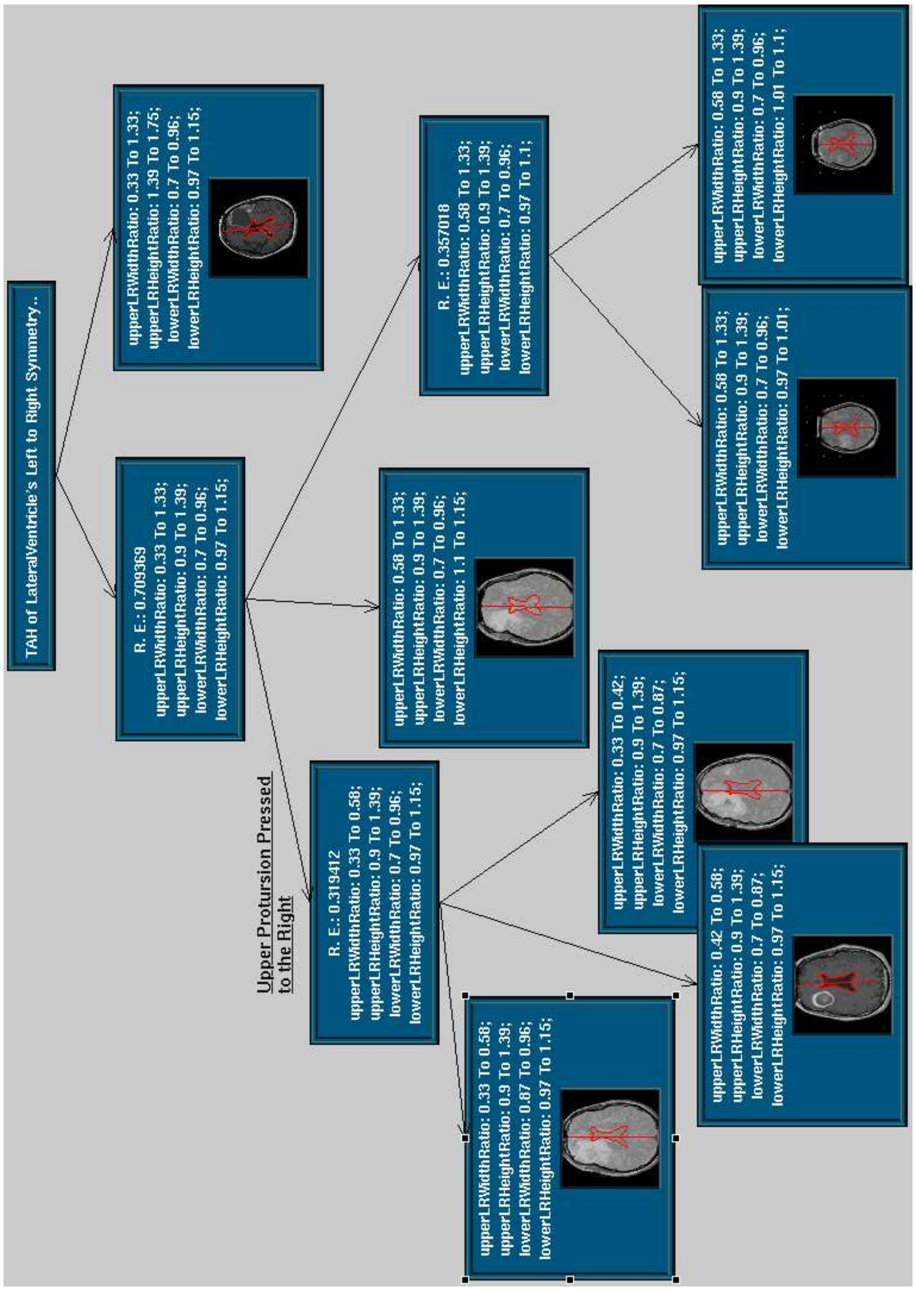


Figure 3 Multi-attribute type abstraction hierarchy (generated by MDISC based on the decomposed four protrusions) representing the left to right symmetry of the lateral ventricles

spatial relationship	representative features	defined semantic terms
SR(l,b)	$(x_c, y_c, r_a), (a_i)$	Slightly_Occupied, Extremely_Occupied
SR(l,f)	(x_c, y_c, r_j)	Slightly_Touching, Intimately_Touching
SR(l,lv)	$(\theta_c, d_c, x_c, y_c)$	Nearby, Far_Away
...	...	...

Table 2 A *spatial relationship description table* for the brain lesion (l: lesion; b: brain; f: frontal lobe; lv: lateral ventricle).

the spatial relationships more precisely. A set of required spatial relationship features should be specified by domain experts. Parameters are provided to describe the topological relationships between two objects (as shown in Figure 5). More important parameters for distinguishing the sub-types under a category are placed first in the list, and parameters used in the higher branches may also be used in their descendant branches. In Figure 5, `Bordering` means that only the surfaces of the two objects are joined (i.e., $r_c > 0, r_j = 0$); `Invading` implies their areas are joined (i.e., one of the object is deformed by the other, $0 < r_j < 100\%$), and `Circumjacent` implies a full containment (*i.e.*, one object fully contains the other, $r_j = 100\%$). The *required constructs/operators* are necessary for every spatial relationship. A *spatial relationship description table* (as shown in Table 2) lists the available parameters and semantic terms for describing the spatial relationships between two objects. The values of these spatial relationships are stored in the database.

This spatial modeling approach captures the containment and directional aspects of binary spatial relationships. Containment describes the relative position and possible *joining* relationship among objects while direction aspect

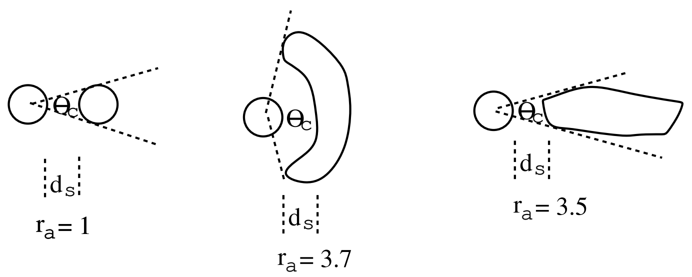


Figure 4 An example showing that merely using semantic operators (e.g., *SEPARATED*) and/or single measurement (e.g., the *shortest distance* (d_s)) is insufficient to capture the spatial relationship of two objects. We need additional features such as *angle of coverage* (θ_c) and *ratio of area* (r_a) to classify the illustrated spatial relationship.

describes the directional relationships from one contour to the other. The constructs for the spatial relationships are as follows:

1. No containment

   - `Disjoined`: This spatial construct is used to describe two objects that are separated from one another. (For example, the micro-lesion is disjoined from the lateral ventricle as shown in Figure 10). The relationships under this construct are further quantified by the spatial relationship features such as d_s, d_c, r_a, and θ_c.

     The values of these spatial relationship features can be conceptually classified via clustering algorithms such as MDISC [9] to derive the meaning of the corresponding semantic terms under the `DISJOINED` category, such as `NEARBY` and `FAR_AWAY`. The reason for using an instance-based clustering algorithm to define these corresponding semantic terms is that the semantics of these detailed spatial relation-

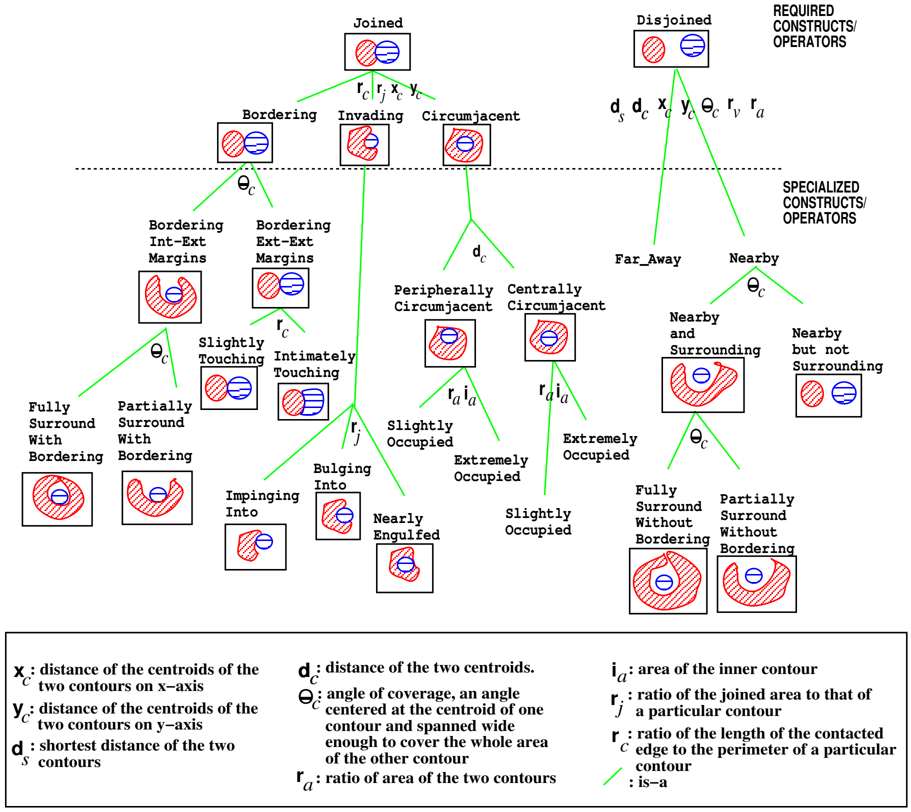


Figure 5 An ontology of spatial relationships (represented in a generalization hierarchy) for different topological categories between two objects (with the representative icons shown) for oncological medical images. The parameters under a branch classify the sub-types under that category.

ship terms are context-dependent (e.g., the meaning of `NEARBY` in astrology is different from its meaning in medicine).

