A Scalable Distributed Data Structure for Multi-Feature Similarity Search
(extended abstract) *

Claudio Gennaro‡, Matteo Mordacchini‡, Salvatore Orlando∗‡, Fausto Rabitti‡

ISTI - CNR, Area della Ricerca di Pisa
Via Giuseppe Moruzzi, 1
56124 Pisa, Italy

Dipartimento di Informatica
Università Ca’ Foscari
Via Torino, 155
30172 Venezia, Italy

Abstract. Similarity search for content-based retrieval (where content can be any combination of text, image, audio/video, etc.) has gained importance in recent years, also because of the advantage of ranking the retrieved results according to their proximity to a query. However, to use similarity search in real world applications, we need to tackle the problem of huge volumes of such mixed multimedia data (e.g., coming from Web sites) and the problem of their distribution on multiple cooperating nodes. The proposed approach is being used in two running projects: SAPIR and NeP4B. In this paper we approach this problem by considering a scenario of a network of autonomous peers maintaining a local collection of metric objects (i.e., mixed mode multimedia content). This network forms a distributed Peer-to-Peer (P2P) search engine for similarity search based on the paradigm of Routing Index. Each peer in the network thus maintains both an index of its local resources and a table for every neighbor, summarizing the objects that are reachable from it. The paper presents techniques that aim to make our P2P similarity-based search system viable, trading approximate results for scalable solutions. Results of simulations that use real collections of images are discussed.

1 Introduction

Nowadays the search for non-textual data (e.g., images, audio, or videos) is mostly done by exploiting the metadata annotations or extracting the text close to the data. For example, in order to performing image discovery, Web search engines simply index the text near the image and the image tags, used to provide a description of an image. Images indexing methods based on content-based analysis and resulting image features (such as colors and shapes) are not exploited

* This work was partially supported by the SAPIR (Search In Audio Visual Content Using Peer-to-Peer IR) project, funded by the European Commission under IST FP6 (Sixth Framework Programme) and by the NeP4B project (Networked Peers for Business), funded by the Italian government
at all. These limitations can be overcome if a similarity based search approach is adopted: content-based retrieval is supported also when the content can be any combination of text, image, audio/video, etc. Moreover, this approach has also the advantage of ranking the retrieved results according to their proximity to a query, and supports relevance feedback [9]. The similarity search approach has been extensively used in content-based image retrieval systems, that retrieve images whose visual appearance is globally similar to a selected example image. This approach, initially proposed by Swain and Ballard [8], was then adopted by a vast majority of content-based image retrieval systems (e.g., IBM QBIC, VisualSEEK, Virage’s VIR Image Engine, and Excalibur’s Image RetrievalWare).

In this paper we propose a scalable P2P indexing structure, based on Routing Index (RI) [4], able to cope with a huge number of indexed objects and queries. According to this approach, no global indexing phase is required, because each peer node is responsible for storing and indexing its own objects. Also adopting this approach, the similarity-based search exploits objects characterized by a metric space, while the multi-feature queries are easily supported. Our proposal, called MRoute, intend to explore the opportunity of combing the advantage of the Routing Index paradigm with the Similarity Search on Metric Spaces approach.

The rest of the paper is organized as follows. In sec. 2 we present our proposed approach. More precisely, in sec. 2.1 we show how our network is structured and formed and give some definition about our system. In sec. 2.2 we define our indexing system and show how it is used to spread objects feature in the network. Sec. 2.3 explain how indices are exploited for routing queries toward potential matchings. Results of simulation experiments with our network are presented in sec. 3.

2 MRoute - Multimedia Routing Index

In this section, we describe our proposal of Multimedia Routing Index that supports similarity search over sets of multimedia attributes for content-based retrieval. This problem of similarity searching can be formalized by the mathematical notion of metric space, so data elements are assumed to be objects from a metric space domain where only pairwise distances between the objects can be determined by a respective distance function. More formally, a metric space is defined by a domain of objects $D$ (elements, points) and a distance function $d$ – a non-negative and symmetric function which satisfies the triangle inequality $d(X,Y) \leq d(X,Z) + d(Z,Y)$, $\forall Z,Y,Z \in D$. The focus of this paper is on similarity-based Range Queries over multi-feature metric objects, defined as follows: given a database $D \subseteq D$ of objects, a query object $Q \in D$, and a positive range $r \in \mathbb{R}$, a query $Q = (Q, r)$ has to retrieve the set $\{X \in D \mid d(X, Q) \leq r\}$.

