Fast and Effective Retrieval of Medical Tumor Shapes

Philip (Flip) Korn, Nicholas Sidiropoulos, *Member, IEEE*, Christos Faloutsos, Eliot Siegel, and Zenon Protopapas

Abstract—We investigate the problem of retrieving similar shapes from a large database; in particular, we focus on medical tumor shapes ("*Find tumors that are similar to a given pattern.*"). We use a natural similarity function for shape-matching, based on concepts from mathematical morphology, and we show how it can be lower-bounded by a set of shape features for safely pruning candidates, thus giving fast and correct output. These features can be organized in a spatial access method, leading to fast indexing for range queries and nearest-neighbor queries. In addition to the lower-bounding, our second contribution is the design of a fast algorithm for nearest-neighbor search, achieving significant speedup while provably guaranteeing correctness. Our experiments demonstrate that roughly 90 percent of the candidates can be pruned using these techniques, resulting in up to 27 times better performance compared to sequential scan.

Index Terms—Content-based retrieval, multimedia indexing, mathematical morphology, pattern spectrum.

1 Introduction

During the past 20 years, the development of new modalities, such as Computed Tomography (CT) and Magnetic Resonance Imaging (MRI), have substantially increased the number and complexity of images presented to radiologists and other physicians. Additionally, the recent introduction of large scale Picture Archival and Communication Systems (PACS) has resulted in the creation of large digital-image databases. A typical radiology department currently generates between 100, 000 and 10 million such images per year. A filmless imaging department such as the Baltimore Veterans Administration Medical Center (VAMC) generates approximately 1.5 terabytes of image data annually.

An algorithm that would be able to search for similar shapes rapidly would have a number of useful applications in diagnostic imaging. Both "experts" such as radiologists and nonexperts could use such a system for the following tasks:

- *Diagnosis/Classification*: Distinguish between a primary or metastatic (secondary) tumor based on shape and degree of change in shape over time correlating this with data about diagnoses and symptoms. Computer-aided diagnosis will be especially useful in increasing the reliability of detection of pathology, particularly when overlapping structures create a distraction or in other cases where limitations of the human visual system hamper diagnosis [34].
- *Forecasting/Time Evolution Analysis*: Predict the degree of aggressiveness of the pathologic process or try to distinguish a particular histology based on patterns of change in shape. In this setting, we would like to find tumors in the database with a similar history as the current tumor.
- *Data Mining*: Detect correlations among shapes, diagnoses, symptoms and demographic data, and thus form and test hypotheses about the development and treatment of tumors.

Some terminology is necessary. Following [16], we distinguish between

a) range queries ("*Find shapes that are within distance ϵ from the desirable query shape.*"), and
b) nearest-neighbor queries ("*Find the first k closest shapes to the query shape.*").

An orthogonal axis of classification distinguishes between a) whole-matching and b) subpattern matching. In whole-matching queries, the user specifies an $S \times S$ query image and requires images of $S \times S$ that are similar; in subpattern matching queries, the user specifies only a small portion and requires all the (arbitrary-size) images that contain a similar pattern.

In all of the above tasks, the central problem is similarity matching: "*Find tumors that are similar to a given pattern,*" including shape, shape changes, and demographic patient data. In this paper we focus on whole-matching of similar shapes only.

This paper is organized as follows: Section 2 reviews some background material on spatial access methods and shape representation for shape indexing. Section 3 gives the problem definition. Section 4 gives the proposed solution. Section 5 presents the experimental results in terms of both effectiveness and efficiency. Section 6 examines why the proposed method is successful. Section 7 lists the conslusions and directions for future research.

- *P. Korn is with AT&T Labs—Research, Shannon Laboratory, 180 Park Ave. Rm. A101, Florham Park, NJ 07932. E-mail: flip@research.att.com.*
- *N. Sidiropoulos is with the Department of Electrical Enginering, University of Virginia, Charlottesville, VA 22903. E-mail: nikos@virginia.edu.*
- *C. Faloutsos is with the Computer Science Department, Carnegie Mellon University, Wean Hall, Rm. 4126, 5000 Forbes Ave., Pittsburgh, PA 15213. E-mail: christos@cs.cmu.edu.*
- *E. Siegel and Z. Protopapas are with the Baltimore Veterans Administration Medical Center, Baltimore, MD. E-mail: esiegel@umaryland.edu, zenonp@worldnet.att.net.*

Manuscript received 15 November 1997; revised 30 June 1998.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 107289.

2 Background

The state-of-the-art in multimedia indexing is based on feature extraction [33], [16]. The idea is to extract n numerical features from the objects of interest, mapping them into points in an n-dimensional space. Then any multidimensional indexing method can be used to organize, cluster, and efficiently search the resulting points. Such methods are traditionally called *Spatial Access Methods* (SAMs). Using this framework, a query (in "object space") of the form, *"Find objects similar to the query object Q,"* becomes the query (in "feature space"), *"Find points that are close to the query point* $\mathbf{q}$," and thus becomes a range query or nearest-neighbor query in a reduced space. A SAM can be used to quickly identify qualifying points and, from them, the corresponding objects.

In Section 2.1, we review work on SAMs, shape representation for shape-matching and feature extraction, and then some work from the database literature on general index design for scalability, drawing on the two latter topics. As was pointed out in [13], work on shape-matching falls into either of two camps:

a) one with an emphasis on the image processing and pattern recognition aspects of shape-matching, and
b) one with an emphasis on indexing and scale-up issues for large databases.

The focus of Section 2.2 is on the former, while the focus of Section 2.3 is on the latter.

2.1 Spatial Access Methods

Spatial access methods are the prevailing mechanism behind multimedia indexing. In a SAM, objects are represented by feature vectors in a multidimensional *vector space*, and searching is made efficient by organizing the objects (or decomposing the space) hierarchically. SAMs fall into one of the following broad classes:

- methods that transform rectangles into points in a higher dimensionality space [26];
- methods that use linear quadtrees [19];
- methods that use space-filling curves (e.g., z-ordering [47], Hilbert curves [14], [31]); and
- methods based on trees (e.g., the R-tree [24], the k-d-tree [5], the k-d-B-tree [51], and the hB-tree [39]).

One of the most promising approaches is the R-tree [24] and its numerous variants (e.g., the R^+-tree [56], the P-tree [32], and the R^*-tree [4]). Fig. 1 illustrates a space of objects and bounding rectangles, with its associated R-tree representation.

2.2 Shape Representation

Shape representation is an interesting enough problem to have attracted many researchers and to have generated a rich array of approaches [48]. There are two closely related problems:

a) how to measure the difference between two shapes so that it corresponds to the visually perceived difference, and
b) how to represent a single shape compactly.

Despite recognized limitations [53], [57], similarity (rather, dissimilarity) is typically measured by a distance function in a *metric space* (e.g., [29]).[1] This is primarily because metric spaces are convenient to work with, due to properties like the triangle inequality, which can be exploited for pruning candidates.[2]

With respect to representations, the most popular methods are the following:

- Landmarks, such as in [3], where information about the eyes, nose, etc., is extracted to represent a face.
- Simpler shapes, such as polygonalization [20] and rectangularization [33].
- Numerical vectors, such as
  a) samples of the 'turning angle' plot [27] (that is, the slope of the tangent at each point of the periphery, as a function of the distance traveled on the periphery from a designated starting point);
  b) some coefficients of the 2-d Discrete Fourier Transform (DFT) [13], [6] or the 2-d Discrete Wavelet Transform [30];
  c) the first few moments of inertia [17], [13]; or
  d) the pattern spectrum from *Mathematical Morphology* [43], [40] (see Appendix A).

Next, we consider approaches from the database literature for indexing based on combining a given shape representation with a SAM for fast searching in a large database.

2.3 Scalable Shape Matching

The seminal paper of [33] introduced the idea of feature extraction for indexing a large collection of shapes by decomposing them into rectangular components, representing the rectangularizations compactly as feature vectors, and then organizing the vectors in a SAM. While this approach is scalable, it is not guaranteed to be 'correct' in that the method may miss some of the qualifying objects with respect to the object distance function (i.e., it may allow false-dismissals). Several other scalable shape indexing schemes have been proposed (e.g., [45], [46], [6], but none of them guarantee against false-dismissals.

In [1], it was shown that a *contractive* mapping of multimedia objects into points in feature space guarantees that a range query issued in feature space must retrieve a superset of the corresponding query issued in object space. Following [15], we call the resulting index a GEMINI.[3] The lower-bounding ensures that there are no false-negatives.[4] Thus, the GEMINI method is correct if the object distance is lower-bounded by the distance in feature space.[5]

Mathematically, let O_i and O_j be two objects (e.g., time sequences) with distance function D_{object} (e.g., the sum of squared errors). Let $F(O_i)$ and $F(O_j)$ be their feature vectors

1. The distance between objects in a metric space is given by a distance function which is a metric, that is, for which reflexivity, symmetry, and the triangle inequality holds.
2. See [57] for a consideration of alternative approaches to distance metrics.
3. GEMINI stands for GEneric Multimedia INdexIng method.
4. False-positives are acceptable since they can be discarded in a post-processing step.
5. This is known as an *admissible heuristic* in the AI literature.

