A Generalized Metric Distance between Hierarchically Partitioned Images

Maude Manouvrier, Marta Rukoz, Geneviève Jomier

A generalized metric distance between hierarchically partitioned images*

Maude Manouvrier§
Paris Dauphine University
LAMSADE, Place du Maréchal de Lattre de Tassigny 75775 Paris Cedex 16 France
manouvrier@lamsade.dauphine.fr

Marta Rukoz
Universidad Central de Venezuela - CCPD, Escuela de Computación, Ciudad Universitaria, Av. Los Ilustres, 1041 Caracas, Venezuela
mrukoz@strix.ciens.ucv.ve

Geneviève Jomier
Paris Dauphine University
LAMSADE, Place du Maréchal de Lattre de Tassigny 75775 Paris Cedex 16 France
jomier@lamsade.dauphine.fr

ABSTRACT

This article presents a generalized metric distance, called Δ-distance, between images represented by a tree structure resulting from a recursive image partition. This distance is used to perform content-based image retrieval queries in databases. Δ-distance allows to retrieve images globally similar to a query image. This distance takes into account the location of the image visual features. It can be performed using a multi-level filtering algorithm. Moreover, Δ-distance allows region-based queries. In this case, the resulting images contain quadrants similar to the quadrants selected by the user in the query image or contain quadrants similar to the entire query image. Because it is a generalized distance function, some particular cases of the Δ-distance appear in existing content-based image retrieval systems.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database Applications—*Image databases*; I.4.10 [**Image Processing and Computer Vision**]: Image Representation—*Hierarchical*; I.5.3 [**Pattern Recognition**]: Clustering—*Similarity measures*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Measurement, Standardization

Keywords

Image database, quad/quin/nona-tree, recursive image decomposition, content-based image retrieval, multi-level filtering, image region similarity

*This work was supported by the CDCH in Venezuela.

§Corresponding author

MDM/KDD'05, August 21, 2005, Chicago, IL, USA.

1. INTRODUCTION

Content-based image retrieval (CBIR) has become a very well studied research area because of the increased number of available image databases. Readers are referred to surveys [21] and [25]. In a CBIR system, the user chooses a query image in the database and the system returns a list of images similar to the query image. The similarity between images is measured using a distance function [19]. The smaller the distance between images is, the more similar the images are.

To improve image retrieval accuracy or to enlarge the query expression, several approaches [1, 2, 5, 7, 9, 10, 11, 12, 13, 15, 20] use a tree to model a recursive image partition and compute a distance between images based on the comparison of the image tree representation. All the distances used in these approaches are analogous and can be generalized into a single generic definition. However, to the best of our knowledge, no general definition of such a distance exists.

The first contribution of this article is to formally define the generalization of the distances used in the aforementioned approaches. We called this generalized distance: Δ-distance. By analyzing Δ-distance, we point up that several distances (global or local) and several distance computation processes (with or without filtering) could be derived from the same definition depending on the parameters used. The last contribution of this article is a short survey showing the distance-based relations between the aforementioned approaches.

The article is organized in the following way. Section 2 deals with the recursive image decomposition for content-based image retrieval and presents the notation used in this article. Section 3 presents the contributions of this article, the Δ-distance and the Δ-based image retrieval processes. Finally, Section 4 concludes this article and offers directions for future work.

2. BACKGROUND AND RELATED WORK

Several content-based image retrieval approaches [1, 2, 5, 7, 9, 10, 11, 12, 13, 15, 20] deal with the same image search process. Firstly, the images from the database are decomposed into several fixed-size quadrants. Secondly, each image is represented by a tree storing the feature vectors of all its

quadrants. Finally, image similarity is computed using a distance between image trees.

This section explains in details the two first steps of the aforementioned image retrieval process – the image decomposition (see Section 2.1) and the tree representation (see Section 2.2) – and presents the notation conventions (see Section 2.3). It provides the necessary background to understand how a generalized Δ-distance, presented in Section 3, could be defined.

2.1 Recursive image partition

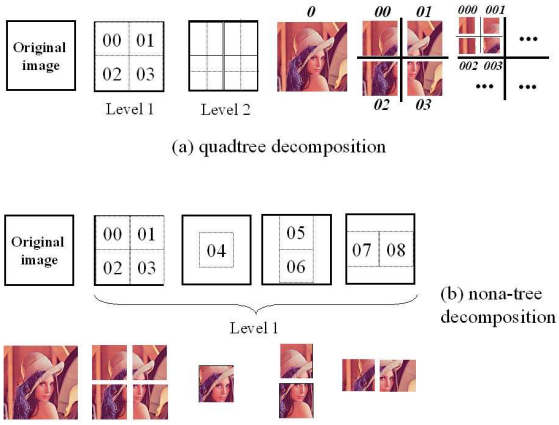

Figure 1: An image recursively decomposed into four or nine quadrants.

An image could be recursively decomposed into fixed-size quadrants. Using a quadtree decomposition, an image is recursively decomposed into four quadrants. Figure 1.(a) shows an example. The root of the quadtree (stored at level 0 in this article) represents the entire image. At level 1, the root node has four child nodes representing the first four quadrants of the image. If the decomposition is stopped after two iterations, level 2 contains 16 nodes, representing the four sub-quadrants of the first level quadrants. This kind of decomposition is used in [1, 2, 5, 7, 9, 10, 12, 11, 15].

Using a quin-tree, an image is recursively decomposed into five quadrants, the first fourth ones following the NW, NE, SW and SE directions and the fifth one representing the center of the image. This kind of decomposition is used in [18]. Using a nona-tree decomposition, an image is recursively decomposed into nine quadrants. As previously, the root of the tree represents the entire image. But at level 1, the root node has nine child nodes representing the first nine quadrants of the image. Figure 1.(b) shows the nona-tree decomposition of the *Lena* image. If the decomposition is stopped after two iterations, level 2 contains 81 nodes, each node of the first level having 9 child nodes. In [20, 27], a nona-tree is used for content-based image retrieval.