In the rest of this paper we use the following conventions. We consider a P2P network composed by $N$ peers. Each peer $P$ knows a a set of other peers directly connected to it. This set is called the neighborhood of $P$ and is denoted with $\text{Nb}(P)$. Moreover, $P$ owns a set of data objects, called local repository. We
denote it with $Data(P)$. We use the qualifier “data” in order to distinguish it from metric objects. Every data object $O$ of such repositories is characterized by a set of multimedia features. A feature is a metric object extracted from the data object. For instance a data object could be a jpeg image, from which we can extract two features, e.g. the texture and the color-histogram. $O^F$ indicates the value of feature $F$ for $O$, and $D^F$ its corresponding domain of metric objects.

### 2.1 Network Organization

Routing Indices (RIs) are thought for unstructured P2P networks. No organization in special shapes are required, in contrast to the DHT-based networks. Nonetheless, particular P2P topologies may pose problems for RIs. Each RI is associated with a network link, and collect and summarize information about objects located in the portion of the network that can be reached along this link. If the network present a loop, the information about the peers involved in the loop could be duplicated. To avoid this problem, we use a node caching mechanism to avoid repeated updates of our indices. The final effect is that we can consider the network topology of our system as a logical tree. Accordingly, the RI associated with a link $P \rightarrow P_j$ represents the values of all the objects stored in the logical sub-tree, rooted on $P_j$ and denoted by $T(P \rightarrow P_j)$.

### 2.2 Data Indexing

The data indexing process is intended to produce a summarized description of all the features of the objects owned by each peer in order to produce efficient multi-feature RIs. The aim of these indices is to provide a concise but yet sufficiently detailed description of the resources available in a given network area. This information is then used at query-resolution time to prune entire system zones from searching, thus easing the search process. To introduce the problem of data indexing, we refer to a single feature scenario. Therefore, for the sake of simplicity, we start by omitting to add a superscript symbol $F$ to each data object $O \in D$.

When dealing with metric objects, the triangle inequality property is exploited to build indices that allow us to select only some subsets of candidate objects, among which the objects relevant to the query can appear. These indices permit to save work: only for each candidate object $O_{cand}$ we need to compute the distances $d(Q, O_{cand})$, in order to determine the relevant objects for query $Q = (Q, r)$.

Several indices have been proposed so far [3]. We will refer to the the well known technique of reference objects (or pivots) to transform each indexed object (and also query object) into a vector of distances from the $m$ reference objects $R = \{R_1, \ldots, R_m\}$. As a consequence of the triangle inequality, given the vector of distances $d(Q, R_1), \ldots, d(Q, R_m) \in \mathbb{R}^m$, the objects $O \in \mathbb{R}^m$ relevant for the query $Q$ within a radius $r$ are those such that each component $d(O, R_i)$ of the vector associated with $O$ falls into interval $[d(Q, R_i) - r, d(Q, R_i) + r]$. In the presence of multi-feature data objects, we have to take into account that...
each feature is characterized by its own set of reference metric objects and, consequently, its own distance definition. Usually, the metric distance $d$ between multi-feature data objects is defined as a linear combination of the various metric distances $d_j$, where each $d_j$ is defined on the basis of the $j$–th feature extracted from each object:

$$d(X, Y) = \alpha_1 d_1(X^{F_1}, Y^{F_1}) + \ldots + \alpha_n d_n(X^{F_n}, Y^{F_n})$$

where $\alpha_j \in \mathbb{R}$, $\alpha_j \geq 0$, are scalar coefficients, and $X^{F_i}, Y^{F_i} \in D^{F_i}$. If each component $d_j$ of this linear combination is metric, it can easily be shown that also distance $d$ is metric. In this way, we obtain a new (combined) metric space $D$ that is the result of the fusion of $n$ metric spaces $(features) D^{F_1}, \ldots, D^{F_n}$.