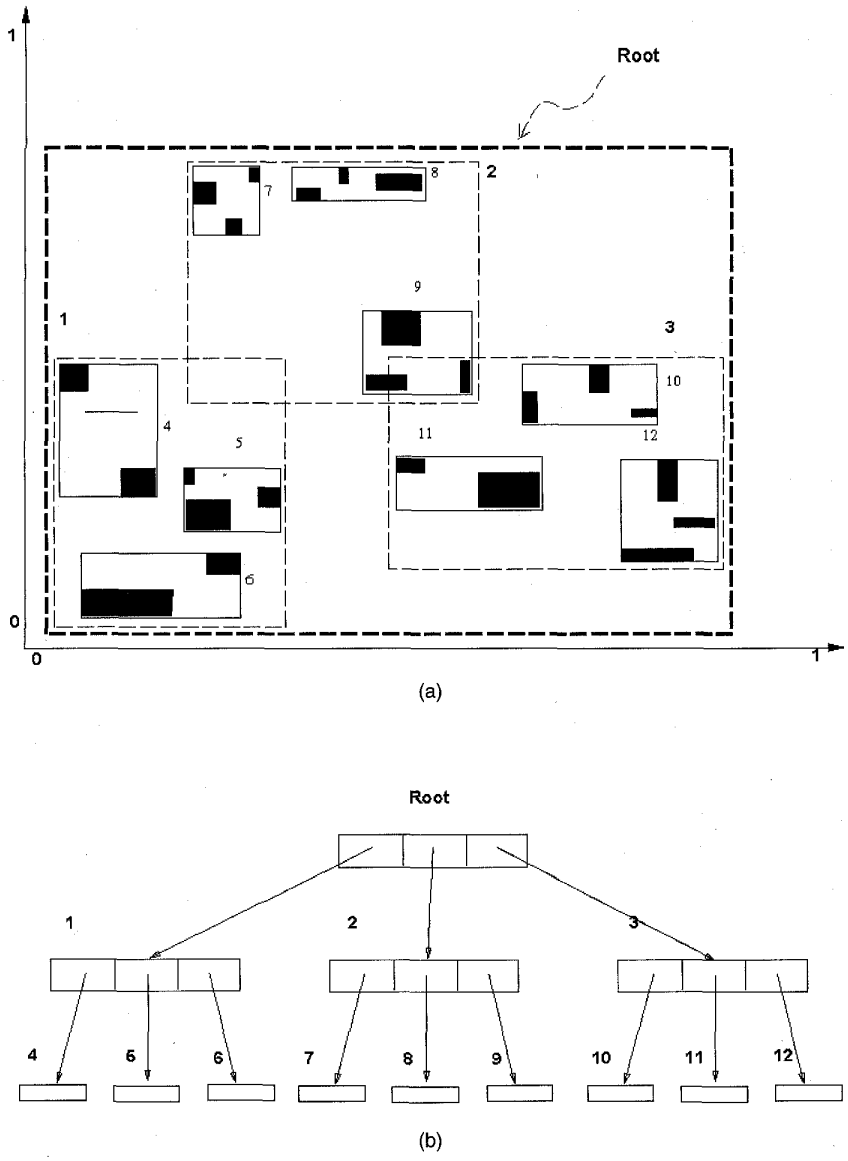


Fig. 1. An example of an R-tree built over rectangular objects: (a) objects and bounding rectangles; (b) R-tree.

(e.g., their first few Fourier coefficients), with distance function $D_{feature}$ (e.g., the Euclidean distance, again).

LEMMA 1 (Lower-Bounding). *To guarantee no false dismissals for range queries, the feature extraction function F should satisfy the following formula:*

$$D_{feature}(F(O_i), F(O_j)) \leq D_{object}(O_i, O_j) \quad (1)$$

PROOF. In reference [1]. □

Thus, for a query object Q with tolerance ϵ, range query searching involves two steps:

1) `Discard quickly those objects whose feature vectors are too far away. That is, we retrieve the objects X such that` $D_{feature}(F(Q), F(X)) < \epsilon$;
2) `Apply` $D_{object}()$ `to discard the false alarms (the clean-up stage).`

This general framework has been used in several settings:

- for time series [1], [16], [22], [50],
- images [17], [23], and
- video [36].

To date, no other admissible feature extraction approach besides GEMINI has been proposed in the database literature.

An alternative approach to multimedia search that is potentially both efficient and correct is to use metric trees (e.g., the M-tree [11], , the mvp-tree [7], the GNAT-tree [8]). Metric trees organize a collection of objects into hierarchical 'clusters' in a metric space for fast search that exploits the triangle inequality. Metric trees are particularly useful when

objects cannot be mapped well (as features) to a vector space. This is because they operate within the original object space and require only the fact that the object distance function be a metric.

SAMs have many of the same advantages as metric trees (e.g., the triangle inequality), and have the following additional benefits due to feature extraction:

- The feature vectors can be plotted in 2-d or 3-d vector space to visualize the data set (see Section 6);
- The feature vectors can be combined with other data types (e.g., text, audio) for data mining;
- As pointed out in [55], metric trees require the distance function to be fixed in advance, whereas SAMs support distance functions which can be interactively adapted at query time à la [54].

Moreover, 'industrial-strength' code for many SAM implementations (e.g., the R*-tree) is widely available. Regardless of whether we organize objects in a metric tree or organize their associated features in a SAM, the GEMINI method is applicable: it can be used in conjunction with metric trees to accelerate searching by pruning the number of object distance computations via the Lower-Bounding Lemma.

3 Problem Definition

The first challenge that arises in content-based shape retrieval is that of finding a good measure of the 'similarity' between two shapes that is appropriate for the given domain. In medical tumor applications, as well as in many other shape applications, the distance function should be invariant to rigid motions (i.e., translations and rotations). Note that scale-invariance is not a desired property in this domain, as the size of a tumor contains critical information.[6] Moreover, in our application we would like to distinguish between tumor or tumor-like shapes, called nodules, based on the 'jaggedness,' or 'ruggedness,' of the periphery, and thus would like a function that pays attention to details at several scales (see Fig. 2 for an example of a real tumor). This *multiscale* characteristic is important, especially for tumors, because the jaggedness of the periphery of a tumor contains a lot of information about it [10].[7] Thus, given two shapes, we would like to examine differences at several scales before declaring the two shapes 'similar.' In Section 3.1, we present a multiscale distance function that was proposed by domain experts in medicine (radiology) and signal processing.

Given such a distance function, the database challenge is to provide a method to access the desired shapes that is significantly faster than sequentially scanning the entire collection. This faster method, however, should not compromise the correctness of the output; rather, it should return exactly the same response set that a sequential scan would without false dismissals. We answer this challenge in Section 4.2, where we propose an indexing method, and in Section 4.3, where we prove its correctness.

6. Nonetheless, scale-invariance could be achieved by normalizing the shapes (and feature vectors) such that the area of the original shape is 1.

7. It is reported in [10] that a tumor periphery with a high fractal dimension is likely to be malignant.

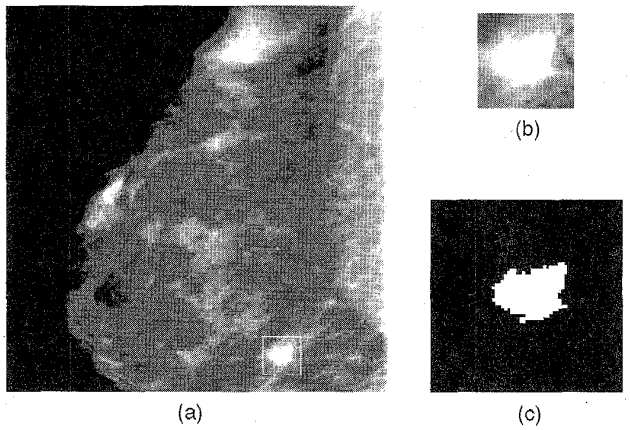


Fig. 2. A real tumor: (a) in a mammogram X-ray; (b) magnified; and (c) thresholded to black-and-white.

The experiments in Section 5 show that the method is indeed both effective and efficient.

3.1 Distance Function

Here, we describe a distance function between shapes, called the *morphological distance*, defined by experts in signal processing and medicine, which will be invariant to rigid motions, and which will 'give attention' to all levels of detail. First we present a naive approach that satisfies the translation/rotation-invariance requirement.

Given two shapes X_i and X_j, a natural design for a distance function involves penalizing the noncommon areas.

Definition 1. *Let $d(\cdot,\cdot)$ denote the area of the symmetric set difference distance measure. For $X_i, X_j \in \mathcal{X}$,*

$$d(X_i, X_j) = |X_i \backslash X_j| = |X_i \cup X_j| - |X_i \cap X_j| \quad (2)$$

It is easily seen that $d(\cdot,\cdot)$ is a distance metric over $\mathcal{X} \times \mathcal{X}$. We want a distance function that is invariant to rotations and translations. This is achieved by requiring that the two shapes are first optimally aligned by allowable motions. The process of optimal alignment of two shapes is called *registration* (see [28], [9]).[8] Formally, we have a new distance function:

Definition 2. *The **floating shape distance** $d^*()$ of two shapes X_i and X_j is*

$$d^*(X_i, X_j) = inf_{R \in \mathcal{R}}\, d(X_i, R(X_j)) \quad (3)$$

where $\mathcal{R}$ is the set of rigid motions.

The $d^*()$ distance is very natural and intuitive; however, it fails to consider details at several levels. Fig. 3 illustrates the point: X_1 is a square, X_2 is an identical square with a line segment coming out of its left side, and X_3 is identical to X_1 with a line segment cutting into it. At the current scale, the distance $d^*()$ among any pair of them is small. For example, if X_1 and X_2 are optimally aligned, making the two squares coincide, then the area of the disjoint part is the area of the protruding line segment, which is zero in the continuum.

8. An efficient way to do this is to translate the shapes to match centroids and rotate the shapes to line up their axes of least inertia.

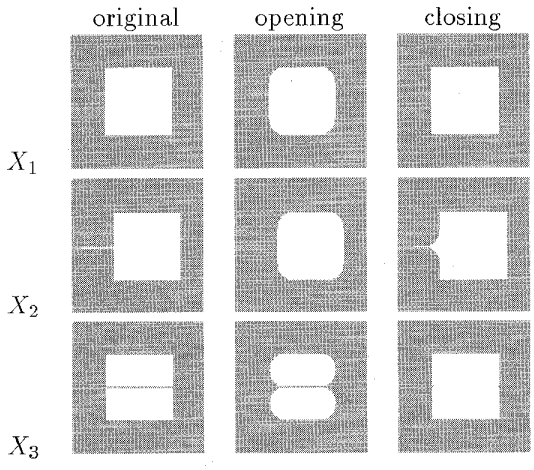


Fig. 3. Three different scales of shapes X_1, X_2, and X_3.

However, the visual difference between the two is nonzero. The same is true for X_1 and X_3. These counterintuitive results can be remedied by applying operators from mathematical morphology: The *closing* smoothes the corners of a shape; the *opening* fills the gaps of a shape (see Appendix A for a more detailed description of these and other morphological operators). After applying a closing (see Fig. 3, column 3), we see that the protruding line segment in X_2 makes its presence more obvious. Similarly, after applying an opening (see Fig. 3, column 2), the 'cut' in X_3 becomes more obvious.