Several types of decomposition can be mixed. For example, level 1 of the hierarchical structure proposed in [13, 14] contains nine nodes, as in a nona-tree. The second level, however, corresponds to a second level of a quadtree containing only 16 nodes.

Such a recursive image decomposition could be used for content-based image retrieval. In this case, in order to compare image visual features, all the images of the database are decomposed in the same way and all the resulting trees have the same number of nodes by level and the same number of levels. The images are represented by full balanced trees. The following section explains more deeply how trees could be used in CBIR systems.

2.2 Tree structures for CBIR

Content-based image retrieval systems allow to retrieve images from a database according to their visual features. The features, including color [24], texture [19] and shape [4], are stored as numerical vectors, called *feature vectors* [9]. When tree structures are used in CBIR system, each tree node stores the feature vector of the corresponding image quadrant. For example in [12], the quadtree nodes store the color histograms of the corresponding quadrant. Figure 2.(b) gives an example of such a quadtree for the *Lena* image. In [11], each quadtree root stores the average color vectors of each corresponding image, in a three-dimensional color space $(R_{avg}, G_{avg}, B_{avg})$, R for red, G for green and B for blue. The other nodes store a color histogram of the corresponding image quadrant. Figure 2.(c) gives an example of such a quadtree storing an average color vector in all its nodes. In [9], the tree nodes store color vectors having 9 dimensions (E_u,E_v,E_w, σ_u,σ_v,σ_w, s_u,s_v,s_w), based on color moments where E represents the average color, σ represents the variance and s represents the skewness of each dimension in the color space (R, G, B). Any kind of feature vector could be stored into the tree nodes, like a shape feature vector [10], a combination of color and texture captured via histograms [15], or a dominant color [5]. We call such a structure a *Multi-Level Feature Vector*. This structure has several purposes in content-based image retrieval.

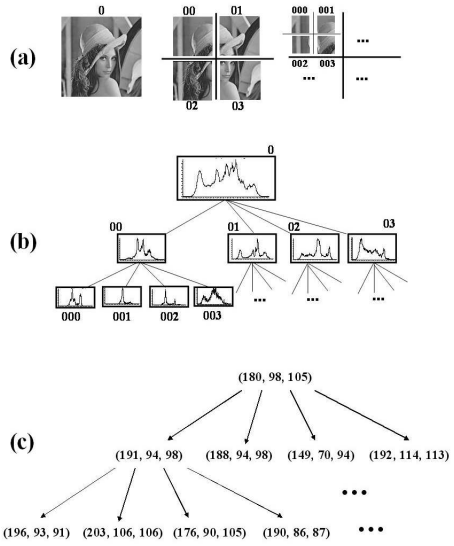

Figure 2: The *Lena* image: (a) decomposed into a quadtree and represented by (b) a multi-level color histogram and (c) a multi-level average color feature vector.

Firstly, a multi-level structure can be used as a *multi-level filtering* structure as it is suitable to a *coarse-to-fine* representation [9]. In this case, each image is first compared with the query image according to its global feature vectors (stored in the root node of the tree). If they are similar enough (with regard to a similarity metric and a given threshold), the first sub-quadrants of the images are compared, and so on. For more details about multi-level filtering using a partition-based tree, readers are referred to [9, 12]. Section 3.2 shows how the generalized distance, proposed in this article, is used for multi-level filtering.

Secondly, a multi-level feature vector allows different kinds of query image called, in [9], *global search*, *sub-image search* or *pattern search*. Global search consists in retrieving images of the database globally similar to a query image. Sub-image search compares a query image with any image quadrant stored in the database whatever is the size of the compared quadrants. Pattern search retrieves images from the database having similar regions chosen in a query image. Among the approaches based on images represented by multi-level feature vectors, the proposition presented in [11, 12] allows global search. The approach of [13, 14] deals with sub-image search. In [15], a solution is presented for pattern search.

2.3 Notation

In this article, the tree nodes and the corresponding image quadrants are identified in the following way. The root node, identified by numeral 0 in this article, represents the initial quadrant containing the whole image. Numerals 0, 1, 2 or 3, following their parent node identifier 0, identify the four first level image quadrants and the four first level tree nodes. This identification is done using a Z order, which corresponds to the NW, NE, SW and SE directions. Numeral 4, following its parent node identifier 0, identifies the central quadrant and the corresponding tree node. Numeral 5 and 6 (resp. 7 and 8), following their parent node identifier 0, identify the vertical (resp. horizontal) central quadrants and the corresponding tree nodes. Recursively, sub-quadrants of an image quadrant n and children of a tree node n are identified by nx where $x \in \{0, 1, 2, 3, 5, 6, 7, 8\}$. This identification convention is used in Figure 1.

In the following, letter ℓ represents a level in a tree. The root node is, in this article, at level 0. Letter n represents a tree node or an image quadrant identifier. Letter $\mathcal{N}$ represents the set of node (resp. quadrant) identifiers appearing in the database. In this article, we consider that all images of the database have the same tree decomposition (same numbers of nodes by level and same number of levels by tree). Letters i, i' and j represent image identifiers and letter q represents the query image identifier. Table 1 (at the end of the paper) summarizes the meaning of all symbols used in this article. Two nodes (or quadrants) with the same identifier in two different trees (or images) are called *homologous nodes* (or homologous quadrants). This convention is used in the definition of the Δ-distance presented in the next section.