While we can always define an index on the basis of the total distance $d$, thus considering all the features extracted from the multimedia objects, one can also be interested in making the query definition more flexible for the users. For example, to allow them to specify similarity-based queries that only consider a subset of all the features extracted from the multimedia objects, or to change the coefficient $\alpha_j$ of the linear combination defined above.

As stated in the previous sections, we make use of RIs for forwarding queries toward network areas of potential interest. RIs provide a concise representation of the presence and relative amount of potentially relevant data objects, available in a given zone of a P2P network. In order to achieve this goal, for each object $O$ we need a different representation of $O$’s features values. For each feature $F_j$ and its set of associated reference objects $R^{F_j} = \{R_{F_j}^1, \ldots, R_{F_j}^m\}$, we encode the value $d_j(O, R_{F_j}^i)$, $i = 1, \ldots, m$, with a $k$ binary vector:

$$DataIdx(O)_{R_{F_j}^i} = (b_0, b_1, \ldots, b_{k-1})$$ (1)

such that $b_c = 1$ if and only if $d(O^{F_j}, R_{F_j}^i) \in [a_c, a_{c+1})$. An example of such a representation is given in Fig. 1. Both the parameter $k$ (number of elements of the vector) and the division points $a_0, a_1, \ldots, a_k$ may be different for each reference object and each feature type. Thus, each feature is now characterized by a set of $m$ bit vectors, one for each of the feature’s reference objects. Each of these vectors shows which is the interval in which falls the real distance of the
metric object from a reference metric object. Such simple object indices become useful to construct a concise description of all the data owned by a peer. Given a peer $P$ and a reference object $R^f_j$, it is obtained by simply summing up the indices of all the objects of the local repository of $P$:

$$\text{NodeIdx}(P, F^i_j)_{R^f_j} = \sum_{O \in \text{Data}(P)} \text{DataIdx}(O)_{R^f_j}$$

The final result is an integer vector, i.e. a histogram, that represents how the objects of a peer $P$ are distributed with respect to the feature of the reference object $R^f_j$.

The next step is concerned with the construction of the various RIs, built by spreading out information about data object location along the network. Let us consider a generic peer $P$. For each neighbor $P_j \in \text{Nb}(P)$, $P$ keeps information on the data items which can be found by following the link $P \rightarrow P_j$ on the overlay network. For this purpose, $P$ maintains a RI called $\text{LinkBitIdx}(P \rightarrow P_j, F^i_j)_{R^f_k}$, for each reference object $R^f_k$ of each feature $F_j$ of the data objects in $T(P \rightarrow P_j)$. Since $P$ can receive information only from its neighborhood, the index is calculated recursively in the following way:

$$\text{LinkBitIdx}(P \rightarrow P_j, F^i_j)_{R^f_k} \equiv \left( \sum_{P \in \text{Nb}(P)} - P \text{LinkBitIdx}(P_j \rightarrow P, F^i_j)_{R^f_k} \right)$$

The final result is that the index contains the sum of all the vector indices $\text{NodeIdx}(P, F)_{R^f_k}$ associated with every peer in $T(P \rightarrow P_j)$.

### 2.3 Query Processing

The query process starts when a peer receives a query from a user. We assume that queries could be originated from any node $P$ in the system and could involve an arbitrary number of features. Let $Q$ be a query and $r$ the query radius. Since $Q$ is defined over a set of attributes, its radius is obtained as a linear combination of the radiuses of the sub-queries over each single feature. Thus, we have that $r = \alpha_1 r_1 + \ldots + \alpha_f r_f$, where $f$ is the number of features requested by $Q$.

In order to compare $Q$ with the RIs defined in the previous section, $Q$ is translated into a set of bit vectors. For each single feature $F_j$ and its associated set of reference objects $R^f_j = \{ R^f_{j1}, \ldots, R^f_{jm} \}$, we consider the interval defined by $[d(Q^F_j, R^f_{j1}) - r_j, d(Q^F_j, R^f_{jm}) + r_j]$, where $r_j$ is the radius of $Q^F_j$ with respect to $F_j$. A $k$-bit vector is constructed by setting to 1 all the entries that correspond to intervals that are covered (even partially) by the requested interval. All the other entries are set to 0. Such an index is denoted as $\text{QueryIdx}(Q^F_j)_{R^f_j}$.