Thus, given any two shapes, each opening and closing will emphasize different details of their differences, resulting in a different value of $d^*()$. The question is how to combine all of these scale-dependent penalties to arrive at a single number. A natural choice is to take an L_p norm on the differences, e.g., the Manhattan ($p = 1$), Euclidean ($p = 2$), or max ($p = \infty$) norm. Formally, the distance between shapes is defined as follows:

DEFINITION 3. *The* ***Morphological Distance***

$$d^H_{morph} : \mathcal{X} \times \mathcal{X} \mapsto \Re_+ \tag{4}$$

is defined as

$$d^H_{morph}(X_i, X_j) \triangleq \left(\sum_{m=-M}^{M} \left| d^*\left(f^H_m(X_i), f^H_m(X_j)\right)\right|^p \right)^{1/p} \tag{5}$$

where

$$f^H_m(X) \triangleq \begin{cases} X \circ mH & 1 \le m \le M \\ X & m = 0 \\ X \bullet mH & -M \le m \le -1 \end{cases} \tag{6}$$

and H is some structuring element.

For the remainder of this paper, we assume some fixed structuring element H (e.g., the unit circle) and drop these indices.

The intuitive meaning of the d_{morph} distance function is the following:

1) Compute $d^*(X_i, X_j)$; that is, take the two shapes X_i and X_j, align them optimally, and compute the area of the disjoint parts;
2) Take their closings using a disk of radius 1, 2, ... M; in each case, compute $d^*(\cdot, \cdot)$ on the resulting shapes;
3) Do the same for openings, with a disk of radius 1, 2, ..., M;
4) Compute the norm (e.g., L_1, L_2, or L_∞) on the (vector of) disjoint areas, and report it as the distance between the two shapes.

LEMMA 2. *d_{morph} is a distance metric on $\mathcal{X}$.*

PROOF. We have to verify the following properties:

- $d_{morph}(X_1, X_2) \ge 0$: follows from $\|\cdot\| \ge 0$ for any norm.
- $d_{morph}(X_1, X_2) = 0$ iff $X_1 = X_2$: trivial because m can equal 0.
- $d_{morph}(X_1, X_2) = d_{morph}(X_2, X_1)$: follows from symmetry of $\mathcal{R}$, and $d(\cdot,\cdot)$.
- $d_{morph}(X_1, X_2) \le d_{morph}(X_1, X) + d_{morph}(X, X_2)$:

$$d_{morph}(X_1, X_2) = \left(\sum_m \left\| f_m(X_1)\right| - \left| f_m(X_2)\right\|^p \right)^{1/p} \tag{7}$$

$$= \left(\sum_m \left| \left(\left| f_m(X_1)\right| - \left| f_m(X)\right| \right) + \left(\left| f_m(X)\right| - \left| f_m(X_2)\right| \right) \right|^p \right)^{1/p} \tag{8}$$

TABLE 1
SYMBOLS, DEFINITIONS, AND NOTATION USED IN THIS PAPER

symbol	*definition*
$\|X\|$	area of a shape X
H	structuring element
$f^H_m(X)$	a smoothed version of X at scale m wrt H
$\mathbf{y}^H_X$	the size-distribution of X wrt H
$d(\cdot,\cdot)$	set-difference distance between two shapes
$d^*(\cdot,\cdot)$	the floating shape distance
$d_{morph}(\cdot,\cdot)$	morphological distance between two shapes
$\delta_{gran}(\cdot,\cdot)$	granulometric distance between two shapes
N	database size (number of images)
n	number of features in feature space
m	size of structuring element ($\equiv$ scale)
a	response set size (number of actual hits)

$$\leq \left(\sum_m \left| \left| f_m(X_1) \right| - \left| f_m(X) \right| \right|^p \right)^{1/p} + \left(\sum_m \left| \left| f_m(X) \right| - \left| f_m(X_2) \right| \right|^p \right)^{1/p} \quad (9)$$

$$= d_{morph}(X_1, X) + d_{morph}(X, X_2) \quad (10)$$

Thus, the proof is complete. □

4 Proposed Solution

The problem we focus on is the design of fast and correct searching methods that will operate on a tumor database to locate the most similar shape to the query shape. The similarity is measured by the translation/rotation-invariant, multiscale distance function from Equation (5) in Section 3.1, namely, the morphological distance d_{morph}. We focus on both nearest-neighbor queries as well as on range queries. More formally, we must solve the following problem:

Given:	a collection of $S \times S$ black-and-white images, an $S \times S$ query image, and the morphological distance function from (5),
Find:	the k most 'similar' images; or all shapes within 'distance' ϵ
Such That:	there are *no false dismissals*.

The proposed solution involves feature extraction. We design features such that the feature extraction mapping is *contractive*, i.e., the distance between any two objects in feature space is no greater than their actual distance was in object space. This enables the use of the GEMINI framework [1].

Given our approach, we must find answers to the following three challenges:

1) what features to use (i.e., how to map tumor-shapes into n – d points);
2) how to prove that the above mapping is contractive, that is, it obeys Lemma 1 (the Lower-Bounding Lemma); and
3) how to use the resulting GEMINI on the feature space to answer nearest-neighbor queries with respect to the *object* distance (as opposed to the distance in *feature* space).

The proposed method is based on the pattern spectrum [42] from mathematical morphology, which we discuss in the next section. Following that, we present the proposed solutions to these three challenges in Sections 4.2, 4.3, and 4.4, respectively.

4.1 Pattern Spectrum and Size Distribution

The concept of the *pattern spectrum* as a compact shape-size descriptor has been developed in [42], based on earlier seminal work on openings of sets in Euclidean spaces called *granulometries* [44]. The importance of the pattern spectrum is that it summarizes important shape characteristics in the sense that it possesses high discriminatory power, as reported in [2], [49].[9] An equivalent compact shape-size descriptor, which is more directly usable for our application, is the *size distribution*. The size distribution contains exactly the same information as the pattern spectrum.

DEFINITION 4. *The* ***Size Distribution*** $\mathbf{y}_X^H$ *of a shape* $X \in \mathcal{X}$, *with respect to a structuring element* H *is defined as*

$$\mathbf{y}_X^H \triangleq \left[\left| f_{-M}^H(X) \right|, \ldots, \left| f_{-1}^H(X) \right|, \left| f_{-1}^H(X) \right|, \left| f_1^H(X) \right|, \ldots, \left| f_M^H(X) \right| \right]^T \quad (11)$$

with $f_m^H(X)$ *defined in* (6). *By definition,* $\left| f_m^H(X) \right|$ *is the area of a smoothed version of* X *at scale* m, *i.e.,* $\left| f_0^H(X) \right|$ *is the area of* X, $\left| f_1^H(X) \right|$ *is the area of* $X \circ H$, *etc.*

In other words, the vector $\mathbf{y}_X^H$, contains measurements of the area of X at different scales, or degrees of shape smoothing, thus constituting the size distribution. The pattern spectrum, as discussed in [42], contains exactly the same information. Its elements are backward differences of the size distribution. In other words, the size distribution can be thought of as the 'cumulative pattern spectrum.' The intuitive meaning of the pattern spectrum is the amount of detail (= additional area) that the next closing will add, or that the next (larger-radius) opening will subtract.

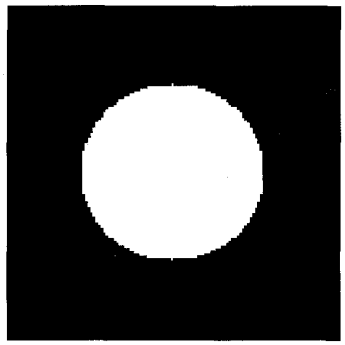
Fig. 4 shows a circle and its corresponding size distribution, as well as a butterfly shape of roughly the same area and its corresponding size distribution. Both are with respect to a a unit disc structuring element H. Notice that, while both shapes have roughly the same area at the first scale, the size distribution of the butterfly begins to trail off at higher scales (closings) while that of the circle remains constant. It is at these scales that the ruggedness of the butterfly shape, in contrast to the circle, can be detected.

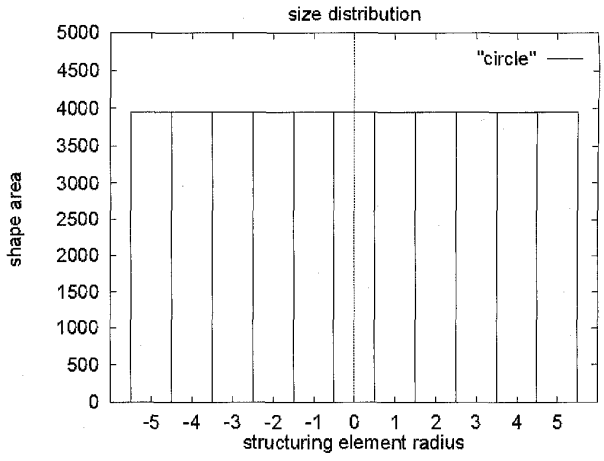
4.2 Proposed Features

Our goal is to derive features that will capture a lot of the shape information, that will be invariant to rigid motions, and that will lead to a feature-distance function that fulfills the Lower-Bounding Lemma. Given the success of the pattern spectrum as a means to capture shape information [2], [42], [41], [58], we start with an equivalent representation of its coefficients as good features. More specifically, we use the coefficients $\boldsymbol{y}_X$ of the size distribution (11). The size distribution derives its translation and rotation invariance from these invariants of the shape-area.

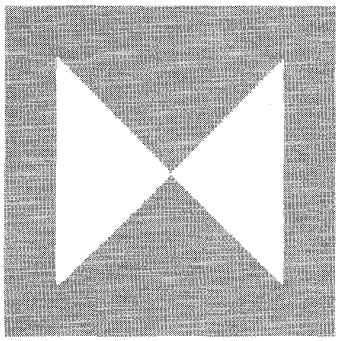
A natural choice for a feature-distance function involves 'penalizing' two shapes for differences at several scales. The question is, what is the best way to combine the penalties of each scale? We take an L_p norm on the penalties (e.g., the Manhattan ($p = 1$), Euclidean ($p = 2$), and max ($p = \infty$) norms). Formally, we define the distance between features as follows:

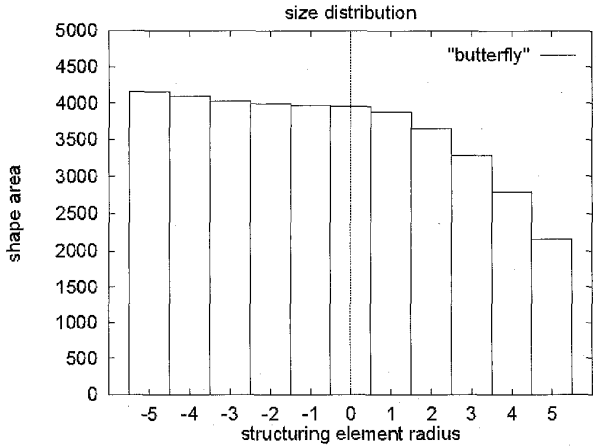
9. It has been shown that, whereas linear operators such as Fourier analysis shift and blur important shape features such as edges, the nonlinear operators of the pattern spectrum better preserve edge information, which is important for tumors.