3. Δ-DISTANCE

This section presents the Δ-distance. We first proposed this distance in [22, 23] for images represented by quadtrees. In this article, our first definition of Δ-distance is generalized to any tree based on recursive decomposition of images. This generalization represents the first step of our final main purpose: to develop a CBIR system allowing to compare different Δ-based distances on the same image set and helping the user to define the right parameters depending on its image database. At the end, this system could also be used to compare similar approaches [1, 2, 5, 7, 9, 10, 12, 11, 15] by implementing their common characteristics. Indeed, until now, it is difficult to really compare the performances of these approaches because their prototypes are not always accessible and are generally based on different frameworks. This article does not deal with this desired CBIR system but gives the first bricks to develop it.

After defining the Δ-distance (see Subsection 3.1), this section points up that Δ-distance is a distance for content-based image retrieval allowing multi-level filtering (see Subsection 3.2) or region-based image queries (see Subsections 3.3 and 3.4). Existing distances used in several CBIR systems appear to be particular cases of Δ-distance, as shown out in Subsection 3.5.

3.1 General definition of Δ-distance

Let $\delta(i, j, n)$ be a normalized metric distance – $\delta(i, j, n) \in [0, 1]$ – between feature vectors of quadrants n of two different images i and j. δ can be any geometric distance of the Minkowski family (see [19] for more details about this family) and can be a weighted distance. Let Δ be the distance between two images i and j, represented by a multi-level feature vector. Δ-distance is defined as a weighted sum of normalized δ-distances between feature vectors stored in tree nodes n:

$$\Delta(i, j) = \frac{\sum_{n \in \mathcal{N}} w_n \delta(i, j, n)}{\mathcal{W}} \quad (1)$$

The coefficient w_n, $w_n \geq 0$, represents the weight of distance δ between homologous nodes n in the Δ-distance computation. Δ-distance is normalized by the denominator $\mathcal{W}$ which is the sum of all weights w_n associated with the image quadrants n: $\mathcal{W} = \sum_{n \in \mathcal{N}} w_n$. At least one w_n must be different from zero: $\Delta \in [0, 1]$. Below we only consider normalized distances.

Δ is a metric distance because it is a linear combination of metric distances (δ) between homologous tree nodes. Thus, $\forall i$, i' and j, three different images represented by multi-level feature vectors:

- $\Delta(i, i) = 0$ because $\forall n \in \mathcal{N}$, $\delta(i, i, n) = 0$
- $\Delta(i, j) = \Delta(j, i)$ because $\forall n \in \mathcal{N}$, $\delta(i, j, n) = \delta(j, i, n)$
- $\Delta(i, i') \leq \Delta(i, j) + \Delta(j, i')$ because δ is a metric distance then $\forall n \in \mathcal{N}$, $\delta(i, i', n) \leq \delta(i, j, n) + \delta(j, i', n)$ and thus $w_n \delta(i, i', n) \leq w_n \delta(i, j, n) + w_n \delta(j, i', n)$ because $\forall n \in \mathcal{N}$, $w_n \geq 0$

The Δ-distance must be used when the spatial position of features is an important criterion for the image similarity. This distance takes into account the feature vectors values and their positions in the images. Figure 3, issued from the prototype of [9], shows an example of this kind of content-based image retrieval. The query image is on the left - see

Figure 3.(a). Several images of the database result from image blurring, pixelization or spreading applied to the query image. Other images correspond to the query image after a rotation. As shown in Figures 3.(b) and 3.(c), the result image ranking is different when a global distance or a Δ-distance is used. Using a global distance, the images are compared with the query image using only their global feature vectors (9-dimensional vectors based on color moments). In this case, the rotated images appear first in the query result because they result from a rotation of the query image without any other image processing operations. Using Δ-distance, the images are compared with the query image using their multi-level feature vectors and then comparing all homologous image quadrants. In this example, the images of the database are represented by a three-level balanced quadtree containing a 9-dimension color vectors. An image quadrant at level ℓ represents $4^{-\ell}$ of the entire image surface. Thus each image quadrant n is associated with a weight $w_n = 4^{-\ell}$ where ℓ is the level of the corresponding quadtree node n. Using Δ-distance, the processed images appear first in the query result. The rotated images appear farther because of the different location of their features comparing to the query image.

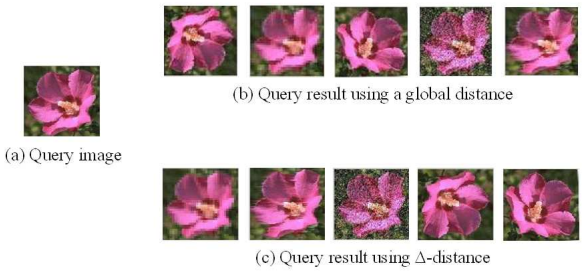

Figure 3: A global search result using a global distance or a Δ-distance.

3.2 Multi-level filtering using Δ-distance

To compute the Δ-distance between two images represented by multi-level feature vectors, an exhaustive comparison of both trees is done. However, the Δ-distance computation can be gradually refined by comparing both trees level by level, following a breadth first order.