- `Bordering`: Two objects contact one another at one or more locations. Bordering is a significant event in the medical domain. The relationships under this construct are further quantified by spatial relationship features such as r_c and θ_c.

2. Containment

   - `Circumjacent`: An object contains another object (for example, a micro-lesion develops inside the brain, as shown in Figure 10). The relationships under this construct are further quantified by the spatial relationship features such as d_c, r_a, and i_a.

3. Partial Containment

   - `Invading`: This spatial construct is used to describe one object that invades the other which causes an object to deform. The relationships under this construct can be further quantified by the spatial relationship features such as r_j and d_c.

The directional information of spatial relationships is captured by spatial relationship features such as the relative distance of the centroids of two objects on the x-coordinate (denoted as x_c), on the y-coordinate (denoted as y_c), and on the z-coordinate (denoted as z_c).

To model a spatial relationship, domain experts need to identify the containment categories of the modeled spatial relationship. Spatial relationship features are then identified to describe the spatial relationship according to the topological categories of the two objects.

For example, in an image with a lesion and lateral ventricle, the spatial relationship instance between the lesion and lateral ventricle is classified as an instance of the class SR(lesion,lateral_ventricle). The containment relationship covers the `DISJOINED` and `BORDERING` categories and requires θ_c, d_c, x_c, and

y_c as the features to describe this spatial relationship. Figure 6 is an image classification hierarchy for the images in the database. The hierarchy is generated by MDISC based on the collected values of spatial relationship features of SR(lesion,lateral_ventricle) instances in the database.

4.4 Temporal Relationship Object Constructs

Let us now consider the modeling of inter-object temporal relations. The constructs that represent the temporal relationships between an object and its supertype or its aggregated type are shown in Figure 7. These temporal relationships define how the characteristics of the supertypes are temporally inherited. **Temporal inheritance** deals with the way time-dependent characteristics of a supertype are inherited by its subtypes. The general rule is that an object may only inherit characteristics from other objects which exist in its own space-time domain [11].

To illustrate the use of the temporal relation object constructs, we apply them to the model describing brain lesion growth. For example, the life of a lesion includes its development from a micro-lesion to a macro-lesion. The temporal inheritance between these object types is expressed in Figure 8 (where the black solid line represents a temporal inheritance relationship).

4.5 Evolutionary Object Constructs

Existing approaches for modeling object evolution involves maintaining object versions [22]. We enhance these basic models to describe the development of

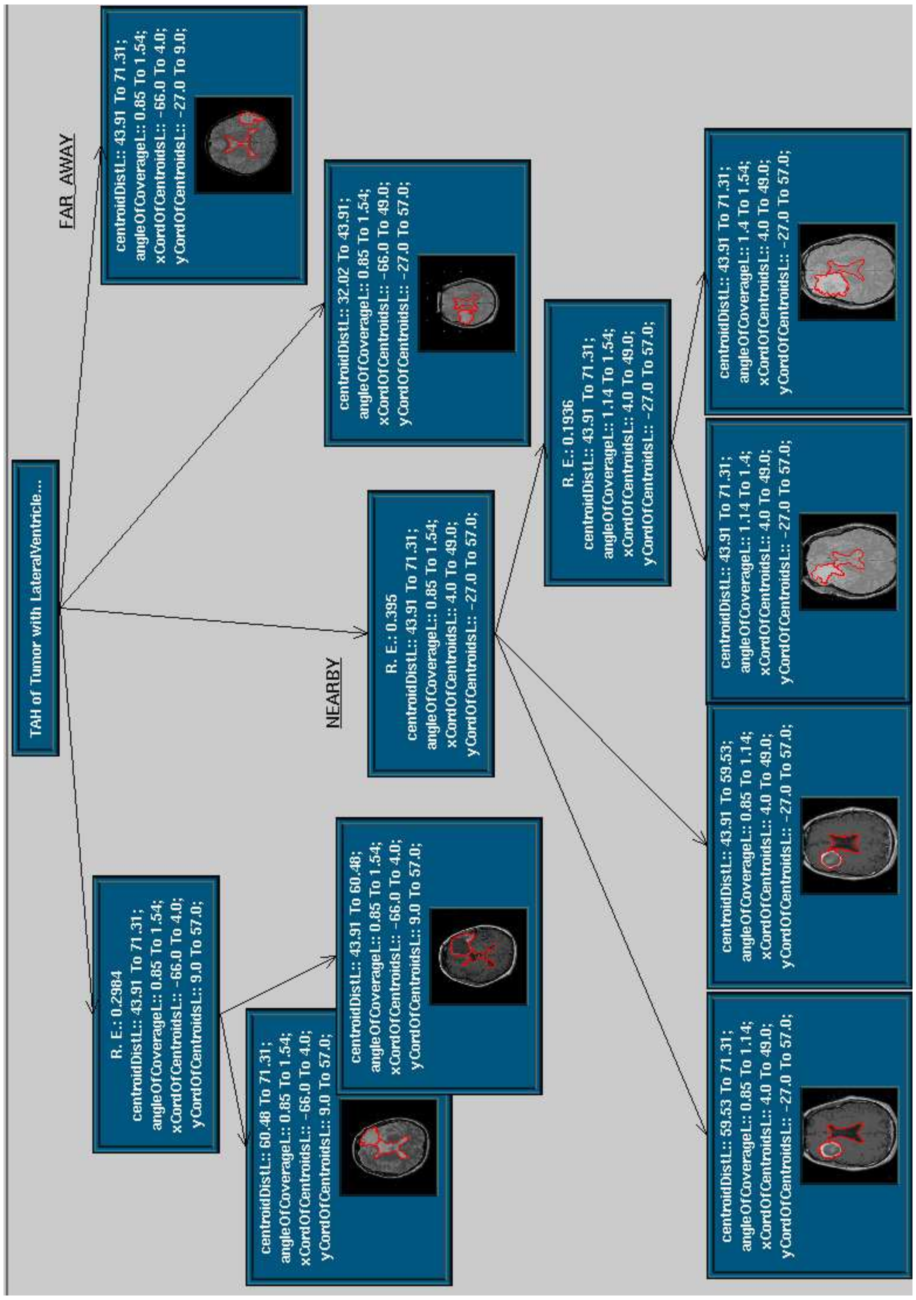


Figure 6 The MDISC-generated TAH (based on d_c, θ_c, x_c, and y_c) for representing the spatial relationship between tumor and lateral ventricle

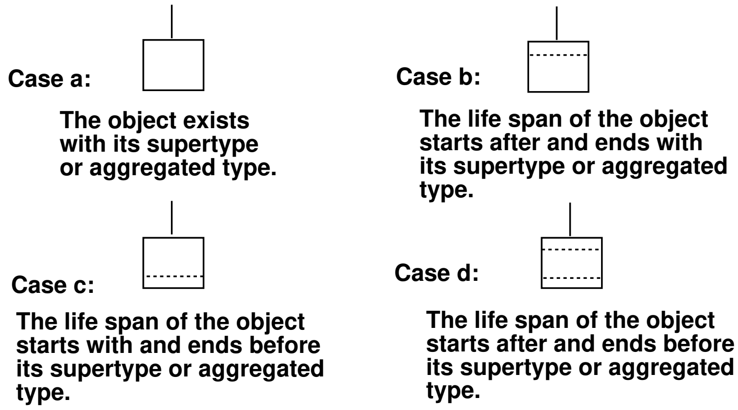


Figure 7 Temporal object constructs

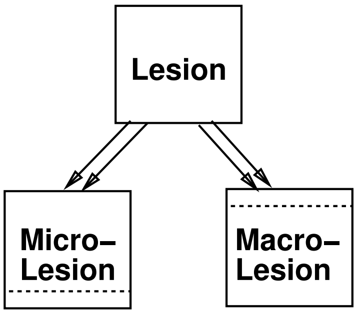


Figure 8 Temporal object constructs of lesion. Double arrows imply inheritance (is_a) relationship.

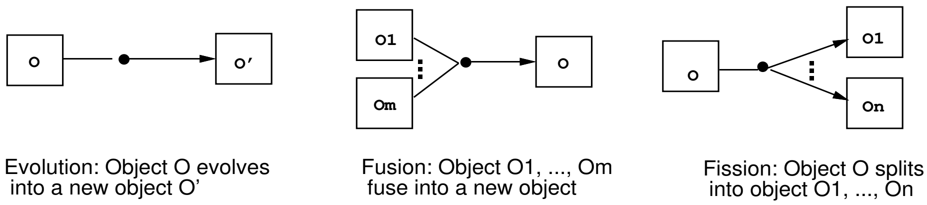


Figure 9 Evolutionary object constructs

objects using *stages* and the *evolution*, *fission*, and *fusion* among stages. A stage represents a distinguishable development status of an object, and a transition between stages represents a significant change in the entropy or growth status of the evolutionary object. Evolution, fusion, and fission are the three possible types of transitions among the evolutionary stages (Figure 9).

1. Evolution: The characteristics of an object may evolve with time. For example, a micro-lesion evolves into a macro-lesion over a period of time, as shown in Figure 10.
2. Fusion: An object may fuse with other objects to form a new object with different characteristics than either of the constituent objects.
3. Fission: An object may split into two or more independent objects.