(a)





(b)

Fig. 4. Image and respective size distribution histograms of: (a) a circle; and (b) a butterfly shape of roughly the same area.

DEFINITION 5. *The* ***Granulometric Distance*** $\delta_{gran}()$ *of two shapes* X_i, X_j *is defined as*

$$\delta^{H}_{gran,p}(X_i, X_j) = \left\| \mathbf{y}_{X_i} - \mathbf{y}_{X_j} \right\|_p \tag{12}$$

where $\|\cdot\|_p$ *is, e.g., the Manhattan* ($p = 1$), *Euclidean* ($p = 2$), *or max* ($p = \infty$) *norm.*

4.3 Lower-Bounding Lemma

The next challenge is to show that the distance in feature space (i.e., the granulometric distance δ_{gran}) lower-bounds the actual distance d_{morph}. This is necessary to guarantee correctness.

LEMMA 3 [Morphological Distance Bounding]. *The granulometric distance* δ_{gran} *lower-bounds the morphological distance* d_{morph}, *that is,*

$$\delta_{gran}(X_i, X_j) \le d_{morph}(X_i, X_j), \quad \forall X_i, X_j \in \mathcal{X} \tag{13}$$

PROOF. Observe that

$$d^*(X_1, X_2) \ge | \, |X_1| - |X_2| \, | \tag{14}$$

with equality achieved if and only if there exists some rigid motion $R \in \mathcal{R}$ which brings all points in X_2 (or X_1) in registration with points in X_1 (X_2, respectively). Then, for all m,

$$d^*(f_m(X_1), (f_m(X_2)) \ge | \, |(f_m(X_1)| - |(f_m(X_2)| \, | \tag{15}$$

and

$$\left(\sum_m \left| d^*\left(f_m(X_1), f_m(X_2) \right) \right|^p \right)^{1/p} \ge \left(\sum_m \left| \left| f_m(X_1) \right| - \left| f_m(X_2) \right| \right|^p \right)^{1/p} \tag{16}$$

Recall that the left-hand side is the definition of d_{morph} and the right-hand side is the definition of δ_{gran}. Thus, the proof is complete. □

This guarantees the *completeness* of range queries. By keeping the dimensionality of the spectra space small (say, $M = 5 \mapsto 2M + 1 = 11$ features), we can use a SAM.

4.4 Nearest-Neighbor Algorithm

We have just described a good set of features, namely, the $2M + 1$ entries of the size distribution of an image, as well as proved that the resulting feature space distance δ_{gran} lower-bounds the actual distance. Thus, the resulting GEMINI will guarantee no false dismissals for range queries.

The next problem is to find the k-nearest neighbors of a query image, given that the images of the collection have already been mapped into n-d points and organized in a SAM. Algorithms to find the k-nearest neighbors of a given point already exist, using a branch-and-bound algorithm [], [18], and have been applied to R-trees recently [52].

However, the search in feature space will return the k-nearest neighbors with respect to the *granulometric* distance δ_{gran} as opposed to the *morphological* distance d_{morph} that we really want. We propose a general nearest-neighbor algorithm (Algorithm 1) that finds the actual k-nearest neighbors from *any* GEMINI where the Lower-Bounding Lemma (Lemma 1) holds. Fig. 5 presents the Algorithm 1.

Algorithm 1 (k-nn)
/*input: query object Q; # of nn k*/
/*output: k-nearest objects X_1 to X_k*/
1. Search the SAM to find the k-nn wrt the feature distance $D_{feature}$ (δ_{gran} in our case).
2. Compute the actual distance $D_{object}(Q, X)$ ($d_{morph}(Q, X)$ in our case) for all the k candidates X, and return the maximum ϵ_{max}.
3. Issue a range query with the feature vector $F(Q)$ of the query object Q and ϵ_{max} on the SAM, retrieve all the actual objects, compute their actual distances $D_{object}()$ from Q, and pick the nearest k.

Fig. 5. Algorithm 1 uses a GEMINI to return the k-nearest neighbors, according to the object distance.

LEMMA 4. *Algorithm 1 guarantees no false dismissals for* k-nn *queries.*

PROOF. Let X_k be the shape returned as the kth nearest neighbor by the algorithm (step 3); let Y be a object that is the jth nearest neighbor (with $j < k$). Then

$$D_{object}(Q, Y) \leq D_{object}(Q, X_k) \tag{17}$$

Suppose for a moment that the algorithm *fails* to return Y. We will show that this leads to a contradiction: After the range query is issued in step 3, all eliminated shapes (including Y) must have *feature* distance $D_{feature}$ greater than ϵ_{max}, that is

$$D_{feature}(Q, Y) > \epsilon_{max} \tag{18}$$

and, from the Lower-Bounding Lemma (Lemma 1),

$$D_{object}(Q, Y) \geq D_{feature}(Q, Y) > \epsilon_{max} \tag{19}$$

However,

$$\epsilon_{max} \geq D_{object}(Q, X_k) \tag{20}$$

(since X_k was obviously retrieved by step 3). Combining the last two inequalities, we obtain

$$D_{feature}(Q, Y) > D_{object}(Q, X_k) \tag{21}$$

which contradicts (17). □

5 EXPERIMENTS

We tested both the effectiveness and efficiency of the proposed method. Obtaining real tumor X-rays is not only labor-intensive, requiring an expert to identify and segment nodules, but also legally sensitive (patient privacy, etc.). Thus, we used a popular stochastic model of tumor growth for generating artificial, but realistic, tumor shapes (see Appendix B for details). Our target class is a collection of images of tumor-like shapes. More specifically, our data consists of 128×128-pixel black-and-white images of encapsulated tumor shapes. Each image encapsulates a tumor shape that either

- grows uniformly in all eight directions;
- is biased vertically and horizontally with slower growth along the diagonals;
- is restricted along one direction (blocked by a barrier such as a bone); or
- is restricted along two directions (cone-shaped).

Fig. 12 displays a representative from each of the four classes. Within each of these four classes of growth, we varied

- the number of iterations, which affects the size of the tumor;
- the directional bias (p_{NS}/p_{EW}), which affects the ratio of height to width.

To measure the effectiveness of the morphological distance function d_{morph}, a set of queries was posed to human subjects. For each one of 10 query shapes, a total of 16 human subjects were asked to browse through a collection of 1, 000 tumor shapes and pick those which they judged to be 'similar' to the query shape. The notion of similarity was left up to each individual's discretion.

The human subjects were partitioned into two disjoint groups: a large group of 13 subjects and a small group of the remaining three subjects. The large group, collectively termed a 'superhuman,' determined relevancy democratically; i.e., only shapes that were picked by two or more members are deemed relevant. These shapes constitute the 'gold standard' in our precision/recall evaluation.

The small group competed against d_{morph}. Based on the results of the psychovisual experiments, a k-nearest-neighbor list was ranked according to the number of votes for each shape, for each of 10 queries.

Based on standard evaluation techniques for information retrieval systems [25], Fig. 6 displays the precision/recall of d_{morph} for three different norms:

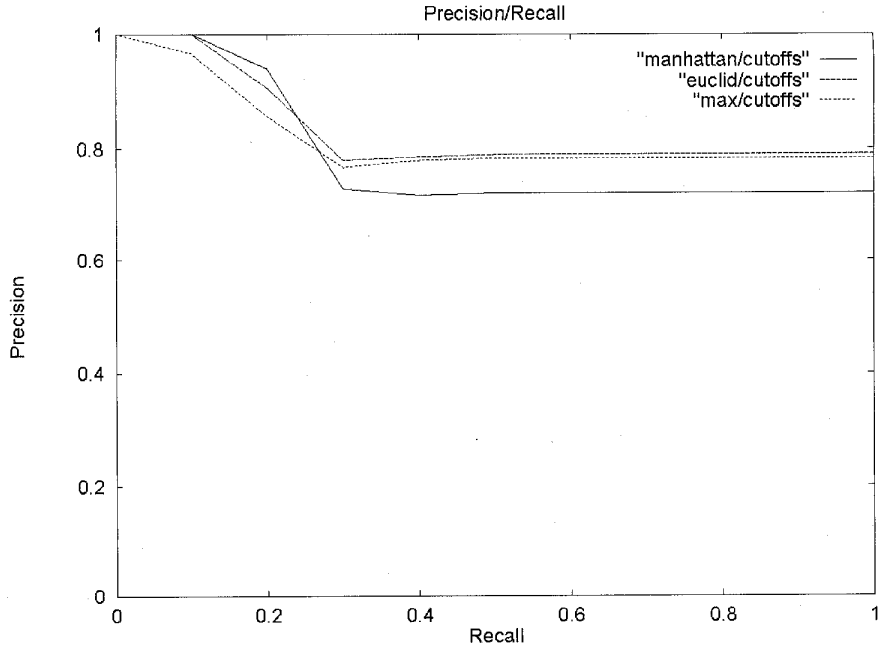


Fig. 6. Precision of d_{morph} at 11 cutoff values of recall for Manhattan, Euclidean, and max norms.

- Manhattan ($p = 1$),
- Euclidean ($p = 2$), and
- max ($p = \infty$).

Recall is the ratio of relevant items retrieved to the total number *in the database;* it measures the ability of a system to present all relevant items. Precision is the ratio of relevant items retrieved to the total number of items *retrieved;* it measures the ability of a system to present only relevant items. Notice that our proposed distance function gives approximately 80 percent precision for 100 percent recall; that is, it can return *all* of the proper items with only 20 percent false-positive errors.

Because the three norms give approximately the same results, we arbitrarily chose the max norm for the rest of our experiments. Fig. 7 compares the precision/recall of the small group with d_∞. The graph illustrates the superiority of d_∞, as the precision curve is above and to the right of the group. Fig. 7 also presents the precision/recall of two other small groups of three human subjects, whose members were selected at random, to demonstrate that the first small group is not a special case.

To test the speed of our approach, we implemented the proposed method and ran experiments. We performed experiments for varying database sizes N, by choosing N images from among a total of 20, 000. Next we describe the set up for measuring the performance of nearest-neighbor queries and for range queries.