Let $\Delta^{(\ell)}(i,j)$ be an approximation of Δ-distance between multi-level feature vectors of images i and j. It is computed from the distances between homologous nodes appearing from root level (level 0) to level ℓ. Intuitively, the $\Delta^{(\ell)}$-distance is computed without taking into account details after a certain tree level ℓ, i.e. the information contained in lower-level tree nodes. Let $\mathcal{N}^{(\ell)}$ be the set of nodes appearing at level ℓ ($\mathcal{N}^{(0)} = 1$). For any given level $\ell \geq 1$ in two multi-level feature vectors of images i and j:

$$\Delta^{(\ell)}(i,j) = \frac{1}{\mathcal{W}} \sum_{k=0}^{\ell} \left(\sum_{n \in \mathcal{N}^{(k)}} w_n \delta(i,j,n) \right) \qquad (2)$$

$\Delta^{(0)}(i,j)$ is computed taking into account only both tree roots, i.e. the global feature vectors of images i and j. If all multi-level feature vectors of the database have, for example, three levels, then for all images i and j: $\Delta^{(3)}(i,j) = \Delta(i,j)$. $\Delta^{(\ell)}$-distance is an increasing function of ℓ:

$$\Delta^{(\ell+1)}(i,j) = \Delta^{(\ell)}(i,j) + \frac{\sum_{n \in \mathcal{N}^{(\ell+1)}} w_n \delta(i,j,n)}{\mathcal{W}} \qquad (3)$$

For any given level $\ell \geq 1$ in multi-level feature vectors:

$$\Delta^{(\ell-1)}(i,j) \leq \Delta^{(\ell)}(i,j) \leq \Delta^{(\ell+1)}(i,j) \qquad (4)$$

Using $\Delta^{(\ell)}$-distance, the global search of images similar to a query image q is computed in several steps of filtering. Let $\mathcal{R}^{(\ell)}$ be the set containing images i from the database such that $\Delta^{(\ell)}(q,i) \leq \alpha$: $\mathcal{R}^{(\ell)} \subset \mathcal{R}^{(\ell+1)}$, the result sets are embedded.

To compute a global search, the user has to specify the query image q, the distance δ (for example the Euclidean distance L_2), the weights w_n and a threshold α. The query result is an ordered set of images i checking $\Delta(i,q) \leq \alpha$ – see Formula (1). The first step of the filtering consists in computing $\Delta^{(0)}$. At the end of this first step, the result is a set $\mathcal{R}^{(0)}$ containing images i from the database such that $\Delta^{(0)}(q,i) \leq \alpha$. Then, for each images in set $\mathcal{R}^{(0)}$, $\Delta^{(1)}$-distance is computed. The result is a set $\mathcal{R}^{(1)}$ containing images i from $\mathcal{R}^{(0)}$ such that $\Delta^{(1)}(q,i) \leq \alpha$. The process stops when the last level of trees is reached, i.e. when Δ-distance is computed. Thus, the exhaustive comparison between all quadrant feature vectors is only computed for a restricted number of images, each step of the filtering process reducing the image set to be compared. Figure 4 summaries this filtering process. This figure deals with a quadtree-based multi-level filtering, but the filtering process can obviously be extended to any image recursive decomposition.

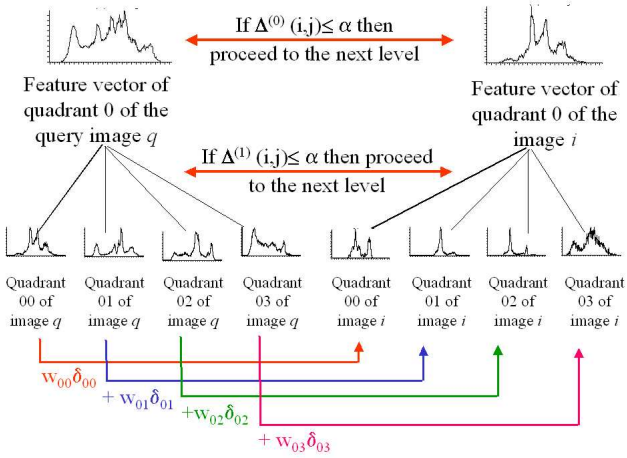

Figure 4: Using the Δ-distance for quadtree-based multi-level filtering.

Formula (3) should be used to approximate Δ-distance when the feature vector of a quadrant n is not an approximation of the feature vectors of its sub-quadrants. For example in

[5], tree nodes store the dominant color of the corresponding quadrant. However, feature vectors are often computed from statistical methods (e.g. color moment). Tree node n can also aggregate its rooted sub-tree's values by containing an approximation of feature vectors stored in its descendant nodes. In these cases, Δ-distance can be approximated by associating $w_n \geq 0$ with all nodes n appearing at level ℓ and associating null weights ($w_n = 0$) with the other nodes. Let $\widetilde{\Delta^{(\ell)}}$-distance be this approximated Δ-distance computed by taking only into account nodes of level ℓ. $\widetilde{\Delta^{(\ell)}}$-distance is defined by the following formula :

$$\widetilde{\Delta^{(\ell)}}(i,j) = \frac{\sum_{n \in \mathcal{N}^{(\ell)}} w_n \delta(i,j,n)}{\sum_{n \in \mathcal{N}^{(\ell)}} w_n} \quad (5)$$

In this formula, nodes n appear at level ℓ in both trees of images i and j ($n \in \mathcal{N}^{(\ell)}$). Formula (5) preserves the previous inequality $\widetilde{\Delta^{(\ell-1)}}(i,j) \leq \widetilde{\Delta^{(\ell)}}(i,j) \leq \widetilde{\Delta^{(\ell+1)}}(i,j)$, because feature vectors stored at level ℓ approximate those stored at level $(\ell+1)$.

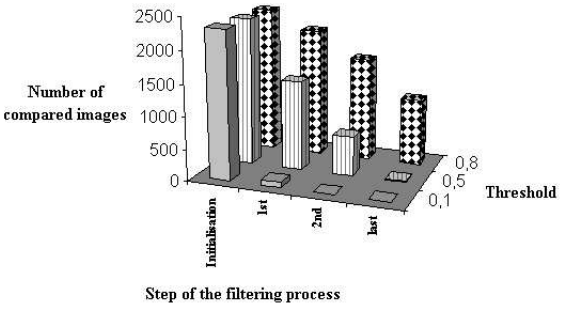

Figure 5: Number of images extracted at each step of the multi-level filtering process.