The clinical manifestations of a lesion depend on its position with respect to its surrounding brain structures. As an example, a brain lesion is modeled with the temporal, evolutionary, and spatial object constructs as shown in Figure 10. Micro-lesions develop **separately** from (modeled in Figure 5) the lateral ventricle. Micro-lesions may also **evolve** (modeled in Figure 9) into macro-lesions which **invade** (modeled in Figure 5) the symmetrical lateral ventricle and make the lateral ventricle become asymmetrical.

Based on the model constructs, we shall present a query language that can express the spatial evolutionary content of the images. The model shown in Figure 10 will help formulate the example queries in Section 5.

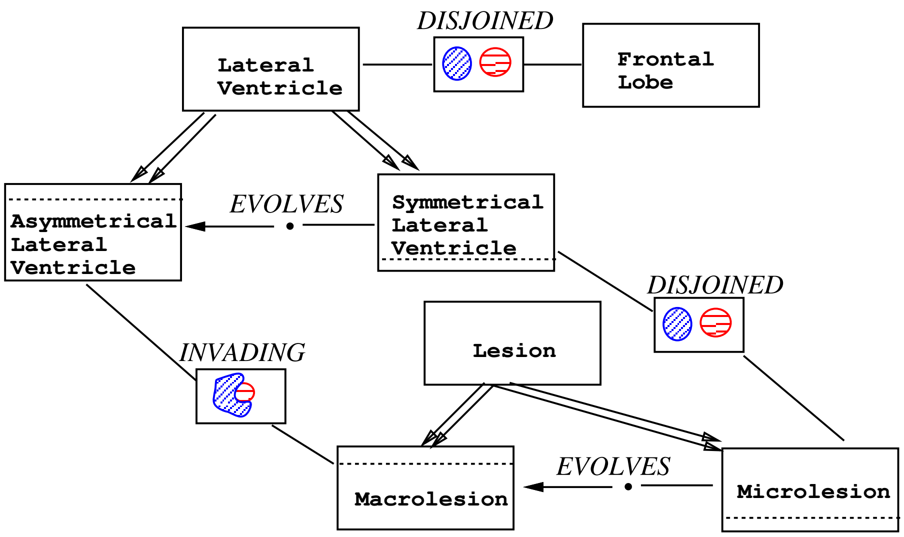


Figure 10 The temporal evolutionary and spatial relationships for the growth of a brain lesion.

5 THE KNOWLEDGE-BASED SPATIAL EVOLUTIONARY QUERY LANGUAGE (KEQL)

Conventional query languages lack the capabilities to query on spatial evolutionary characteristics of medical images [21, 37, 28, 36]. To remedy this, we propose a declarative spatial evolutionary query language, KEQL, which operates on the spatial evolutionary domains of the medical images. In addition to alphanumeric predicates, KEQL contains constructs to specify spatial, temporal, and evolutionary conditions with approximate and conceptual terms.

A KEQL query consists of the following five optional clauses:

```
[SELECT clause]
[WHERE clauses]
[Operations]
```

The `SELECT` clause selects the desired data items. `WHERE` clauses describe the selection criteria using alphanumeric predicates, conceptual spatial predicates, predicates describing the evolutionary processes among object types, and predicates for selecting appropriate snapshot(s) of the database. `Operations` specify the system or user-defined operations such as display, rotate, and superimpose on the selected data.

5.1 SELECT and Operations

The SELECT clause identifies the attributes and object types to be retrieved for the query.

The operations in the `Operations` clauses can either be system-defined or user-defined data, image manipulation, or visualization operations [6], such as movie_loop, display, contour, rotate, and superimpose.

5.2 WHERE

WHERE clauses are classified into the following categories.

Symbolic/Semantic Spatial Relationship Predicate

The user can query on the spatial relationship among objects. For example, to retrieve images of the macro-lesion *nearby* the lateral ventricle, the user enters:

```
WHERE Macro-lesion Nearby LateralVentricle
```

To avoid ambiguity in specifying the operators, a pull-down menu is available that displays the available spatial semantic operators, as in the spatial relationship description table (Table 2), for the user to select to use in the query.

Spatial Feature Predicate

Spatial feature queries involve the spatial characteristics of the image objects. By inspecting the object classes, the user knows what spatial features are available for querying, and hence, if the user wants to retrieve images of macro-lesions with diameter equal or larger than 25mm, the user enters the following predicate:

```
WHERE Macro-lesion diameter >= 25mm
```

The user may also use the $\sim=$ operator to represent *"approximately equal"* in the spatial feature predicate, and the query processor can automatically interpret the predicate to include a suitable value range based on the specific feature and query context. The value range is discovered by the DISC algorithm [9] which provides a different value range for different feature value based on the value distribution of the feature. For example, if the user wants to retrieve the images with micro-lesions of approximately 9mm in diameter, the user writes the following predicate:

```
WHERE Micro-lesion diameter ~= 9mm
```

Based on the result of running DISC on `Micor-lesion.diameter`, 9mm will be relaxed to a suitable value range (such as 8mm to 12mm).

Predicate with Conceptual Terms

Spatial queries may involve conceptual terms, such as *large* and *small* in the query. For example:

```
WHERE Macro-lesion.size IS 'large'
```

A pull-down menu is also used to display the available conceptual terms for the specified object attribute in the predicate as in the shape feature description table (Table 1). The conceptual term is to be interpreted by the knowledge residing in the KL.

Similar-to Operator

Similar-to operators compare the similarity of image objects based on shape or spatial relationship features. The similar-to predicate first selects a target image or object by constraining one of the operand variable to a specific image or object instance (e.g., `Image1.patient.name = "Mary Doe"` in the following example) and then uses the `BASED_ON` subclause to specify the shape features and/or specific spatial relationships between objects that characterize the target image similarity for the *similar_to operator* [12, 19]. For example:

```
WHERE Image1 SIMILAR_TO Image2 (Image1.patient.name = "Mary Doe")
         BASED_ON macro-lesion.size and macro-lesion.location
```

If no `BASED_ON` subclause is presented, then the features supplied in the user model will be used (see next Section).

Temporal Relationship Predicates

Temporal relationship predicates select the appropriate snapshot of the data of interest at particular points or periods in time. Temporal relationship predicates include:

- Temporal functions

  Temporal functions manipulate the time points: `START_TIME`, `END_TIME`, `RECORD_TIME`, and `EVENT_TIME`. They are the operators used in the query language to retrieve the start, end, record, and event times of objects.

- Temporal ordering functions

  Temporal ordering of an object history sorts the object versions in ascending order based on their time stamps, so that retrieval of object versions in a specific order can be identified. These functions include `FIRST`, `LAST`, `N_th`, `PRIOR`, and `NEXT`.

- Temporal interval comparison operators

To specify a more complex temporal condition, the interval specified in the temporal relationship predicates may be subjected to temporal interval comparison operators [37, 28], such as `PRECEDES`, `FOLLOWS`, `CO-OCCURS`, and `DURING`. These operators specify how the intervals following the temporal relationship predicates are related to some other time intervals. The time point comparison operators such as `BEFORE` and `AFTER` are also included.

Evolutionary Predicates

The evolutionary predicate describes various evolutionary processes on a set of evolving object sequences. The evolution of a brain abnormality, such as a lesion, provides a good example of the use of evolutionary operators. The evolution of the lesion can be described as evolving from a micro-lesion to macro-lesion. There are three kinds of evolutionary processes in general: `EVOLUTION`, `FUSION`, and `FISSION`.

The condition "$object\_type_1$ `EVOLVED_FROM` $object\_type_2$" selects all evolutionary sequences in which an object instance of $object\_type_2$ evolves into an object instance of $object\_type_1$. For example, the following evolutionary predicate applies the evolution operator `EVOLVED_FROM` to retrieve sequences where macro-lesions evolve from micro-lesions:

```
WHERE       macro-lesion EVOLVED_FROM microadenoma
```

All the growing instance sequences of lesion which developed from micro-lesion to macro-lesion will be retrieved.

For a more complete discussion of fusion and fission modeling constructs, the interested reader is referred to [11, 10].

5.3 Sample Queries

In this section, we shall present a set of sample queries to illustrate how to use KEQL constructs to express certain clinical queries associated with radiographic findings.

Query 1: "Find patients with similar brain lesions to the patient named `'Mary Doe'` *based on the lesion size and whose lesion location is* `NEARBY` *the lateral ventricle"*

```
SELECT   I1
FROM     Image I1, IT
WHERE    I1 SIMILAR_TO IT (IT.patient.name = 'Mary Doe')
           BASED_ON (IT.lesion.size,
                     IT.[lesion,lateral_ventricle].(x_c, y_c, θ_c, d_c))
```

Query 2: "Find a `large` *lesion* `NEARBY` *the lateral ventricle"*

```
SELECT Lesion.image
FROM   Lesion, Lateral_Ventricle
WHERE  Lesion Nearby Lateral_ventricle and
       Lesion.size IS 'Large'
```

Query 3: "Find the lateral ventricle whose upper protrusion is pressed to the right"

```
SELECT Lateral_Ventricle.image
FROM   Lateral_Ventricle
WHERE  Lateral_ventricle.left_to_right_symmetry
         IS 'upper_protrusion_pressed_to_the_right'
```

The knowledge representing `upper_protrusions_pressed_to_the_right` is provided in Figure 3.