Competing methods:

- **scan**: Given a shape for a range query, the algorithm sequentially scans the collection and computes its morphological distance from the query shape, keeping track of the shapes with the minimum distance. Because each object distance computation is CPU-intensive, it is extremely inefficient.
- **GEMINI**: Given a query shape, its size distribution is computed and submitted for a range query in a SAM which has been populated with the (n-dimensional) size distribution features of the existing collection. False-hits are discarded in a clean-up stage. In the case of a k-nearest-neighbor query, Algorithm 1 (see Fig. 5) is executed.
- **G-scan**: (A hybrid of **scan** and **GEMINI**) Given a query shape, its size distribution is computed and the results from a range query or k-nearest-neighbor query in feature space is determined sequentially from a list of feature vectors, *without the aid of a SAM.* False-hits are discarded in a clean-up stage.

Measurements:

We are interested in the response time, that is, the time until the last actual hit is returned to the user (*after* the system has discarded possible false-hits). For some settings we report actual (wall-clock) time, from the `time` utility of Unix. However, the time t_m to compute the morphological distance between two images is high (t_m = 12.69 sec on average) and shows small variance (a standard deviation of 0.036 sec). Thus, to accelerate the execution of experiments on large databases, we `time` all the other steps of the algorithms involved, and simply add a delay of t_m seconds for each morphological distance computation that we omit.

Hardware and software:

The methods were implemented in `C` and `ksh` under Unix. The experiments ran on a dedicated Sun SPARCstation 5 with 32 Mb of main memory, running SunOS 4.1.3. The disk drive was a Fujitsu M2266S-512 model 'CRANEL-M2266SA' with minimum positioning time of 8.3 ms and maximum positioning time of 30 ms.

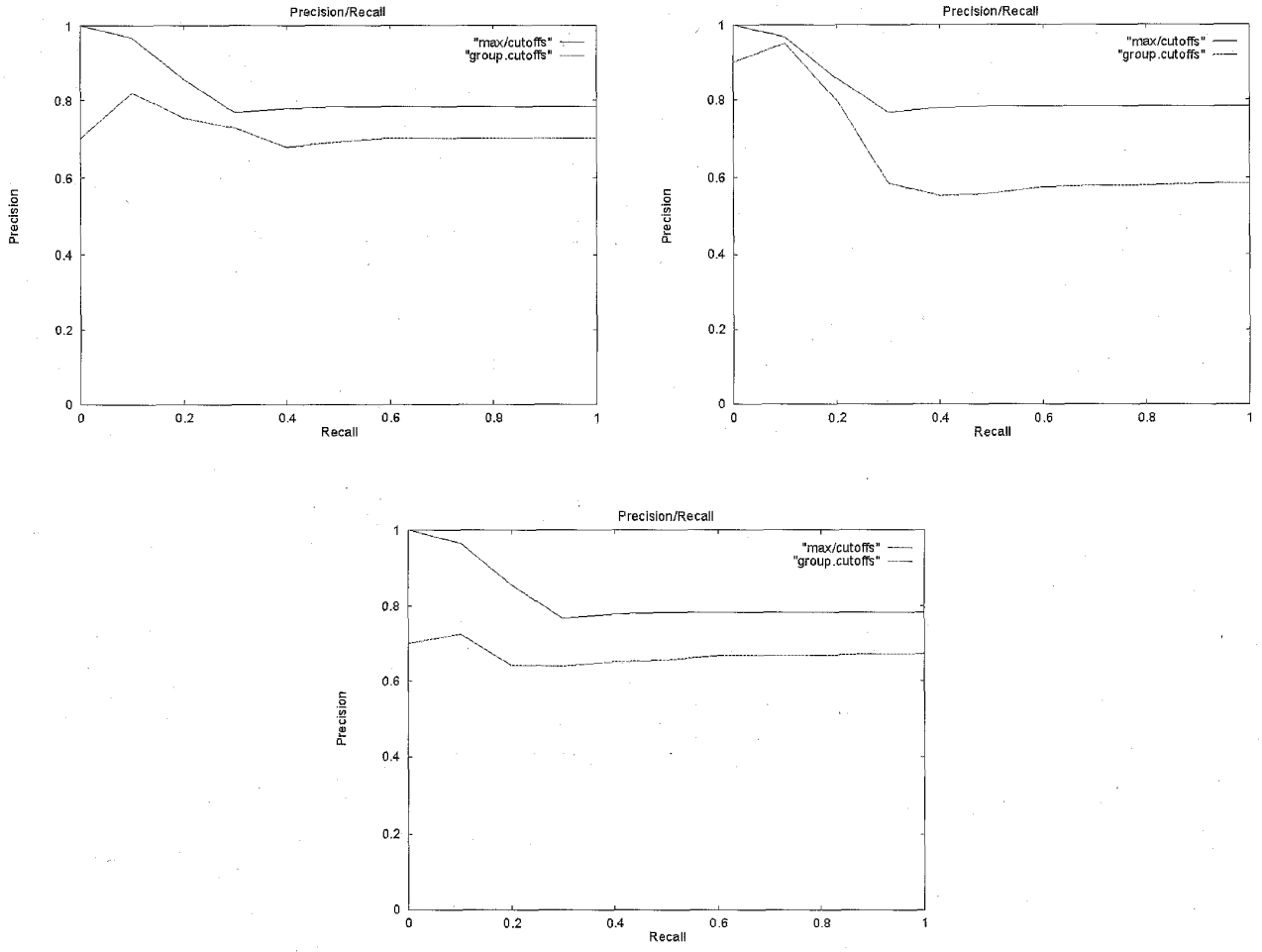


Fig. 7. Precision/recall of d_{∞} vs. group of three, for three different groups.

5.1 Nearest-Neighbor Queries

To measure the speedup of using the GEMINI approach, we examined the percentage of the original N images that were pruned due to the Lower-Bounding Lemma. Table 2 presents the results for k-nearest-neighbor queries where $k = 2$, 3, 4, and 10, where

a) $N = 1,000$,
b) $N = 10,000$, and
c) $N = 20,000$.

In all of these data sets, GEMINI was able to eliminate roughly 90 percent of the candidates.

TABLE 2
TABLE OF PERCENT DB PRUNED FOR K-NEAREST-NEIGHBOR QUERIES FOR SEVERAL N

k	$\%pruned$
2	91%
3	90%
4	86%
10	84%

(a) $N = 1,000$

k	$\%pruned$
2	91%
3	90%
4	86%
10	84%

(b) $N = 10,000$

k	$\%pruned$
2	92%
3	89%
4	87%
10	85%

(c) $N = 20,000$

For the same queries, we also recorded the average response time to demonstrate the speedup achieved by this pruning. For brevity, we do not present the results from G-scan because its results were virtually the same as those of GEMINI for data sets of the sizes we worked with.[10] Fig. 8 shows

a) the results of GEMINI compared to scan, for k-nearest-neighbor queries with $k = 10$ and increasing N, and
b) the results of GEMINI only, for $k = 2$, 3, 4, and 10.

Each data point represents the average response time (in sec) for 100 random query images taken from the database. The values of the data points for nearest- neighbor queries with $k = 10$ are given in Table 3.

Fig. 9a shows response time of scan and GEMINI as a function of k, for $N = 10,000$ and $N = 20,000$. Fig. 9b shows these results for GEMINI only. Again, each data point represents the average response time over 100 queries.

The observations are the following:

10. For larger data set sizes, however, we would expect there to be an increasing gap between the performance of GEMINI and G-scan.

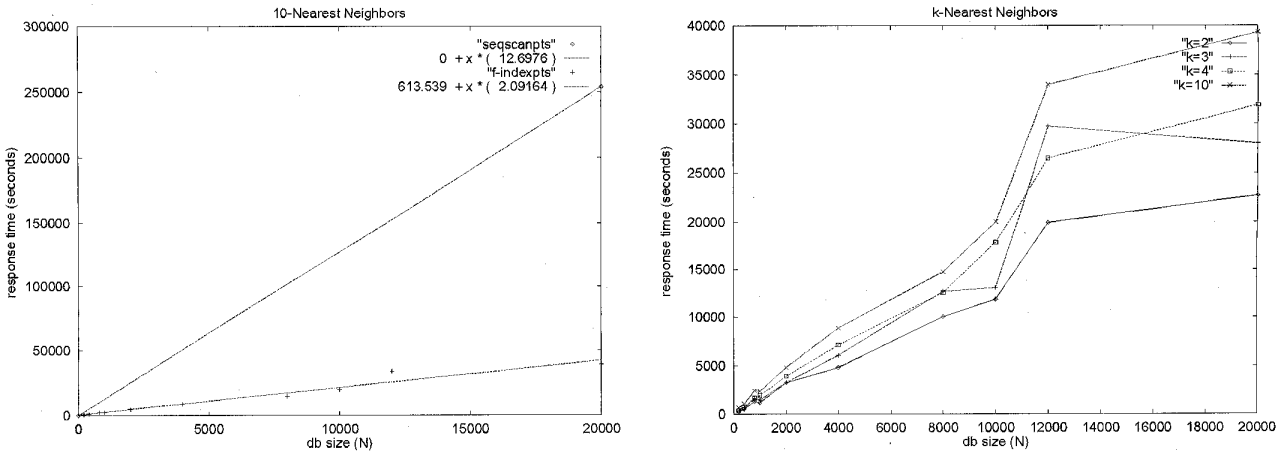


Fig. 8: (a) Response time vs. db size(N), for $k = 10$ nn queries for both **scan** and **GEMINI**; (b) Response time vs. N for $k = 2, 3, 4$, and 10.