Figure 5 shows the experiment results obtained with the prototype of [9]. The database contains 2311 images. All images are represented by a three-level feature vectors whose nodes contain 9-dimensional color vectors based on the first third color moments. $\widetilde{\Delta^{(\ell)}}$-distance is used to approximate Δ-distance. The number of compared images decreases at each step of the filtering process. The larger the threshold α, the less the decrease of the number of images to be compared. With $\alpha = 0,5$, the first step eliminates 914 images from the result set: only 1397 images verify $\widetilde{\Delta^{(0)}}(q,i) \leq \alpha$. After the second filtering step (computing $\widetilde{\Delta^{(1)}}(q,i) \leq \alpha$), set $\mathcal{R}^{(1)}$ contains 611 images: 414 images do not verify $\widetilde{\Delta^{(1)}}(q,i) \leq \alpha$.

3.3 Pattern search using Δ

A multi-level feature vector allows pattern search. It consists in retrieving images having regions similar to a query pattern given by the user [9]. The user selects one or several quadrants in a grid image, for example a $4 * 4$ or $16 * 16$ grid depending on the decomposition of the images. The selected quadrants represent a pattern p which becomes the query pattern q_p. Δ-distance can be used to compute such a pattern search associating $w_{n_s} \geq 0$ with all nodes n_s corresponding to the image quadrants selected by the user and $w_n = 0$ for the other quadrants. We call such a distance: $\Delta_p(i,j)$.

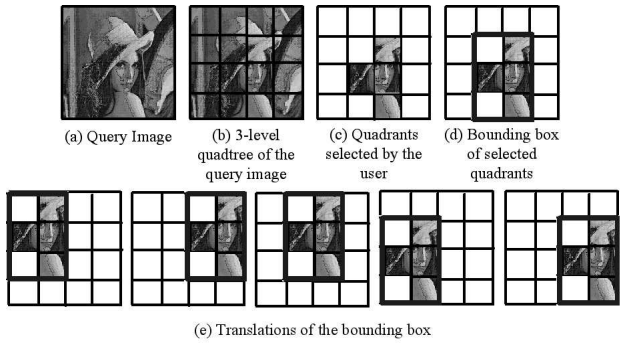

Figure 6: A pattern query (adapted from [15]).

To perform pattern query, the user specifies weights w_n and a threshold α for each selected quadrant. Existing image quadrants of the database can also be selected as query pattern - see the example represented in Figures 6.(a) to 6.(c), adapted from [15]. The CBIR system described in [15] detects the minimum bounding rectangle of the selected quadrants – see Figure 6.(d). The initial query q_p can be transformed into several queries, each one representing a geometric transformation (translation or rotation) of the initial minimum bounding rectangle. In [15], for example, the initial query of Figure 6.(d) is transformed into 6 queries: the initial query of Figure 6.(d) and the 5 queries of Figure 6.(e). Each query results from a translation of the selected quadrants in the initial query q_p. An detailed algorithm to compute Δ_p-distance is described in [9].

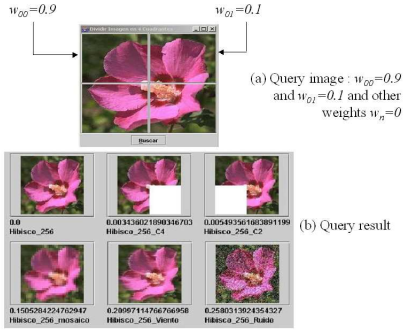

Figure 7: A pattern search using Δ-distance.

Figure 7 shows an example of a pattern search realized by the prototype of [9]. The user wants to retrieve images from the database having NW and NE quadrants of level 1 similar to the homologous quadrants of the query image (see at the top of the figure). Weight 0.9 is associated with the query image quadrant 00 and weight 0.1 with the query image quadrant 01. The other quadrants are associated with a null weight. In this example, no translation of the query

pattern is done.

3.4 Sub-image search using Δ

A multi-level feature vector also allows sub-image search. In this kind of search, a query image is compared with all image quadrants stored in the database, each quadrant being considered as an independent image. Δ-distance can be used to computed such a sub-image search. In this case, $\Delta(q, i)$ is computed for all images i of the database and all image quadrants of the images.

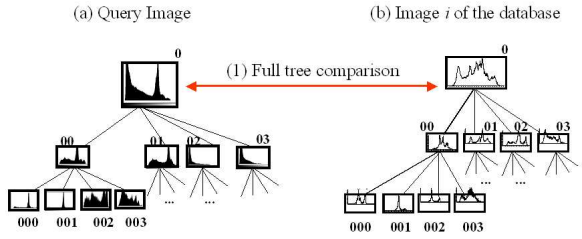

Figure 8: The first step of sub-image search: a global search.

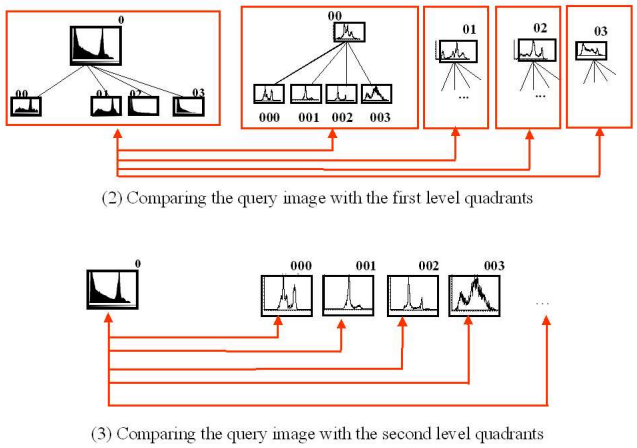

Figure 9: The next steps of a sub-image search: a comparison between the query tree and the subtrees representing the image quadrants.