A brain surgeon wishes to retrieve patients in the database whose brain MR image has similar spatial characteristics to the target MR image. The query is expressed as follows:

Query 4: "Find images in the database that have similar spatial characteristics as the image shown on the screen"

```
SELECT   P1, P1.image
FROM     Patient P1, PT
WHERE    P1.image SIMILAR_TO PT.image
                  (PT.image SELECTED_ON_THE_SCREEN)
```

`SELECTED_ON_THE_SCREEN` is a special function to specify the image on the screen as the target image for matching. The intended features and spatial relationships of *Query 4* are derived from the image content in `PT.image` and the user type (i.e., brain surgeon) according to the knowledge in the knowledge layer. The corresponding query expressed graphically is illustrated in Figure 17 in Section 7.

Query 5: "Retrieve the image frames in which a micro-lesion is nearby the lateral ventricle and approximately 9 mm in diameter. The micro-lesion evolves into a macro-lesion with diameter equal or larger than 25 mm and invades the lateral ventricle in one year"

This query involves lesion growth and its spatial characteristics with an adjacent object. Based on the object relationship captured in the data model (see Figure 10), as well as the object classes and their relationships defined for each object, *Query 5* can be expressed in the following KEQL query by using the `AFTER`, `EVOLVED_FROM`, `NEARBY`, and `INVADES` operators (the parentheses in the SELECT and DISPLAY clause enforce the pairing of the returned objects):

```
SELECT  (Micro-lesion.Image, Macro-lesion.Image)
FROM    Micro-lesion, Macro-lesion, Lateral_Ventricle LV1 LV2
WHERE   Micro-lesion EVENT_TIME + 1 year AFTER
                       Macro-lesion EVENT_TIME AND
        Macro-lesion EVOLVED_FROM Micro-lesion AND
        LV1 Nearby  Micro-lesion AND
        Micro-lesion diameter ~= 9 mm AND
        Macro-lesion Invades LV2 AND
        Macro-lesion diameter >= 25mm AND
        LV1.patient = LV2.patient
```

```
DISPLAY SEQUENCES( (Micro-lesion.Image, Macro-lesion.Image) )
```

6 KNOWLEDGE-LEVEL METADATA FOR INTELLIGENT INTERPRETATION AND ACCESS

The knowledge-level metadata consists of the classification of images based on the selected features. The classification is context and user sensitive. The criteria of our image feature clustering algorithm is to minimize the averaged pair-wise Euclidean distance of image feature values in a cluster. Such a measure, known as the *relaxation error* [9], considers both the *frequency* of the value occurrence and the difference between *values*. Based on minimizing the summed *relaxation error* of all the newly partitioned clusters in each iteration, the clustering algorithm, MDISC, recursively partitions the data set to generate a multi-attribute feature type abstraction hierarchy (MTAH). Since both the feature value distribution and the correlation among different attributes of a feature are considered in MDISC, our clustering algorithm provides better image feature classification than if we used standard deviation to represent image similarity [1].

This automatic generation of the knowledge-level metadata based on the selected features of images makes our knowledge-based query answering approach scalable.

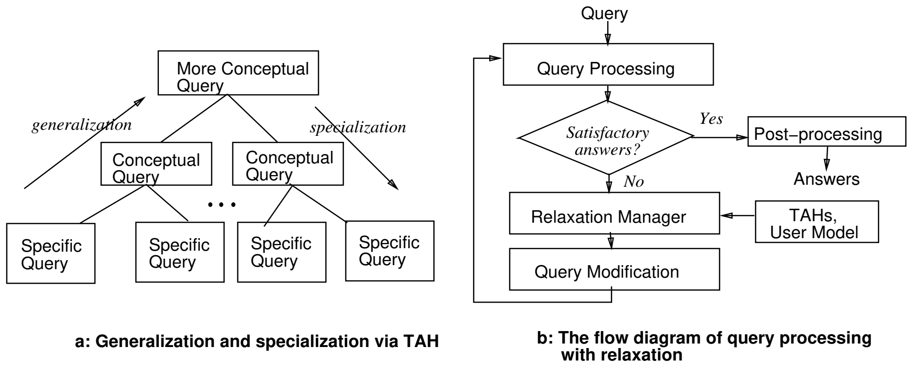


Figure 11 Knowledge-based query relaxation

6.1 Query Interpretation via TAH

The classification of image features are represented in a hierarchical representation, called *type abstraction hierarchies* [8] for processing *similar-to* and semantic predicates. The concept in the TAH nodes is represented by the feature value ranges (see Figures 3 and 6). These value ranges can be used to retrieve similar images. As shown in Figure 11.a, higher nodes in the TAH represent more generalized concepts (i.e., broader range of feature values) than that of the lower nodes (i.e., narrower range of the feature values). Based on the existing values of the representative spatial relationship features, MDISC can be used to classify image features to represent these semantic spatial relationship operators. The resultant TAH nodes are labeled by domain experts with the conceptual terms (e.g., `Nearby`, `Far_Away`, `Large`, `Small`, `upper_protrusion_pressed_to_the_right`) to represent the specific domain concept of the images under the node and the value range is used as the query constraint to retrieve the images satisfying the conceptual predicates[1] These conceptual terms are selected from the *domain dependent descriptive ontologies*. For example, Figure 5 is an ontology for various spatial relationships applicable to oncological medical images. A descriptive ontology in a generalization hierarchy for a specific image domain which provides 1) the conceptual terms for labeling the TAH nodes, and 2) the shape, spatial relationship, and/or temporal features for clustering the object instances. These ontologies can be obtained from generic medical terminology (e.g., UMLS [40]) and domain experts. They provide a restricted set of vocabularies for querying image content. Note that the proposed ontology for describing entities (e.g., size, color, distance) are different from the common sense generic ontologies such as Cyc and WordNet[27].

To solve a *similar-to* query whose intended similarity includes the features or spatial relationship classified by the TAH, the lower TAH nodes are attached with more specific value ranges. In solving the *similar-to* query, we shall first locate the TAH node that has a value range closest to that of the target image based on these selected (spatial relationship) features. By traversing up (i.e., generalizing) and down (i.e., specializing) the selected TAH, the feature value range in the matched TAH node can be used for retrieving similar images from the database (Figure 11.b). The traversal along the TAH nodes is controlled by the relaxation policy which is provided either directly from users or from the user model.

[1]The conceptual terms can also be labeled via other approach (e.g., neural networks, vector computation [?], etc.). Our distribution-sensitive clustering algorithm, which is based on the selected features, provides sufficient classification information for the domain experts to label the TAH nodes.

There is a *TAH directory* in the system that stores such information as object names, sets of features, spatial relationships, user type of the TAH. Based on this information, the system (or user) selects and retrieves the appropriate TAHs for processing the query. If the retrieved TAH does not match the user's specification, the TAH can be edited to meet the user's requirement.

The time complexity to generate a multi-attribute hierarchy by MDISC is $O(mnlog(n))$, where m is the number of attributes and n is the number of distinct instances used in generating the TAH [9]. Our experiment reveals that generating a MTAH with about one hundred images based on four features takes a fraction of a second's processing time on a Sun Sparc 20 workstation.

6.2 User Model

The user model specifies the user preference for customizing a query so that it is context and user sensitive. The preference of a user under a query context is characterized in a user profile.

In a user profile, the query context to be matched is represented by two different types of objects: *mandatorily matched objects* and *optionally matched objects. Mandatorily matched objects* of a user profile specify the objects that must be matched with a given query for the user profile to interpret the query. *Optionally matched objects* provide guidelines for adding additional object feature constraints when the corresponding objects are matched. Optional matching permits a partial matching of user profiles based on the objects appearing in the query and increases the matching occurrences.

The user preference specifies the set of interested features and spatial relationships for a given query context, and the policies for relaxing query conditions when no satisfactory answer is found.

The relaxation policy describes how to relax the selected TAHs when no satisfactory answer is found, since classifying images may involve multiple attributes as well as multiple TAHs to provide more efficient and accurate query answering. Control operators such as the *relaxation order* (e.g., which TAHs should be relaxed first), non-relaxable features, preference list, and *relaxation level* can be used to provide relaxation control.

For example, given MR brain images with lesion(s), the features that are of interest to a typical brain surgeon include the lesion locations and the spatial relationships with other objects in the brain (such as lateral ventricles) as shown in Figure 12. The information in the user model can be used for selecting the interested features and the relaxation policy if they are not specified in the query (see *Query 4*). For a given image, different types of users (e.g., radiologists, surgeons, and clinicians) may have different interests which can be specified in the user model.