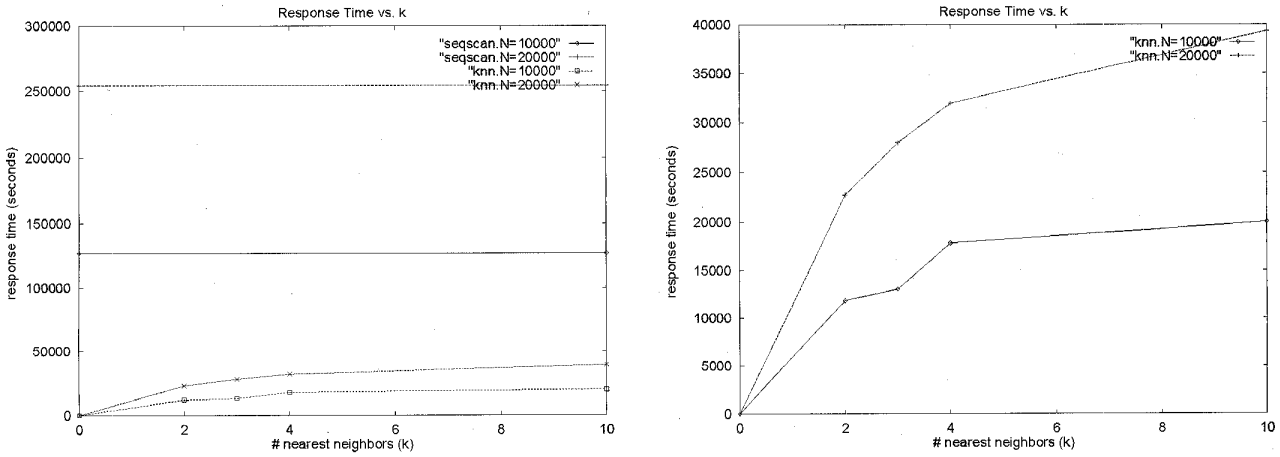


Fig. 9. Response time per query vs. k for $N = 10,000$ and $N = 20,000$: (a) **GEMINI** vs. **scan**; and (b) **GEMINI** only (magnified).

- The use of a GEMINI makes nearest-neighbor search 3-7 times faster than a sequential scan, even for a large value of k (e.g., 10), due to 90 percent of the candidates being pruned.

TABLE 3
RESPONSE TIMES FOR 10-NEAREST-NEIGHBOR QUERIES OVER VARYING N, FOR SCAN VS. **GEMINI**

db size N	**scan** time (sec)	**GEMINI** time(sec)	ratio (1) : (2)
200	2539.52	675.74	3.76
400	5079.04	1084.19	4.68
800	10158.08	2422.04	4.19
1000	12697.60	2375.44	5.35
2000	25395.20	4814.59	5.27
4000	50790.40	8855.21	5.74
8000	101580.80	14743.66	6.89
10000	126976.00	20002.95	6.35
12000	152371.20	33973.27	4.49
20000	253952.00	39340.32	6.46

- The savings of **GEMINI** compared to a **scan** seems to increase with the database size N;
- Response time grows slowly with k.

We posed 20 range queries to a database of $N = 1,000$ images, using **scan** and **GEMINI**. Fig. 10a plots the response time for **GEMINI** as a function of the response-set size a (i.e., number of actual hits, after the false-hits have been eliminated), for several values of the tolerance. It also shows the response time for **scan** for comparison, which is estimated to take 12697.6 sec. Fig. 10b shows **GEMINI** in more detail. Table 4 displays the data points in a table. The performance gap between the two methods is very large, with **GEMINI** achieving a 15- to 27-fold savings.

6 DISCUSSION

Two factors are responsible for the speedup that a GEMINI can achieve:

1) The use of feature vectors in a quick-and-dirty pruning of the database;

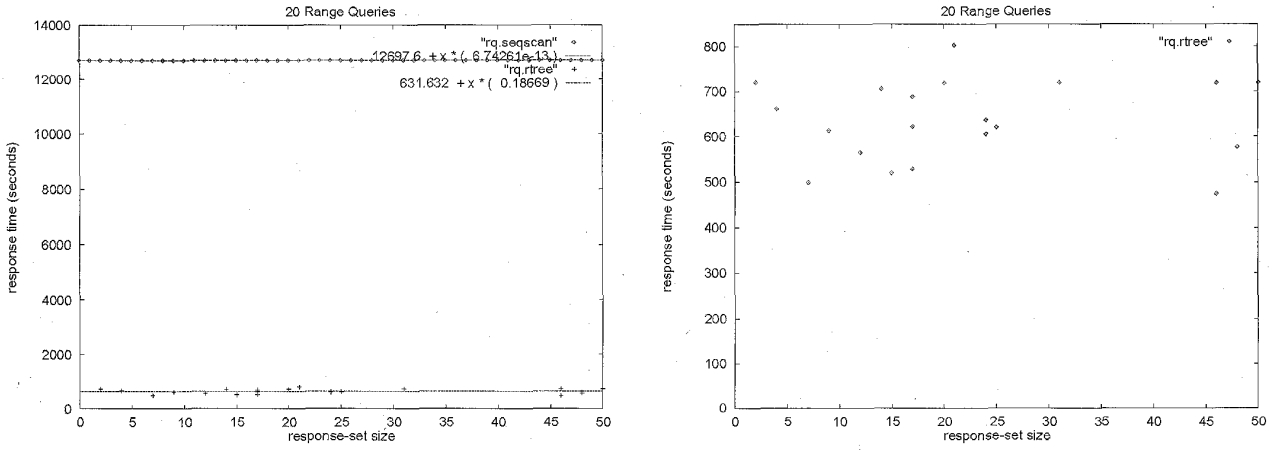


Fig. 10. Response time vs. response-set size a of **GEMINI** for range queries: (a) vs. **scan**, and (b) zoomed in.

TABLE 4
RESPONSE TIMES FOR RANGE QUERIES OVER VARYING RESPONSE-SET SIZES a, SCAN VS. GEMINI

response-set size a	scan (1) time (sec)	GEMINI (2) time(sec)	ratio (1) : (2)
2	12697.6	720.25	17.63
4	12697.6	660.83	19.21
7	12697.6	499.15	25.44
9	12697.6	613.03	20.71
12	12697.6	564.96	22.48
14	12697.6	705.85	17.99
15	12697.6	520.10	24.41
17	12697.6	528.76	24.01
20	12697.6	718.83	17.66
21	12697.6	803.25	15.81
24	12697.6	605.33	20.98
25	12697.6	620.13	20.48
31	12697.6	720.00	17.64
46	12697.6	474.45	26.76
48	12697.6	576.88	22.01
50	12697.6	719.95	17.64

2) The use of a SAM to speed up nearest-neighbor search time in feature space.

We investigate the former. We examine the discriminatory power of the proposed features by visualizing their clustering properties in feature space. For this, we use *Principal Component Analysis* (PCA) [35] to project the size distribution feature vectors into the plane. PCA finds the optimal projection (in a least squares sense) by identifying the k axes (orthogonal directions) of maximum variance among M-dimensional vectors ($k \leq M$), after centering the vectors about the origin. The M-dimensional vectors are then projected along these principal axes, resulting in k-dimensional vectors.

Fig. 11 shows a plot of the 11-d size distribution feature vectors from the data set described in Section 5 projected into the plane

a) with some overlapping points removed to give a general sense of the four separate classes of tumors, and
b) with only points from the barrier and cone shape classes remaining.

Note how well the features discriminate between the four classes of tumor shapes:

1) barrier,
2) biased,
3) cone, and
4) uniform.

Most of the members from each class are confined to mutually distinct clusters. This illustrates the high discriminatory power of the size distribution. Indeed, mathematical morphology seems to be an appropriate tool for this domain.

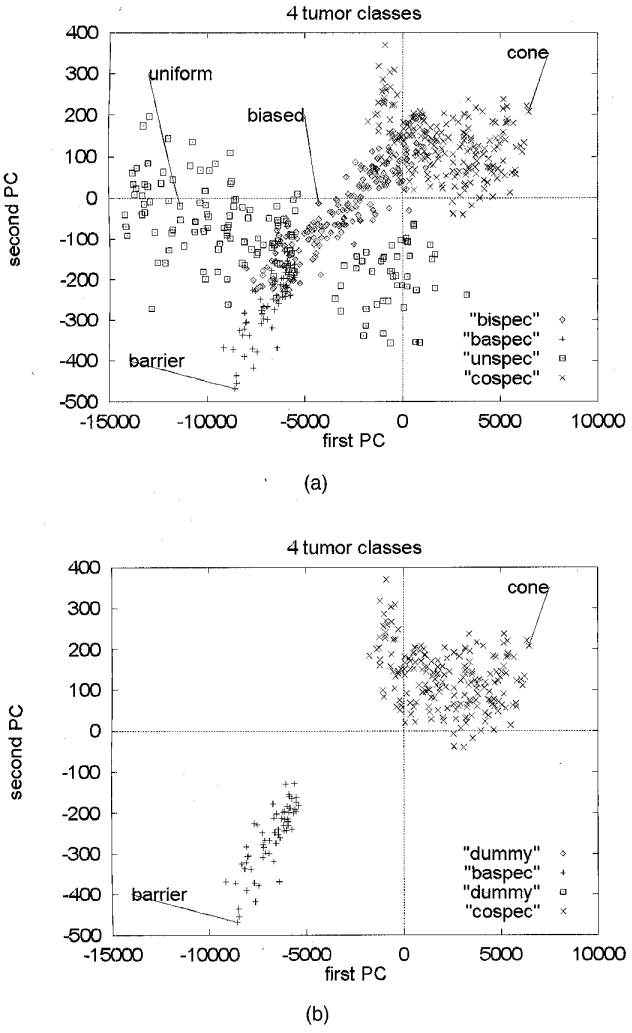


Fig. 11. A 2-d plot of the 11-d features reduced to two PCs by the K-L transform.

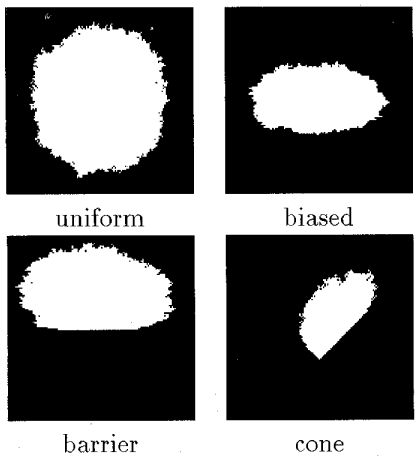


Fig. 12. Blow-ups of the four highlighted points from Fig. 11.

7 Conclusions

We have focused on fast searching for similar shapes with an emphasis on tumor-like shapes. To solve the problem, we used a *multiscale* distance function, the so-called 'morphological' distance. This distance function is based on concepts from signal processing, specifically mathematical morphology. The distance is invariant to translations and rotations, and gives attention to all levels ('scales') of detail. Precision/recall experiments from Section 5 cross-validate that the proposed distance function correlates well with visual perception, giving 80 percent precision at 100 percent recall.