A sub-image search is performed in several steps: as many steps as the number of levels in the image trees. The first step of a sub-image search is a global search (see Section 3.2). The second step compares the query image with all image quadrants of the first level. The last step compares the query image with the last level quadrants all images. Figures 8 and 9 summarize the sub-image search process, using a quadtree-based multi-level feature vectors – it can be generalized to any tree resulting from a recursive image decomposition. Figure 10 shows an example of a sub-image search, from the prototype of [9]. The query image is on the left. Some images of the database result from a composition of real images in order to see when a quadrant appear at different level. As shown in Figure 10, the result images are globally similar to the query image (see the first returned image) or have some quadrants similar to the query image (see the last three returned images).

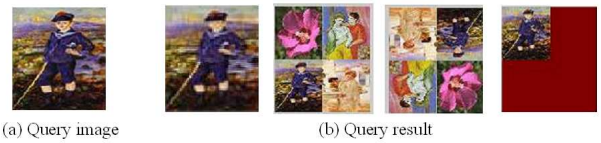

Figure 10: A sub-image search using Δ-distance.

3.5 Particular cases of Δ-distance

Particular cases of Δ-distance can be found in [1, 2, 5, 7, 9, 10, 11, 12, 13, 15, 20], depending on the weights w_n and on the δ-distance used to compare image quadrants n. In the following, due to a lack of space, particular cases presented in this article are limited to the distances used in [9, 10, 11, 12, 13, 15, 20]. For the other approaches, readers are referred to [23].

The authors of [9, 11] and [12] represent each image of the database by a full fixed-depth balanced quadtree whose nodes contain a color feature vector of the corresponding image quadrants. In [11], the root nodes contain average color vectors of the entire images and the other quadtree nodes contain color histograms. In [12], all quadtree nodes contain a color histogram. In both approaches, the distance between images is a Δ-distance such that weights w_n correspond to surface coefficients ($w_n = 4^{-\ell}$ for all nodes n at level ℓ) and δ is an Euclidean distance. Both approaches use a multi-filtering process, using a $\Delta^{(\ell)}$-distance. Moreover, in the DISIMA image DBMS [17], implementing the proposition of [11], sub-image queries can be answered using selected portions of the multi-scale color histograms. This kind of distance can be defined using a Δ_p-distance. In [9], quadtree nodes contain a 9-dimensional color vector based on the first third color moments. This approach allows multi-filtering global search using a $\Delta^{(\ell)}$-distance and pattern or sub-image searches using a Δ_p-distance.

The authors of [15] also represent each image by a full fixed-depth balanced quadtree to perform region queries. The user, after choosing an image from the database, has to select several image quadrants at a chosen level ℓ, using a $16 * 16$ grid. A bounding box is defined as the smallest sub-image containing the selected quadrants (see Figure 6). Then, each image of the database is compared with the initial query image and the query images obtained after a translation of the bounding box containing selected quadrants. The visual content of image quadrants is represented by a high dimensional feature vector combining color and texture feature. The authors define a distance which is a normalized linear combination of distances. This distance corresponds to a Δ_p-distance. All chosen regions in the query image have the same size, thus all nodes n representing the regions selected by the user in the query image have the same value of weight w_n. The values of weights w_n are zero for the other nodes n which do no represent the selected pattern.

In [13], images are represented by a tree structure, whose

first level corresponds to a nona-tree, containing 9 nodes, and the last level corresponds to a quadtree, containing 16 nodes. Tree nodes store the mean and the covariance color of the corresponding image quadrants in the $L^*a^*b^*$ color space. The approach only deals with region-based image retrieval, taking into account only the leaf node distances. The distance computed is a Δ-distance such that weights w_n are zero for all internal nodes n ; only leaf nodes are compared. This approach is extended in [14]. A relevance feedback process (adapting weights w_n) is proposed to improve the image search. Moreover, the tree nodes contain BIC feature vectors based on the *Border/Interior pixel Classification* defined in [26].

In [10], the quadtree decomposition is not applied to images but to shapes contained in the images of the database. A shape is divided into four sub-regions by two principal axes corresponding to the two eigenvectors at the center of mass of the shape. Each sub-region is subdivided into four sub-regions in the same way. The sub-division process is repeated a predetermined number of times resulting in a fixed-depth balanced quadtree whose nodes correspond to the regions of the shape derived from the above process. Four parameters, invariant to translation, rotation and scale, are calculated for the corresponding regions of each node while only two parameters are extracted from the root node. The similarity distance used to compare two shapes corresponds to a Δ-distance where δ is a distance L_1 (say Manhattan or city block[1]) between quadtree nodes, and where weights $w_n = 1$ for all nodes n.

In [20], a nona-tree (with only 49 nodes at level 2, redundant quadrants being stored only once) contains texture vectors. The query image is compared with all the different image quadrants stored in the database, traveling through all nona-trees. This approach deals with sub-image search using a Δ_p-distance. When the query image is compared with an image quadrant n, the compared image quadrant n is associated with weight $w_n = 1$ (the other quadrants being associated with a null weight). The δ-distance used is an Euclidean distance between texture feature vectors.

Table 2, at the end of the article, summarizes these particular cases of Δ-distance.