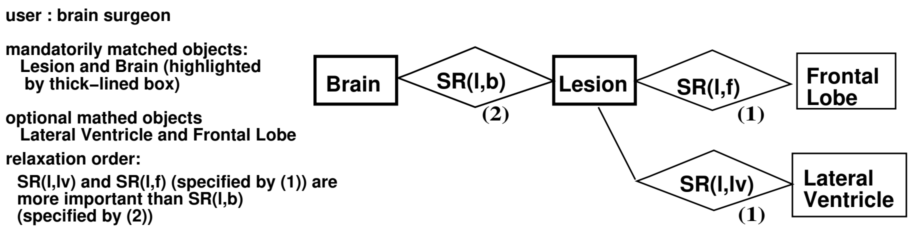


Figure 12 A user profile for brain surgeons

7 KNOWLEDGE-BASED QUERY PROCESSING

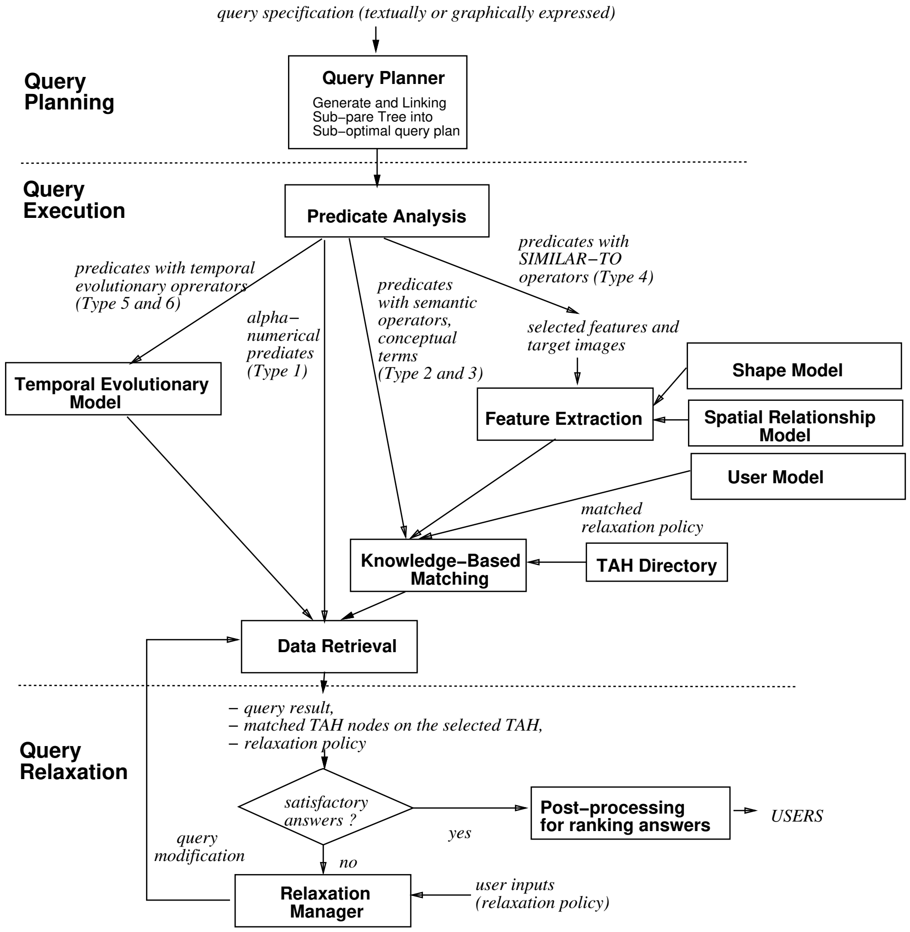


Figure 13 Knowledge-based query processing flow diagram for KEQL queries

The knowledge-based query processing is divided into three phases: *query planning, query execution, and query relaxation* (Figure 13) as discussed in the following sections.

7.1 Query Planning Phase

A KEQL query is parsed and transformed into a *query execution graph*, a directed acyclic graph (dag) [4] representing the execution plan for the query. The first step is to generate a *sub-parse-tree* for each predicate. Then a query execution graph is formed by linking together the sub-parse-trees by matching their input and output data variables.

There are six types of predicates supported in the KEQL.

- *Type1: alphanumeric predicates*

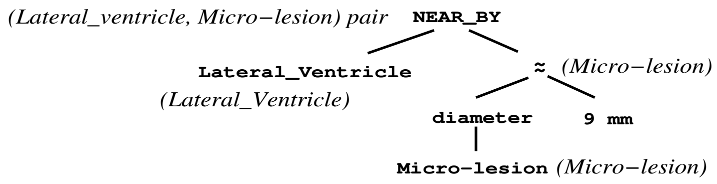
  Alphanumeric predicates contain arithmetic operators such as $>$, $=$, and $<$ which specify the constraints for the objects. The input of this type of predicate is a set of extracted objects as specified in the predicate. The output is a set of objects satisfying the specified alphanumeric predicates.

- *Type 2: predicate with conceptual terms*

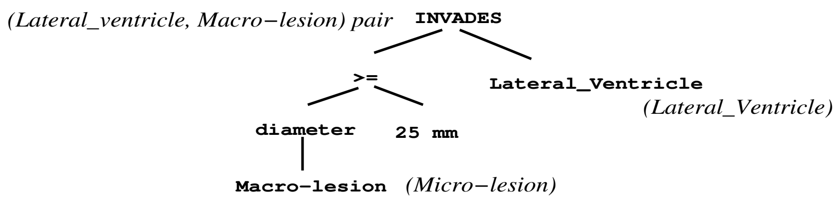
  This type of predicate is similar to the alphanumeric predicate except that the constraints are specified by the conceptual terms (e.g., *large* tumor.size) [19]. The output is the set of objects that satisfy the conceptual terms.

- *Type 3: semantic spatial relationship predicates*

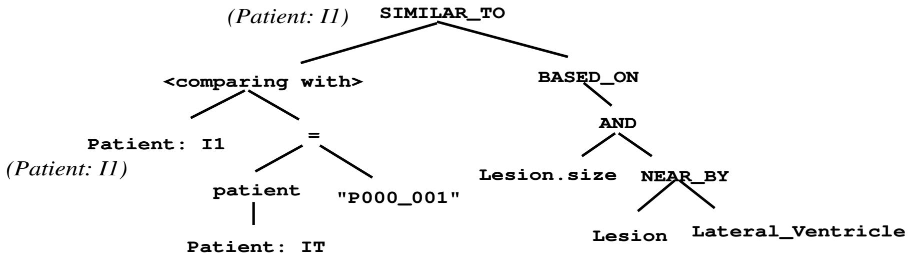
  Semantic spatial relationship predicates are used to specify the spatial relationship constrains for a pair of objects [19]. The input is two sets of spatial object instances which form a set of instance pairs of (spatial) objects for testing against the specified spatial relationship operator. The output is a set of object instance pairs from the two input sets that satisfies



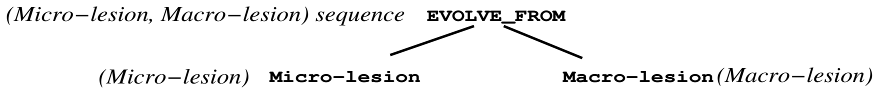
(a) Sub−parse−tree of semantic spatial relationship predicate (type 3 predicate)



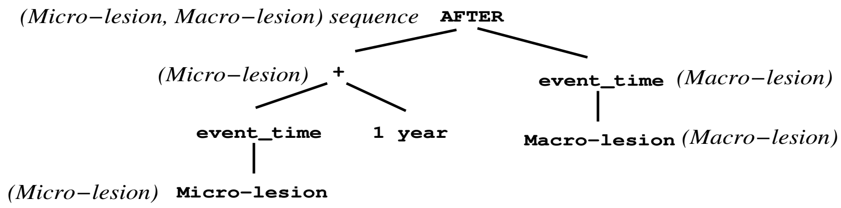
(b) Sub−parse−tree of semantic spatial relationship predicate (type 3 predicate)



(c) Sub−parse−tree of similar−to predicate (type 4 predicate)



(d) Sub−parse−tree of evolutionary predicate (type 5 predicate)



(e) Sub−parse−tree of temporal relationship predicate (type 6 predicate)

Figure 14 The sub-parse-trees for different types of predicates. (a), (b), (d), and (e) are based on predicates from *Query 5* and (c) is based on the predicate from *Query 1*. The italic strings specify the format of the input and output variables and data type via variable populating for linking the sub-parse-trees together.

the specified spatial relationships. Example sub-parse-tree structures of such predicates are shown in Figure 14(a) and (b).

- *Type 4: similar-to predicates*

  In this type of predicate, a set of objects or image instances is compared with a target object or a target image instance [19]. The target image(s) can be derived from the additional predicates in the query (e.g., patient name is *Mary Doe*). The predicate may also specify the features or spatial relationships characterizing the desired similarity of the *similar-to* predicate. The inputs are the target object, the set of objects to be compared, and the *similar-to characteristics specification*. The output is the object instances similar to the target instance. The sub-parse-tree for the *similar-to* predicate is shown in Figure 14(c).

- *Type 5: evolutionary predicates*

  Evolutionary constraints are specified in this type of predicate. The input is two or more collections of objects where each collection contains object instances of the same evolutionary stage, and the objects of different stages are linked via the specified evolutionary constructs. The output is a set of evolutionary sequences satisfying the specified evolutionary operator. An example of such sub-parse-tree is illustrated in Figure 14(d).

- *Type 6: temporal relationship predicates*

  Temporal relationship predicates specify the temporal relationship of two related objects. The two sets of temporal object inputs are to be evaluated by the temporal relationship operator. The output is a set of temporal object pairs which satisfies the specified temporal relationship operator. An example of such sub-parse-tree is illustrated in Figure 14(e).

The individual sub-parse-trees are linked together to form an execution plan based on: 1) the output (variables) of a sub-parse-tree should match the input (variables) of the linked sub-parse-tree, and 2) the derived query execution plan should minimize the execution cost (or a given performance requirement).

Optimizing KEQL query patterns yields the following heuristics: *type 1 predicates are processed first, then type 2 predicates, type 3 predicates, type 4 predicates, type 5 predicates, and lastly the type 6 predicates.* Following this heuristic, the sub-parse-trees are linked together based on this pre-determined order. Figure 15 shows the result of linking the sub-parse-trees of *Query 5* (shown in Figure 14) together. When more than two separate sub-parse-trees in the lower layer output the same variables and only one sub-parse-tree in the upper layer takes the specified variable as input, depending on the logical operators (i.e., AND and OR) connecting the two corresponding lower-layer predicates, the results from the two sub-parse-trees are either unioned or joined based on their object ID or sequence ID before sending to the upper layer sub-parse-tree.