From the database end, we used the GEMINI approach [15], [1], [16], which is the state-of-the-art in multimedia indexing. The main contribution of this work is that it manages to integrate the morphological distance into the GEMINI framework. This is done by using the coefficients of the size distribution as features, and by showing that any L_p distance in the resulting feature space lower-bounds the morphological distance. Given the Lower-Bounding Lemma (Lemma 1), this guarantees no false dismissals for range queries. We then show how to use this machinery in an efficient, general nearest-neighbor algorithm, and prove the correctness of the algorithm. An additional contribution is the implementation of the proposed method and the experimentation on a synthetic, but realistic, database of tumor-like shapes. The proposed method was shown to achieve dramatic speed-ups (up to 27-fold) over a straightforward sequential scan.

Appendix A
Introduction to Mathematical Morphology

Mathematical Morphology is a rich quantitative theory of shape which incorporates a multiscale component. Since the 1980s, morphology and its applications have become extremely popular. Fig. 13 gives picture definitions for some important morphological operators. The formal definitions of the operators and the intuition behind them is presented below. For a more detailed yet accessible introduction, the reader is referred to [21].

In mathematical morphology, mappings are defined in terms of a *structuring element*, a small, primitive shape, which interacts with the input image to transform it and, in the process, extract useful information about its geometrical and topological structure. The basic morphological operators are dilation, erosion, opening, and closing.

Consider black-and-white images in 2-d space; the 'white' points of an image are a subset of the 2-d address space, while the background is, by convention, black. More formally, let $\mathcal{X}$ (the "shape space") be a set of compact subsets of $\Re^2$, and $\mathcal{R}$ be the group of rigid motions $R: \mathcal{X} \mapsto \mathcal{X}$. Next, we present the definitions and concepts that we need for our application. Table 5 lists the symbols and their definitions.

Let X_h denote the translation of shape X by the vector h, and let H^s denote the reflection of shape H with respect to the origin:

$$H^s = \{-h \mid h \in H\} \tag{22}$$

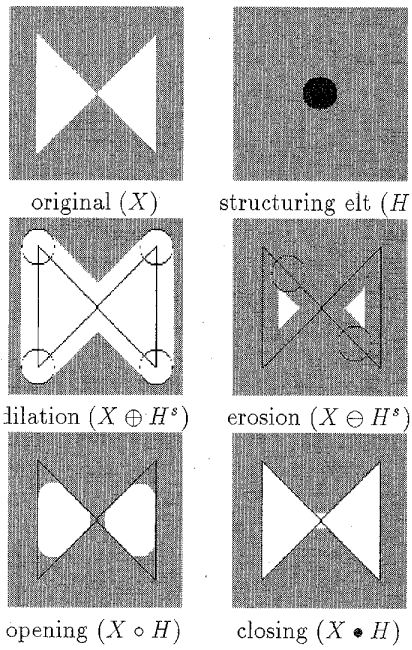


Fig. 13. Original image (top left), structuring element (top right), opening, and closing.

TABLE 5
SYMBOLS FROM MATHEMATICAL MORPHOLOGY

symbol	*definition*
$\Re$	the set of reals
$\Re_+$	the set of non-negative reals
$\mathcal{Z}_+$	the set of non-negative integers
$\oplus$	the operator for dilation
$\ominus$	the operator for erosion
$\circ$	the operator for opening
$\bullet$	the operator for closing

DEFINITION 6. *The **dilation**, $X \oplus H^s$, of a shape $X \subset \Re^2$ by a structuring element H, is defined as*

$$X \oplus H^s = \bigcup_{h \in H} X_{-h} = \{(x, y) \in \Re^2 \mid H_{(x,y)} \cap X \neq \emptyset\} \quad (23)$$

Fig. 13 shows a 'butterfly' shape X, dilated by the unit circle H. Intuitively, a dilation 'blows-up' the original shape X by tracing its perimeter (and all the internal points) with a 'brush' of foot H.

We also use mH, $m \in \mathcal{Z}_+$ to denote $H \oplus H \oplus \cdots \oplus H$ ($m - 1$ times), i.e., a structuring element of size m. Intuitively, if the structuring element H is the unit ball, then the shape mH is a ball of radius m.

DEFINITION 7. *The **erosion**, $X \ominus H^s$, of a shape $X \subset \Re^2$ by a structuring element H, is defined as*

$$X \ominus H^s = \bigcup_{h \in H} X_{-h} = \{(x, y) \in \Re^2 \mid H_{(x,y)} \subseteq X\} \quad (24)$$

Fig. 13 shows the butterfly shape X eroded by the unit circle H, which is a dilation of the complement of X. Intuitively, an erosion deletes part of the original shape X by tracing its perimeter with an 'eraser' of foot H.

Erosion and dilation are *dual* operators, in the sense that $X \ominus H^s = (X^c \oplus H^s)^c$, where ^{c} stands for complementation with respect to $\mathcal{R}^2$.

Two important composite morphological operators are opening and closing:

DEFINITION 8. *The **opening**, $X \circ H$, of a shape $X \subset \Re^2$ by a structuring element H, is defined as an erosion followed by a dilation:*

$$X \circ H = (X \ominus H^s) \oplus H \quad (25)$$

Fig. 13 shows the opening of shape X. Intuitively, the opening is the set of points that a brush of foot H can reach, when the brush is confined inside the shape, and is barely allowed to touch the periphery of the shape.

DEFINITION 9. *The **closing**, $X \bullet H$, of a shape $X \subset \Re^2$ by a structuring element H, is defined as a dilation followed by an erosion:*

$$X \bullet H = (X \oplus H^s) \ominus H \quad (26)$$

Fig. 13 shows the closing of shape X, which is the opening of the complement of X. Intuitively, the closing is the set of points that remain after an eraser of foot H sweeps the outside of the dilated X.

Thus, the opening by a circle of radius n in effect 'cuts the corners'; that is, it eliminates the protruding details of the shape X, with radius less than n. Symmetrically, the closing 'fills the concavities' of the appropriate scale.

APPENDIX B
TUMOR GROWTH MODEL

We use a discrete-time version of Eden's tumor growth model [12], which is illustrated in Fig. 14 and Fig. 15. At time $t = 0$, only one grid-cell is 'infected'; each infected grid-cell may infect its four nondiagonal neighbors with equal probability p at each time-tick.

On the basic model. we have added the notion of East-West/North-South bias to capture the effects of anisotropic growth patterns, due to anisotropies in the surrounding

		6	9	9				
8	5	4	7	4	6	9		
	6	3	2	1	6	8		
	9			9	9			

Fig. 14. Lattice at $t = 9$. The infection time of each infected cell is marked.

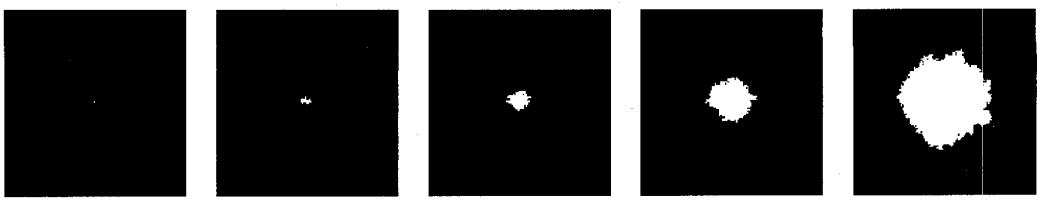


Fig. 15. Initial seed (left column) and snapshots of tumor at later time steps, with probability of infection $p_{EW} = P_{NS} = 0.7$.

tissue (e.g., lesions shaped by their location within the lung, breast, or liver.) Thus, in our model, an infected grid-cell has probability p_{NS} to infect its North and South neighbors, and probability p_{EW} to infect its East/West ones, with p_{NS} not necessarily equal to p_{EW}.

ACKNOWLEDGMENTS

The research of Philip Korn, Nicholas Sidiropoulos, and Christos Faloutsos was performed while they were with the University of Maryland, College Park.

REFERENCES

[1] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient Similarity Search in Sequence Databases," *Proc. FODO, Fourth Int'l Conf. Foundations of Data Organization and Algorithms*, pp. 69–84, Evanston, Ill., Oct. 1993; also available through anonymous ftp, at olympos.cs.umd.edu: ftp/pub/TechReports/fodo.ps

[2] V. Anastassopoulos and A.N. Venetsanopoulos, "Classification Properties of the Spectrum and Its Use for Pattern Identification," *Circuits, Systems, and Signal Processing*, vol. 10, no. 3, 1991.

[3] J.R. Bach, S. Paul, and R. Jain, "A Visual Information Management System for the Interactive Retrieval of Faces," *IEEE Trans. Knowledge and Data Eng.*, vol. 5, no. 4, pp. 619–628, Aug. 1993.

[4] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The r*-Tree: An Efficient and Robust Access Method for Points and Rectangles," *Proc. ACM SIGMOD*, pp. 322–331, May 1990.

[5] J.L. Bentley, "Multidimensional Binary Search Trees Used for Associative Searching," *Comm. ACM*, vol. 18, no. 9, pp. 509–517, Sept. 1975.

[6] S. Berchtold, D. Keim, and H.-P. Kriegel, "Using Extended Feature Objects for Partial Similarity Retrieval," *VLDB J.*, vol. 6, no. 4, 1997.

[7] T. Bozkaya and M. Ozsoyoglu, "Distance-Based Indexing for High-Dimensional Metric Spaces," *Proc. ACM SIGMOD '97*, pp. 357–368, Tucson, Ariz., May 1997.

[8] S. Brin, "Near Neighbor Search in Large Metric Spaces," *Proc. 21st Int'l Conf. VLDB*, pp. 574–584, Zurich, Sept. 1995.

[9] L. Gottesfeld Brown, "A Survey of Image Registration Techniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 325–376, Dec. 1992.

[10] C.J. Burdett, H.G. Longbotham, M. Desai, Walter B. Richardson, and John F. Stoll, "Nonlinear Indicators of Malignancy," *Proc. SPIE—Biomedical Image Processing and Biomedical Visualization*, vol. 1905, part 2, pp. 853–860, Feb. 1993.

[11] P. Ciaccia, M. Patella, and P. Zezula, "M-Tree: An Efficient Access Method for Similarity Search in Metric Spaces," *Proc. 23rd Int'l Conf. VLDB*, pp. 426–435, Athens, Greece, Aug. 1997.

[12] M. Eden, "A Two-Dimensional Growth Process," *Proc. Fourth Berkeley Symp. Math. Statistics and Probability*, J. Neyman ed., Univ. of California Press, Berkeley, Calif., 1961

[13] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and Effective Querying by Image Content," *J. Intelligent Information Systems*, vol. 3, nos. 3/4, pp. 231–262, July 1994.