4. CONCLUSION AND FUTURE WORK

In this article, a generalized metric distance, Δ-distance, is presented for images recursively decomposed into fixed-size quadrants and represented by a multi-level feature vectors. This distance allows specific content-based image retrieval queries. It allows to retrieve images globally similar to a query image, taking into account not only the visual features of the images but the location of the visual features too. Thanks to the tree representation of the images, Δ-distance can be computed using multi-level filtering which reduces the number of images to be compared at each filtering step. Δ-distance also allows region-based queries. In this kind of queries, the resulting images contain quadrants similar to the quadrants selected by the user in the query image (*pattern search*) or contain quadrants similar to the entire query image (*sub-image search*). Because Δ-distance is a generalized distance, some existing CBIR systems [9, 10, 13, 11, 12, 15, 20] use particular cases of it.

The Δ-distance computation is more time-consuming than any global distancescomputed between global feature vectors (for example, the color histogram of the entire image): the time used to compute Δ-distance is proportional to the number of image quadrants resulting from the image decomposition. Thus, index structure must be used to improve the query performances. Several approaches use an index on the root of the multi-level feature vectors: a R-tree [8] in [12], a K-d-tree [3] in [15] or a structure based on extendible hashing [6] in [11]. With this index, the number of images having to be compared, quadrant by quadrant, is limited because the index clusters the images according to their global similarity. Deeper quadtree level comparisons are only performed for the images contained in the filtering result set [9]. As explained in [11], performing the filtering on the entire image database gives a result set containing false hits but no false dismissals. When a system should perform not only global search but region-based search too, an index including all nodes of the multi-level feature vectors is more efficient. Such an index structure is proposed in [9].

This article represents the first steps to develop a prototype allowing users to compare different Δ-based distances. The purpose of such a prototype is to help users to fix the right weights w_n and to choose the δ-distance between image quadrants, depending on the query they want and on the images of their database. In this article, Δ-distance is only defined to compare images represented by a full balanced tree. However, such a tree could be unbalanced, for example when it results from an image segmentation or when it is used for image storage (see in [16] for a survey on quadtree uses in image domain). Particular cases of Δ-distance can be defined for images represented by unbalanced trees [22, 23]. Our future prototype will also integrate this kind of distances.

[1]The sum of the absolute difference of each feature vectors stored in homologous tree nodes.

5. ACKNOWLEDGMENTS

The authors would like to thank the referees of this article and Vincent Oria (NJIT -USA) for their helpful comments and suggestions.

6. REFERENCES

[1] I. Ahmad and W. Grosky. Indexing and retrieval of images by spatial constraints. *Journal of Visual Communication and Image Representation*, 14(3):291–320, 2003.

[2] E. Albuz, E. Kocalar, and A. Khokhar. Quantized CIELab* Space and Encoded Spatial Structure for Scalable Indexing of Large Color Image Archives. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages 1995–1998, Istanbul (Turkey), June 2000.

[3] J. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[4] K. Chakrabarti, M. Ortega-Binderberger, K. Porkaew, P. Zuo, and S. Mehrotra. Similar Shape Retrieval in MARS. In *IEEE Int. Conf. on Multimedia and Expo (ICME II)*, pages 709–712, New York, NY, USA, July/Aug. 2000.

[5] F. De Natale and F. Granelli. Structured-Based Image Retrieval Using a Structured Color Descriptor. In *Int. Workshop on Content-Based Multimedia Indexing (CBMI'01)*, pages 109–115, Brescia (Italy), sept. 2001.

[6] R. Fagin, J. Nievergelt, N. Pippenger, and H. Strong. Extendible hashing - a fast access method for dynamic files. *ACM Trans. Database Syst.*, 4(3):315–344, 1979.

[7] A. Gupta and R. Jain. Visual Information Retrieval. *Communications of ACM (CACM)*, 40(5):70–79, 1997.

[8] A. Guttman. R-trees : A Dynamic Index Structure for Spatial Searching. In *Proc. of ACM SIGMOD Int. Conf. on the Management of Data - SIGMOD Record 14(2)*, pages 45–57, 1984.

[9] G. Jomier, M. Manouvrier, V. Oria, and M. Rukoz. Multilevel index for global and partial content-based image retrieval. In *Proc. of the 1st IEEE Int. Workshop on Managing Data for Emerging Multimedia Applications (EMMA'05)*, pages 66–75, Tokyo (Japan), April 8-9th 2005. in conjunction with 21th IEEE Conference on Data Engineering (ICDE).

[10] H.-K. Kim and J.-D. Kim. Region-based shape descriptor invariant to rotation, scale and translation. *Signal Processing: Image Communication*, 16(1-2):1–293, Sept. 2000.

[11] S. Lin, M. Tamer Özsu, V. Oria, and R. Ng. An Extendible Hash for Multi-Precision Similarity Querying of Image Databases. In *Proc. of the 27th Int. Conf. on Very Large DataBases (VLDB'2001)*, pages 221–230, Roma (Italy), Sept. 2001.

[12] H. Lu, B.-C. Ooi, and K.-L. Tan. Efficient Image Retrieval by Color Contents. In *First Int. Conf. on Applications of Databases (ADB-94)*, Vadstena (Sweden), June 1994. Lecture Notes in Computer Sciences - 819 - Springer Verlag.

[13] J. Luo and M. Nascimento. Content Based Sub-image Retrieval via Hierarchical Tree Matching. In *Proc. of the First ACM Int. Workshop on Multimedia Databases (ACM MMDB 2003)*, pages 63–69, New Orleans, Louisiana (USA), Nov. 2003.

[14] J. Luo and M. Nascimento. Content-Based Sub-Image Retrieval Using Relevance Feedback. In *Proc. of the 2nd ACM Intl. Workshop on Multimedia Databases (ACM MMDB 2004)*, Washington DC (USA), Nov. 2004.