Optimizing frequently appearing query patterns with the above heuristics is based on the following observations. The semantic spatial relationship predicates and predicates with conceptual terms are converted into value ranges of the corresponding spatial features. Image retrieval using these pre-processed feature values takes less time than processing the evolutionary and temporal predicates, since traditional indexing structures can be easily built to speed up this processing. Therefore, the alphanumeric predicates (Type 1), predicates with conceptual terms (Type 2), and semantic spatial relationship predicates (Type 3) are placed at the bottom of the query plan. Processing the evolutionary and temporal predicates requires comparing the temporal occurrences of the specified objects. Navigating through the links among evolutionary objects usually requires loading the instances into memory to inspect the links. Therefore more processing time is needed. Thus, the temporal and evolutionary predicates are placed above the semantic spatial relationship predicates and predicates with conceptual terms.

Generating the optimal query execution plan depends on many factors. By expressing each type of predicate in an algebraic notation, equivalent query plans can be generated by algebraic permutation. These plans produce the same query result but with different execution cost. The one with the least execution cost can then be selected as the optimal query execution graph.

7.2 Query Execution Phase

The query processing can be performed based on the query execution plan. Depending on its type, each predicate is processed accordingly. Since the processing of alphanumeric predicates is well-known [30], we shall present the process of processing our proposed conceptual spatial, temporal, evolutionary predicates.

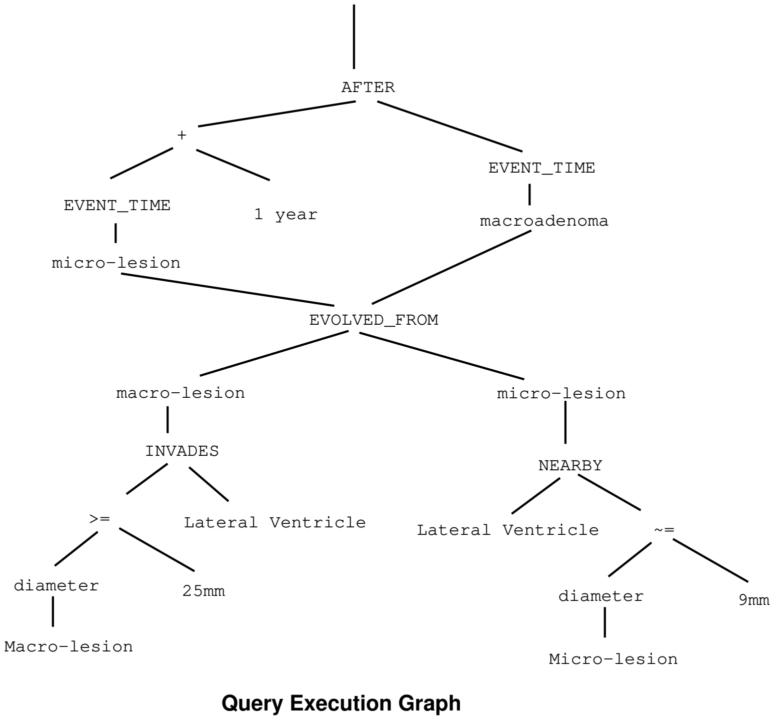


Figure 15 *Query execution graph* for *Query 5.* The execution is from bottom up.

Processing Predicates With Conceptual Terms, Semantic Spatial Relationship Operators, and Similar-to Operator

To process predicates with conceptual terms, spatial relationship operators and similar-to operators, it requires reference to domain knowledge represented in TAHs. Processing such types of predicates requires *predicate analysis and feature selection*, and *knowledge-based content matching.* In the *predicate analysis and feature selection*, based on the target image, query context, and user type, the system analyzes and selects the relevant features and spatial relationships for processing these sub-parse-trees. In the *knowledge-based content matching*, the selected features, spatial relationships, and user types are used to match the TAH(s) that resolve the conceptual terms, semantic spatial relationship operators, and/or similar-to predicates. The system uses the value ranges from the best matched TAH nodes to retrieve images from the image feature database.

To process the predicates with semantic spatial relationship operators and/or conceptual terms, the spatial relationship operators and conceptual terms are used to select the TAH(s) and the TAH node(s) for processing these queries. The TAH nodes that yield best matching of the semantic operators and/or conceptual terms (which are represented by value ranges) are used as the query conditions to retrieve the images. Their corresponding value ranges are used in the image retrieving.

For the *similar-to* predicate, the features and spatial relationships specified in the BASED_ON subclause are used to match with those of the target image. If no BASED_ON subclause is specified, then the query processor uses the default features and spatial relationships specified in the user model.

The selected features, spatial relationships, and user types are used to choose the TAH(s) for query processing. The selected TAHs are traversed to locate the node(s) whose value range is closest to that of the target image as the best matched TAH node. The set of images contained in that TAH node represents the set of images similar to the target image.

Processing Evolutionary Predicates (Type 5)

There are three types of evolutionary operators, *EVOLVE*, *FUSE*, and *SPLIT* [11, 10]. There is a method defined for each evolutionary operator which takes instances from two or more sets of object collections (*EVOLVE_FROM* takes two *input collections* and *SPLIT_TO* and *FUSED_FROM* take more than two *input collections*) as inputs and returns the set of evolving sequences that satisfy the query conditions. The matched evolving sequences will be returned as a collection of sequences. For more discussion on evolutionary query processing, the interested reader is referred to [11].

Processing Temporal Predicate (Type 6)

During the evaluation of a temporal predicate, if the two input collections are not evolutionary-related, then a cartesian product is applied to the instances of the two collections for constructing all possible object pairs for the evaluation. Every pairing of objects from the two input collections that satisfy the temporal predicate is a meaningful pairing. However, if the two input collections are evolutionary-related (i.e., two different collections of evolutionary instances from one set of evolutionary sequences), then the temporal predicate can be directly evaluated on the input evolutionary sequences embedded inside the two different collections of evolutionary instances. Temporal operators are applied to select the appropriate temporal objects where the temporal relationships of their life span satisfy the required temporal constraints.

For example, in *Query 5*, the event time of the micro-lesion and the macro-lesion are compared to determine whether the evolution took place in less than one year's time. Note only the object pairs appearing in the input evolutionary sequences are evaluated against the temporal predicate *AFTER*. The result of *Query 5* is shown in Figure 16.

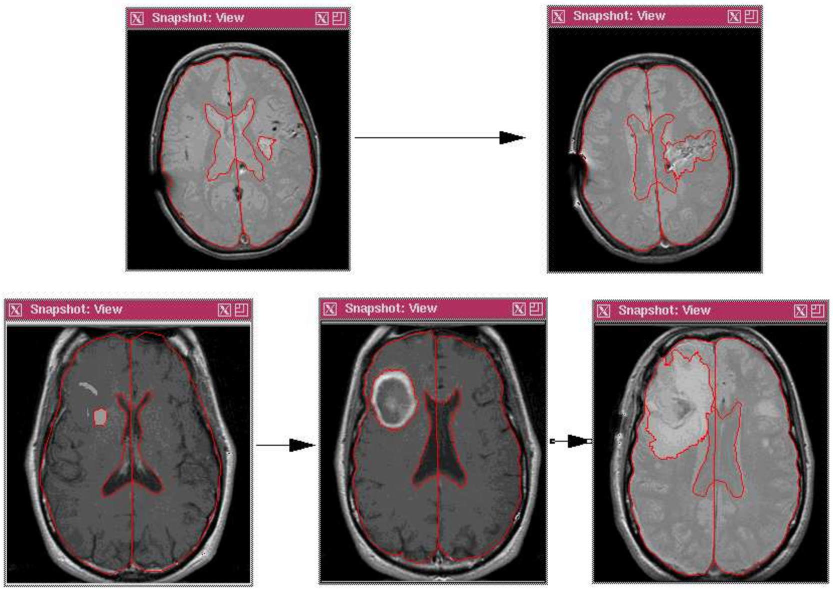


Figure 16 The returned image sequences showing the macro-lesion invading the lateral ventricle for the *Query 5*

7.3 Query Relaxation Phase

Within the query relaxation phase, when no answer is returned or the returned answer is not satisfactory, the query conditions are relaxed in accordance with the relaxation specification. For the queries requiring relaxation, an appropriate TAH is selected to relax the query condition. The relaxation is performed by traversing up and down the selected TAH starting from the TAH nodes that best match the current query context. In every relaxation iteration, the query