[14] C. Faloutsos and S. Roseman, "Fractals for Secondary Key Retrieval," *Proc. Eighth ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Systems PODS*, pp. 247–252, Mar. 1989. also available as UMIACS-TR-89-47 and CS-TR-2242.

[15] C. Faloutsos, *Searching Multimedia Databases by Content*, Kluwer Academic, 1996.

[16] C. Faloutsos, M. Ranganathan, and Yannis Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," *Proc. ACM SIGMOD*, pp. 419–429, May 1994; also available as CS-TR-3190, UMIACS-TR-93-131, ISR TR-93-86.

[17] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by Image and Video Content: The Qbic System," *Computer*, vol. 28, no. 9, pp. 23–32, Sept. 1995.

[18] K. Fukunaga and P.M. Narendra, "A Branch and Bound Algorithm for Computing k-Nearest Neighbors," *IEEE Trans. Computers*, vol. 24, no. 7, pp. 750–753, July 1975.

[19] I. Gargantini, "An Effective Way to Represent Quadtrees," *Comm. ACM*, vol. 25, no. 12, pp. 905–910, Dec. 1982.

[20] Gary and Mehrotra, "Shape Similarity-Based Retrieval in Image Database Systems," *Proc. SPIE*, vol. 1662, pp. 2–8, 1992.

[21] C. Giardina and E. Dougherty, *Morphological Methods in Image and Signal Processing*. Englewood Cliffs, N.J.: Prentice Hall, 1988.

[22] D.Q. Goldin and P.C. Kanellakis, "On Similarity Queries for Time-Series Data: Constraint Specification and Implementation," *Proc. Int'l Conf. Principles and Practice of Constraint Programming*, Sept. 1995.

[23] V. Gudivada, "Indexing for Efficient Spatial-Similarity Query Processing in Multimedia Databases," *Proc. SPIE Proc.*, vol. 2916, pp. 46–52, Boston, Nov. 1996.

[24] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," *Proc. ACM SIGMOD*, pp. 47–57, June 1984.

[25] D. Harman, *Proc. TREC 3, Third Text Retrieval Conf.*, special publication, National Inst. of Standards and Technology, Gaithersburg, Md., 1995.

[26] K. Hinrichs and J. Nievergelt, "The Grid File: A Data Structure to Support Proximity Queries on Spatial Objects," *Proc. WG Int'l Workshop Graph Theoretic Concepts in Computer Science*, pp. 100–113, 1983.

[27] B. Horn, *Robot Vision*, MIT Electrical Engineering and Computer Science Series. Cambridge, Mass.: MIT Press, 1986.

[28] B. Horn, *Robot Vision*, Cambridge, Mass.: MIT Press, 1986.

[29] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE Trans. PAMI*, vol. 15, no. 9, pp. 850–863, Sept. 1993.

[30] C. Jacobs, A. Finkelstein, and D. Salesin, "Fast Multiresolution Image Querying," *Proc. SIGGRAPH*, pp. 277–286, Los Angeles, Aug. 1995.

[31] H.V. Jagadish, "Linear Clustering of Objects with Multiple Attributes," *Proc. ACM SIGMOD Conf.*, pp. 332–342, May 1990.

[32] H.V. Jagadish, "Spatial Search with Polyhedra," *Proc. Sixth IEEE Int'l Conf. Data Eng.*, Feb. 1990.

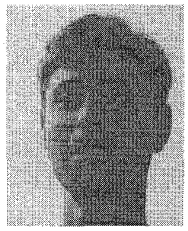
[33] H.V. Jagadish, "A Retrieval Technique for Similar Shapes," *Proc. ACM SIGMOD Conf.*, pp. 208–217, May 1991.

[34] T. Ji, M. Sundareshan, and H. Roehrig, "Adaptive Image Contrast Enhancement Based on Human Visual Properties," *IEEE Trans. Medical Imaging*, vol. 13, no. 4, Dec. 1994.

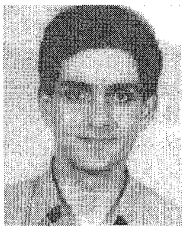
[35] I.T. Jolliffe, *Principal Component Analysis*. Springer Verlag, 1986.

[36] V. Kobla, D. Doermann, K.-I. Lin, and C. Faloutsos, "Compressed Domain Video Indexing Techniques Using DCT and Motion Vector Information in MPEG Video," *Proc. SPIE*, vol. 2916, Boston, Nov. 1996.

[37] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Zenon Protopapas, "Fast Nearest Neighbor Search in Medical Image Databases," *Proc. 22nd VLDB Conf.*, pp. 215–226, Bombay, India, Sept. 1996.
[38] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas, "Fast and Effective Similarity Search in Medical Tumor Databases Using Morphology," *Proc. SPIE*, vol. 2916, pp. 116–129, Boston, Nov. 1996.
[39] D.B. Lomet and B. Salzberg, "The hb-Tree: A Multiattribute Indexing Method with Good Guaranteed Performance," *Proc. ACM TODS*, vol. 15, no. 4, pp. 625–658, Dec. 1990.
[40] P. Maragos, "Morphological Skeleton Representation and Coding of Binary Images," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 34, pp. 1,228–1,244, 1986.
[41] P. Maragos, "Morphology-Based Symbolic Image Modeling, Multi-Scale Nonlinear Smoothing, and Pattern Spectrum," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, Ann Arbor, Mich., pp. 766–773, June 1988.
[42] P. Maragos, "Pattern Spectrum and Multiscale Shape Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 701–716, July 1989.
[43] P. Maragos and R.W. Schafer, "Morphological Skeleton Representation and Coding of Binary Images," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 34, pp. 1,228–1,244, 1986.
[44] G. Matheron, *Random Sets and Integral Geometry*, Wiley. New York, 1975.
[45] R. Mehrotra and J. Gary, "Feature-Based Retrieval of Similar Shapes," *Proc. Ninth Int'l Conf. Data Eng.*, pp. 108–115, Vienna, Apr. 1993.
[46] 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," *Proc. SPIE 1993 Int'l Symp. Electronic Imaging: Science and Technology*, Conf. 1908, Storage and Retrieval for Image and Video Databases, Feb. 1993; also available as IBM Research Report No. RJ 9203 81511, Feb. 1993.
[47] J. Orenstein, "Spatial Query Processing in an Object-Oriented Database System," *Proc. ACM SIGMOD*, pp. 326–336, May 1986.
[48] T. Pavlidis, "Algorithms for Shape Analysis of Contours and Waveforms," *IEEE T. Pattern Analysis and Machine Intelligence*, vol. 2, pp. 301–312, 1980.
[49] S. Pong and A.N. Venetsanopoulos, "Rotationally Invariant Spectrum: An Object Recognition Descriptor Based on Mathematical Morphology," *Circuits, Systems, and Signal Processing*, vol. 11, no. 4, pp. 455–492, 1992.
[50] D. Rafiei and A. Mendelzon, "Similarity-Based Queries for Time Series Data," *Proc. ACM SIGMOD*, pp. 13–25, Tucson, Ariz., May 1997.
[51] J.T. Robinson, "The k-d-b-Tree: A Search Structure for Large Multidimensional Dynamic Indexes," *Proc. ACM SIGMOD*, pp. 10–18, 1981.
[52] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest Neighbor Queries," *Proc. ACM SIGMOD*, pp. 71–79, May 1995.
[53] S. Santini and R. Jain, "Similarity Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1996.
[54] T. Seidl and H.-P. Kriegel, "Efficient User-Adaptable Similarity Search in Large Multimedia Databases," *Proc. 23rd VLDB Conf.*, pp. 506–515, Athens, Aug. 1997.
[55] T. Seidl and H.-P. Kriegel, "Optimal Multi-Step *k*-Nearest Neighbor Search," *Proc. ACM SIGMOD*, pp. 154–165, Seattle, June 1998.
[56] T. Sellis, N. Roussopoulos, and C. Faloutsos, "The r+ Tree: A Dynamic Index for Multi-Dimensional Objects," *Proc. 13th Int'l Conf. VLDB*, pp. 507–518, England, Sept. 1987; also available as Technical Report Nos. SRC-TR-87-32, UMIACS-TR-87-3, CS-TR-1795.
[57] V.S. Subrahmanian, *Principles of Multimedia Database Systems*. San Francisco: Morgan Kauffman, 1998.
[58] Z. Zhou and A.N. Venetsanopoulos, "Morphological Skeleton Representation and Shape Recognition," *Proc. IEEE Second Int'l Conf. ASSP*, New York, pp. 948–951, 1988.



Philip (Flip) Korn received the ScB degree in applied mathematics and computer science from Brown University in 1993; and the MSc and PhD degrees from the University of Maryland, College Park, in 1995 and 1998, respectively. He is currently a senior member of the technical staff at AT&T Labs in Florham Park, New Jersey. His current research interests include multimedia indexing, data mining, data compression, and decision support.



Nicholas Sidiropoulos (M'92) received the Diploma in electrical engineering from the Aristotelian University of Thessaloniki, Greece, and the MSc and PhD degrees in electrical engineering from the University of Maryland, College Park, in 1988, 1990, and 1992, respectively. From 1988 to 1992, he was a Fulbright fellow and a research assistant at the Institute for Systems Research of the University of Maryland (ISR-UMCP). From September 1992 to June 1994, he served his Greek military service as a lecturer at the Hellenic Air Force Academy. From October 1993 to June 1994, he also was a member of the technical staff, Systems Integration Division, G-Systems Ltd., Athens, Greece. He rejoined the ISR-UMCP as a postdoctoral research associate in August 1994, and was promoted to assistant research scientist in January 1996. He joined the University of Virginia as an assistant professor of electrical engineering in July 1997. His research interests are in the broad area of signal and image processing. He received the National Science Foundation/ Career Award (Signal Processing Systems Program) in June 1998. He is a member of the IEEE Computer Society.

Christos Faloutsos received the BSc degree in electrical engineering in 1981 from the National Technical University of Athens, Greece; and the MSc and PhD degrees in computer science from the University of Toronto, Canada, in 1982 and 1987, respectively. From 1985-1997, he was with the Department of Computer Science at the University of Maryland, College Park. He is currently an associate professor in the Computer Science Department at Carnegie Mellon University in Pittsburgh, Pennsylvania. In 1989, he received the National Science Foundation Presidential Young Investigator Award. His research interests include physical database design, searching methods for text, geographic information systems, indexing methods for medical and multimedia databases, and data mining.