[15] J. Malki, N. Boujemaa, C. Nastar, and A. Winter. Region Queries without Segmentation for Image Retrieval by Content. In *3rd Int. Conf. on Visual Information Systems (Visual'99) - Lecture Notes in Computer Science 1614, Springer Verlag*, pages 115–122, Amsterdam (The Netherlands), June 1999.

[16] M. Manouvrier, M. Rukoz, and G. Jomier. *Spatial Databases: Technologies, Techniques and Trend*, chapter IV - Quadtree-Based Image Representation and Retrieval, pages 81–106. Y. Manolopoulos, A. Papadopoulos and M. Vassilakopoulos (Eds), IDEA Group Publishing, Information Science Publishing and IRM Press, 2005.

[17] V. Oria, M. Tamer Özsu, S. Lin, and J. Iglinski. Similarity Queries in the DISIMA Image DBMS. In *Proc. of ACM Multimedia*, pages 475–478, Ottawa (Canada), Sept. 2001.

[18] J. Park, V. Govindaraju, and S. Srihari. OCR in a hierarchical feature space. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(4):400–407, 2000.

[19] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann. Empirical Evaluation of Dissimilarity Measures for Color and Texture. In *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV'99)*, pages 1165–1173, Sept. 1999.

[20] E. Remias, G. Sheikholeslami, A. Zhang, and T. Syeda-Mahmood. Supporting content-based retrieval in large image database systems. *Multimedia Tools and Applications*, 4(2):153–170, March 1997.

[21] Y. Rui, T. Huang, and S.-F. Chang. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10:39–62, March 1999.

[22] M. Rukoz, M. Manouvrier, and G. Jomier. Distances de similarité d'images basées sur les arbres quaternaires. In *18èmes Journées Bases de Données Avancées (BDA'02)*, pages 307–326, Evry (France), oct. 2002.

[23] M. Rukoz, M. Manouvrier, and G. Jomier. Quadtree-based Distances between Images. Technical Report 37 (Note de recherche en ligne ISSN : 1769-7417), Paris-Dauphine University, 2003. Submitted to publication - Revision in progress.

[24] R. Schettini, G. Ciocca, and S. Zuffi. *Color Imaging Science: Exploiting Digital Media*, chapter A Survey on methods for colour image indexing and retrieval in image databases. J. Wiley, R. Luo and L. MacDonald (Eds.), 2001.

[25] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content based image retrieval at the end of the early years. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[26] R. Stehling, M. Nascimento, and A. Falcao. A Compact and Efficient Image Retrieval Approach Based on Border/Interior Pixel Classification. In *Proc. of the ACM Intl. Conf. on Information and Knowledge Management (CIKM'02)*, pages 102–109, McLean, VA, USA, 2002.

[27] W. Wang, A. Zhang, and Y. Song. Identification of Objects from Image Regions. In *IEEE Int. Conf. on Multimedia and Expo (ICME 2003)*, Baltimore (USA), July 2003.

Table 1: Meaning of symbols used

Symbol	Meaning
ℓ	a tree level
n	tree node or image quadrant identifier
x	integer, $x \in \{0,1,2,3,4,5,6,7,8\}$
$\mathcal{N}$	set of all nodes (resp. quadrants) identifiers appearing in the image database
$\mathcal{N}^{(\ell)}$	set of all nodes (resp. quadrants) identifiers appearing at level ℓ
i, i', j	image identifiers
q	query image
q_p	query pattern (i.e. quadrants selected in an image recursively decomposed)
$\Delta(i,j)$	distance between multi-level feature vectors of images i and j
$\delta(i,j,n)$	normalized metric distance between feature vectors of homologous quadrants n in images i and j
w_n	weight of homologous nodes (resp. quadrants) n in the Δ-distance computation
$\mathcal{W}$	sum of all weights w_n associated with the image quadrants, $\mathcal{W} = \sum_{n \in \mathcal{N}} w_n$
$\Delta^{(\ell)}(i,j)$	approximation of Δ-distance computed from nodes appearing from the root level (0) until level ℓ
$\widetilde{\Delta^{(\ell)}}(i,j)$	approximation of Δ-distance taking into account only nodes at level ℓ
α	similarity threshold ; images having a Δ-distance below α are similar
$\mathcal{R}^{(\ell)}$	result set of the $(\ell+1)^{\text{th}}$ step of the multi-level filtering process
$\Delta_p(i,j)$	Δ-distance computed for a query pattern q_p

Table 2: Particular cases of Δ-distance

Ref.	Tree	Feature vectors	Δ-distance	δ	weights w_n
[9]	quadtree	color moments	$\widetilde{\Delta^{(\ell)}}$ and Δ_p	L_2	$4^{-\ell}$ for all nodes n at level ℓ
[10]	quadtree	shape feature	Δ	L_1	1 for all nodes n
[11]	quadtree	average color in the root and color histograms in the other nodes	$\widetilde{\Delta^{(\ell)}}$ and Δ_p	L_2	$4^{-\ell}$ for all nodes n at level ℓ
[12]	quadtree	color histograms	$\widetilde{\Delta^{(\ell)}}$	L_2	$4^{-\ell}$ for all nodes n at level ℓ
[13]	mixed between nona and quadtree	mean and covariance color	Δ	L_1	$w_n = 0$ for all internal nodes n and $w_n = 1$ for leaf nodes n
[15][2]	quadtree	histograms combining color and texture	Δ_p	d	$w_n = 0$ for all quadrants n not selected by the user
[20]	nona-tree	texture vector	Δ_p	L_2	$w_n = 1$ for each compared quadrants n $w_n = 0$ otherwise

[2]In [15]: $\delta = d$, where d is a specific distance based on Gaussian assumption and using the Kronecker symbol.