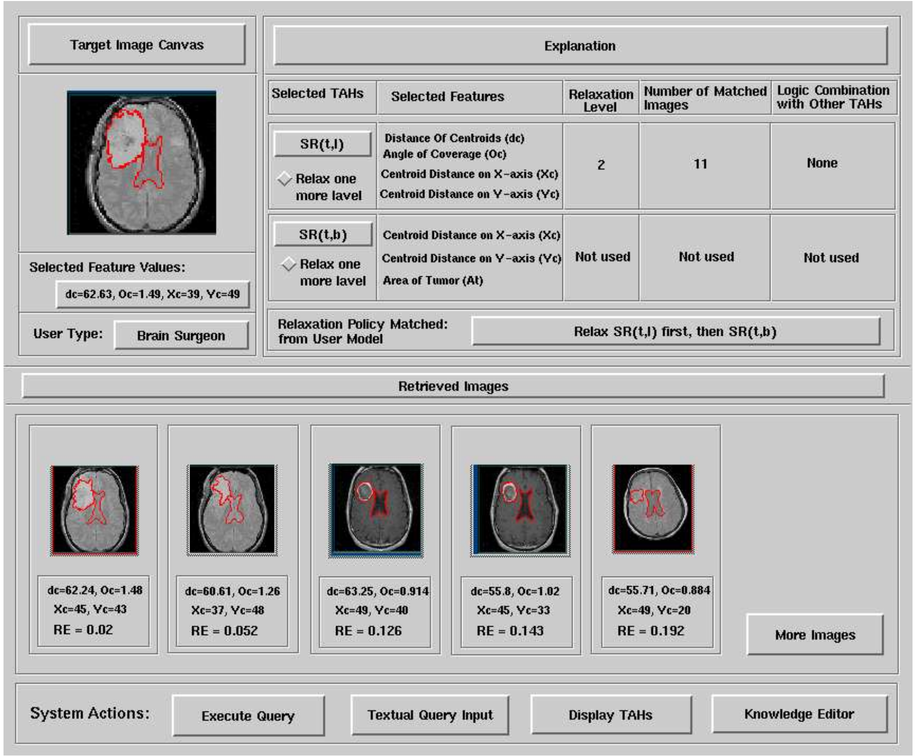


Figure 17 The graphical user interface (GUI) of the knowledge-based query answering showing the retrieved images that are most similar to the given target image

conditions are modified by the value from the TAH nodes. This relaxation process terminates when the answer set satisfies a given user requirement (e.g., number of similar images obtained, relaxation error, etc. [8]). The returned images can be ranked based on a *relaxation error (RE)*, a similarity measure based on the weighted distance of the specified shape and spatial relationship features. Since the relaxation is conducted within a sub-parse-tree, it does not alter the query execution plan.

Figure 18 illustrates the query processing for a spatial predicate with *similar-to* operator for the target image as shown in Figure 17. The system allows user relaxation control input to overwrite the one provided by the user model. Since neither relaxation control policy nor `BASED_ON` subclause is provided by the user in this example, the relaxation control and matching policy specified in the user model are used (details of the matched user model are shown in Figure 12). Thus, SR(lesion,lateral_ventricle) is the first candidate TAH to be relaxed. Based on the TAH shown in Figure 6, the value ranges for retrieving similar images are:

$(43.91 \leq SR(lesion, lateral\_ventricle).d_c \leq 71.31)$,
$(0.85 \leq SR(lesion, lateral\_ventricle).\theta_c \leq 1.54)$,
$(4.0 \leq SR(lesion, lateral\_ventricle).x_c \leq 49)$,
$(-27 \leq SR(lesion, lateral\_ventricle).y_c \leq 57)$.

These conditions correspond to the value range of the TAH node two levels higher than the matched leaf node (see Figure 6). The retrieved images are shown and ranked with the *relaxation error* attached to each retrieved image (see Figure 17). There is an *explanation* window which displays the selected features and spatial relationships used for the matching, the relaxation level, and the number of matched images for this TAH node. During the relaxation process, if the relaxation of a TAH reaches a certain *relaxation error threshold*, then the system, according to the relaxation policy, selects another TAH for relaxation. Users can also retrieve images based on complex query conditions by selectively combining the TAHs with logical operations (e.g., AND, OR, etc.).

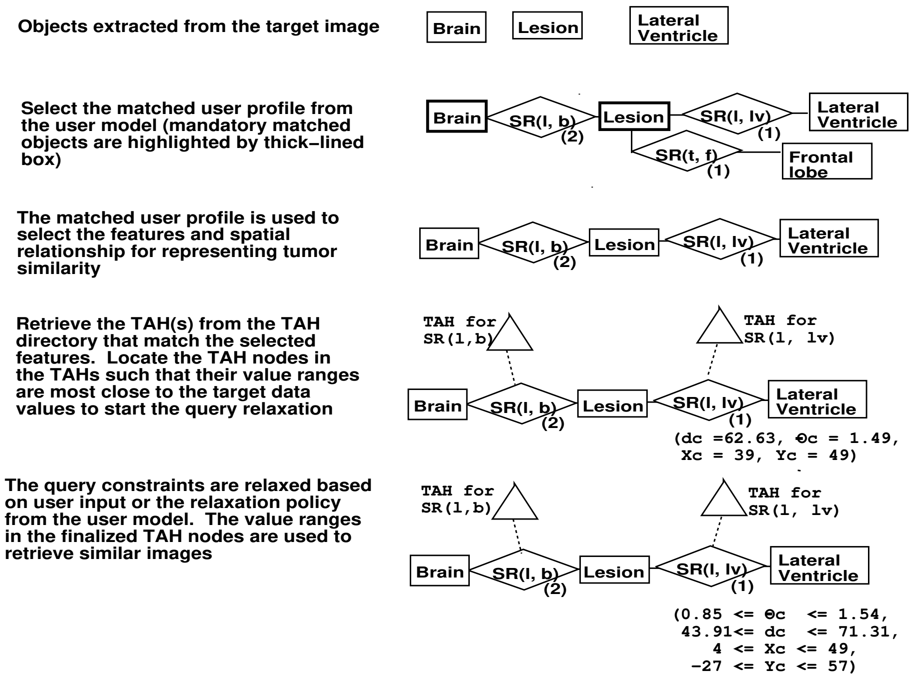


Figure 18 Processing steps for *Query 4* (the TAH for *lesion* `Nearby` *lateral ventricle* is shown in Figure 5)

8 IMPLEMENTATION

Figure 19 illustrates the overall flow and key components of our system. Our raw data set is the images stored in the UCLA Picture Archiving and Communication System [20]. These images are sent through segmentation routines to generate contours of interested objects in the images. Methods of the proposed feature extraction technique (see Section 3) compute image features and spatial relationships from the object contours. These features are stored in a *feature database* and mapped with image objects and spatial relationships in the Semantic Layer. The evolutionary constructs are used to describe the temporal characteristics of object development. In addition, the features are classified by the MDISC clustering algorithm to produce the type abstraction hierarchies required for knowledge-based query answering.

A knowledge-based spatial evolutionary query language (KEQL) is developed to express spatial evolutionary queries which provides direct manipulation of image objects.

The knowledge-based query processor parses the KEQL statement and uses type abstraction hierarchies for the approximate and conceptual matching of image content specified in the KEQL statement. The query result is returned for further presentation and visualization.

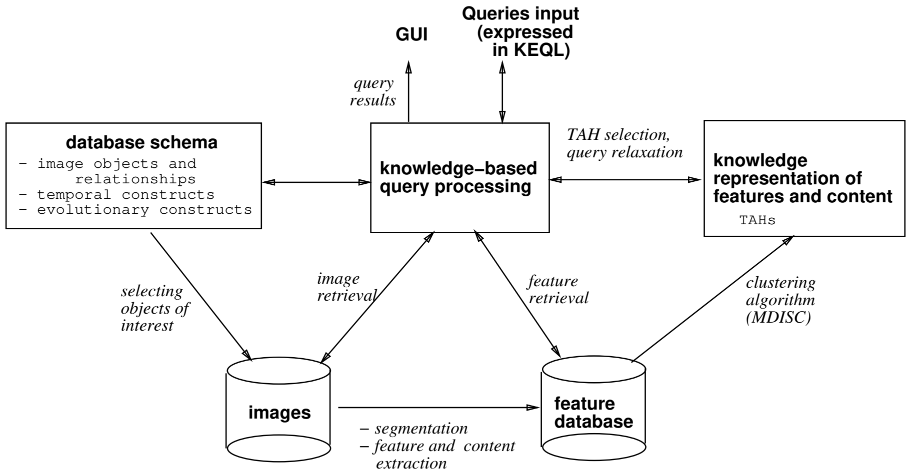


Figure 19 Data flow and system architecture.

Using Gemstone,[2] an object-oriented database, and VisualWorks as the application development tools, a prototype medical image management system [7, 19] has been implemented at UCLA to demonstrate the feasibility of the approach. Currently, the system runs on a Sparc 10 with a RAID of 16 gigabyte disk storage. Using VisualWorks, we have developed a graphical interface, MQuery [14], with pull-down menus and "point-click-drag" features for querying the database. With graphical representation for both objects (e.g., icons) and streams (used for temporal and evolutionary constructs), various KEQL predicates can be directly specified on the objects or streams to formulate a query. Buttons are included in the query window for schema navigation, object display, and query confirmation.

At present, the prototype system has a few hundred hand X-ray and MR brain image studies. Feature extraction for hand X-rays is performed automatically for patients between the ages of 3 and 11 years [32]. A commercially available MR image segmentation and rendering system from ISG Corporation is used to extract features from brain MR images. Contours of objects showing good signal-to-noise ratios can often be automatically acquired using available minimum/maximum thresholding methods. The system also includes a semi-automatic region growing program after specification of an initial seed point. Manual contouring methods are also available. A comprehensive set of tools exists to edit (remove and splice) existing contours.

[2]We selected Gemstone because it provides richer data types than relational database management systems. This is essential to developing the new modeling constructs proposed in the paper. Further, Gemstone has a gateway to Sybase, which allows us to retrieve the stored patient demographics at the UCLA medical center.

The time to process a query depends on the complexity of the query and the size of the database involved. The response time ranges from 2 seconds to 15 seconds with an average of 5 seconds. The display of images requires an additional 5 to 30 seconds, depending on the size and number of images.

The response time of a large image system depends on the size of the database and the query optimization technique used. These issues are still under investigation.

9 CONCLUSIONS

In this chapter, we presented a knowledge-based approach for retrieving images by their features and content. The model supports semantic operators (e.g., `Joined, Disjoined, Nearby, Far_Away`), similar-to operators, and conceptual terms (e.g., `Large, Small`) in the queries.

The image content is captured through three types of metadata: pixel-level metadata; spatial, temporal and evolutionary metadata; and knowledge-level metadata. A three-layered data model is proposed to organize these three types of metadata: the Representation Layer stores the pixel-level metadata; the Semantic Layer stores spatial, temporal and evolutionary metadata; the Knowledge Layer stores knowledge-level metadata. These layers integrate the image representation (i.e., generated image contours in an image) together with the knowledge required to capture and interpret the image content to provide querying by feature and content. The model provides the framework and guidelines for image preprocessing to extract features and relations of objects in the images. Our model also considers shape structure and shape features as well as spatial relationship features. These features can be automatically or semi-automatically extracted from the image contours and stored in a feature database. A new query language, KEQL is developed to express spatial evolutionary queries with conceptual and approximate terms. With semantic information captured in the model, spatial evolutionary queries are answered efficiently. Based on the specified features and spatial relationships, knowledge of the image semantics and image similarity can be generated by our conceptual clustering algorithm automatically. The knowledge is represented in a hierarchical knowledge structure, called a Type Abstraction Hierarchy (TAH). Knowledge-based query processing can be performed by generalization/specialization on the TAHs. The value ranges of the matched TAH node are used to modify the query conditions for retrieving images. A user model is introduced to allow users to customize their specifications in object matching and relaxation policy for query answering. The system also presents the quality of the approximate answers (e.g., measured in *relaxation error*) to the user. Since the feature computation and knowledge acquisition are automated, our proposed technique is scalable.

A knowledge-based medical image system, KMeD, has been implemented at UCLA to demonstrate the feasibility of our approach. The system has been built with a commercial object-oriented database, Gemstone, with ObjectWorks/VisualWorks as the data modeling and user interface development environment. The system runs on a Sun Sparc 10 workstation with a RAID of 16 gigabyte storage capacity. Our preliminary experience reveals that the proposed knowledge-based approach is a feasible and effective way to retrieve images by content for large pictorial databases, and can also be applied to other multimedia applications.

Acknowledgements

This work is supported in part by the National Science Foundation Scientific Database Initiative, Grant IRI9116849, and also in part by DARPA contract F30602-94-C-0207.

The authors would like to thank John David N. Dionisio for implementation of the graphical user interface of the query language, Kuorong Chiang and Timothy Plattner for developing the programs for generating TAHs for images, Christine Chih for her assistance in image segmentation, and Prof. Alfonso Cardenas for his stimulating discussions during the course of writing this paper.

REFERENCES

[1] J. R. Bach, S. Paul, and R. Jain. A visual information management system for the interactive retrieval of faces. *IEEE Transaction on Knowledge and Data Engineering*, October 1993.

[2] M. Bart, P. H. Romeny, L. Florack, J. Koenderink, and M. Viergever. Scale space: its natural operators and differential invariants. In *Information Processing in Medical Imaging, Proc. 12th International Conference, IPMI*, 1991.

[3] M. Brummer, A. Van Est, and W. Menhardt. The accuracy of volume measurements from MR imaging data. In *Proc. 8th Ann. Meegint SMRM*, page 610, Amsterdam, 1989.

[4] C. N. Fischer and R. J. LeBlanc, Jr. *Crafting a Compiler*. The Benjamin/Cummings Publishing Company, Inc., 1988.

[5] S. K. Chang, C. W. Yan, and D. C. Dimitroff. An intelligent image database system. *IEEE Transaction on Software Engineering*, May 1988.

[6] M. Chock, A. F. Cardenas, and A. Klinger. Database structure and manipulation capabilities of a picture database management system (PICDMS). *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6(No. 4), July 1984.

[7] W. W. Chu, A. F. Cardenas, and R. K. Taira. A knowledge-based multimedia medical distributed database system — KMeD. *Information Systems*, 20(2):75–96, 1995.

[8] W. W. Chu and Q. Chen. A structured approach for cooperative query answering. *IEEE Transactions on Knowledge and Data Engineering*, 6(5), October 1994.

[9] W. W. Chu, K. Chiang, C. C. Hsu, and H. Yau. An error-based conceptual clustering method for providing approximate query answers. *Communications of ACM, Virtual Extension Edition (http://www.acm.org/cacm/extension)*, December 1996.

[10] W. W. Chu, I. T. Ieong, and R. K. Taira. A semantic modeling approach for image retrieval by content. *Jounral of VLDB*, 1994.

[11] W. W. Chu, I. T. Ieong, R. K. Taira, and C. M. Breant. A temporal evolutionary object-oriented data model and its query language for medical image management. In *Proceedings of the 18th International Conference on Very Large Databases*, pages 53–64, Vancouver, Canada, August 1992. Morgan Kaufmann Publishers, Inc.

[12] W. W. Chu, H. Yang, K. Chiang, et al. A scalable and extensible cooperative information system. *Journal of Intelligent Information Systems*, 6(6), 1996.

[13] D. Daneels, D. van Campenhout, et al. Interactive outlineing: An improved approach using active contours. In *Image and Video Storage and Retrieval, SPIE*, 1993.

[14] J. D. N. Dionisio and A. F. Cardenas. Mquery: A visual query language for multimedia, timeline, and si mulation data. *Journal of Visual Languages and Computing*, 1996. Special issue on *Image and Video Databases: Visual Browsing, Querying, and Retrieval*. To appear.

[15] H. Duong, L. Sarazin, P. Bourgouin, and J. L. Vezina. Magnetic resonance imaging of lateral ventricular tumours. *Canadian Association of Radiologists Journal*, 46(6):434–442, Dec 1995.

[16] W. E. L. Grimson. *Object Recognition By Computer: the Role of Geometric Constraints*. MIT Press, 1990.

[17] W. I. Grosky and R. Mehrotra. Index-based object recognition in pictorial data management. *Comput. Vis. Graph. and Image Proc.*, Vol. 52(No. 3):416–436, 1990.

[18] A. Gupta, T. Weymouth, and R. Jain. Semantic queries with pictures: the VIMSYS model. In Guy M. Lohman, Amilcar Sernadas, and Rafael Camps, editors, *Proceedings of the 17th International Conference on Very Large Databases*, pages 69–79, Barcelona, Spain, September 1991. Morgan Kaufman.

[19] C. C. Hsu, W. W. Chu, and R. K. Taira. A knowledge-based approach for retrieving images by content. *IEEE Transaction on Knowledge and Data Engineering*, August 1996.

[20] H. K. Huang, N. J. Mankovich, and R. K. Taira. Picture archiving and communication systems (PACS) for radiological images: State of the art. *CRC Critical Reviews in Diagnostic Imaging*, 28(4), 1988.

[21] W. Kim. A model of queries for object-oriented databases. In *Proc. of the 15th VLDB*, 1989.

[22] W. Kim and H.T. Chou. Versions of schema for object-oriented databases. In *Proc. of the 14th VLDB*, 1988.

[23] M. D. Levine. *Vision in Man and Machine.* McGraw Hill Publishing Co., Inc., 1985.

[24] B.J. Liu, R.K. Taira, J. Shim, and P. Keaton. Automatic segmentation of bones from digital hand radiographs. In *Proceedings of the SPIE: Medical Imaging, Image Processing*, volume 2434, pages 659–669, February 1995.

[25] Y. Lu and R. Jain. Reasoning about edges in scale space. *IEEE Transactions on PAMI*, 14(450), 1992.

[26] D. Marr and E. Hildreth. Theory of edge detection. In *Proc.R.Soc.*, page 187, 1980.

[27] G. Miller. Wordnet: A lexical database for english. *Communications of CACM*, November 1995.

[28] S. B. Navathe and R. Ahmed. A temporal relational model and a query language. *International journal of information science*, Vol. 48(No. 2):57–73, 1989.

[29] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying images by content using color, texture, and shape. In *Storage and Retrieval for Images and Video Databases, SPIE*, 1993.

[30] J. Orenstein et al. Query processing in the objectstore database system. In *ACM SIGMOD 1992*, 1992.

[31] J. Orenstein and F. A. Manola. Probe spatial data modeling and query processing in an image database application. *IEEE Transaction on Software Engineering*, May 1988.

[32] E. Pietka, M. McNitt-Gray, M. L. Kuo, and H. K. Huang. Computer assisted phalangeal analysis in skeletal age assessment. *IEEE Trans. on Medical Imaging*, 10(4):616–620, 1991.

[33] D. W. Piraino, S. C. Amartur, B. J. Richmond, J. P. Schils, J. M. Thome, and P. B. Weber. Segmentation of magnetic resonance images using an artificial neural network. In *Proc. Ann. Symp. Comp. Appl. Med. Care*, pages 470–472, 1991.

[34] F. Rabitti and P. Savino. Image query processing based on multi-level signatures. In *Proceedings of ACM Information Retrieval*, 1991.

[35] F. Rabitti and R. Savino. An information retrieval approach for image databases. In *Proceedings of 18th VLDB Conference*, 1992.

[36] N. Roussopoulos, C. Faloutsos, and T. Sellis. An efficient pictorial database system for PSQL. *IEEE Transaction on Software Engineering*, May 1988.

[37] R. Snodgrass. The temporal query language for TQUEL. *ACM TODS*, pages 247–298, June 1987.

[38] M. Sonka, V. Hlavac, and R. Boyle. *Image, Processing, Analysis and Machine Vision*, volume 1. Chapman & Hall Computing, 1993.

[39] R. Taira and H. Huang. A picture archiving and communication system module for radiology. *Computer Methods and Programs in Biomedicine*, 30, 1989.

[40] U.S. Department of Health and Human Services, National Institutes of Health, National Library of Medicine. *UMLS Knowledge Sources*, August 1992.