These indices can be directly compared with the RI ones, in order to find which are the neighbors connected with the network zones with the highest chance to contain relevant objects. The matching phase is performed in the following way. Let $P$ be the peer that receives a query $Q$, $\text{QueryIdx}(Q^F_j)_{R^f_j}$ the
set of indices of the query relative to the feature $F_j$ and $\text{LinkBitIdx}(P \rightarrow P_j, F_j)$ the RIs on the same feature for a peer $P_j \in \text{Nb}(P)$. For each reference point $R_{ij}^F$ of $F_j$, the indices $\text{QueryIdx}(Q_{Fj}^i)$ and $\text{LinkBitIdx}(P \rightarrow P_j, F_j)_{R_{ij}^F}$ are compared. A match is considered to happen if and only if there exists at least one entry $e$ for which both indices have a value greater than zero. In order to forward $Q_{Fj}$ from $P$ to $P_j$, all the indices of all the reference points of all the features must match the query corresponding ones.

Since the RIs can also be regarded as histograms of the distribution of objects with reference to every feature, it is also possible to know the number of potentially matching objects. This possibility gives us the ability to perform different query forwarding strategies. One of these strategies consists in allowing a peer $P$, which has to forward a query, to select its neighbors on the basis of the number of potential matchings they have in their subnetworks. The selection is made using what we call the requested coverage, i.e. the neighbors the query is to be forwarded to must have a number of matchings that is equal or above a given percentage of all the possible matchings given by the RIs. The aim of this process is to further reduce the number of query messages, while trying to collect the largest number of results.

3 Simulation Results

In this section we present the results obtained by a simulation of our system. The aim of this simulation experiments is to the number of nodes that receives a query because their indexes show that they have potential matchings for the query. Simulation results were computed as intervals with 90% confidence level and each measurement was repeated multiple times in order to get confidence intervals having width of less than 5% of the central value. The simulation model has been implemented using the C++ library described in [5]. The simulation settings are as follows: The network topology is a tree generated randomly, with at most five children per node; We use a network with 204 nodes that represent an equal number of clusters of the available data; The object used in the network are 159805 real images taken from the Flickr archive (http://www.flickr.com). The clusters are not disjunct, i.e., although each data object is assigned univocally to only one peer, the space supervised by each peer may overlap with the one owned by another peer. Each object is characterized by two MPEG-7 standard features, scalable color and edge histogram. We used 10 different reference objects for constructing the indices. They are chosen as 10 random objects taken from the 10 peers with the largest local repositories.

Fig. 2 illustrates the nodes involved in the forwarding of queries that use both the feature at the same time. The combined search involves from about 12% to 20% percent of the total resources. The fact that the number of nodes involved exceeds the number of the resource that exactly match the queries is due to false positives derived from the approximation involved in the index creation. We have compared the performances of the combined search with the ones of the searches for each single feature, using the same network and the same radii
used for the combined queries (see 3(a) and 3(b)). The results show that the use of the indices allow to pruning from searching uninteresting network areas both for multi- and single-feature queries. Thus, the RIs achieve the primary goal of creating an efficient routing system that could exploit all the features of the objects, singularly or in a combined manner. As can be seen, the result of the combined search is better of that of each single resource. This is due to the fact that the use of the same radii in the combined searches led to more selective queries. Thus the data and the peers involved are fewer than that each single feature query.

![Graph showing the percentage of nodes involved in the combined range query](image1.png)

**Fig. 2.** Nodes involved in the combined range query

![Graphs showing the percentage of nodes involved in the propagation of single-feature range queries](image2.png)

**Fig. 3.** Nodes involved in the propagation of single-feature range queries

### 4 Conclusions and Future Work

In this paper we presented an approach for distributed similarity search on a network of autonomous peers maintaining a local collection of metric objects.
with mixed mode multimedia content (i.e., any combination of text, image, audio/video, etc.). This network forms a distributed Peer-to-Peer (P2P) search engine for similarity search based on the paradigm of Routing Index (MRoute – Multimedia Routing Index).

These are the operative assumptions of SAPIR and NeP4B. In the context of these projects we will have the opportunity to test and validate these techniques with really large volumes of multimedia data.

We have presented some simulation results showing the characteristics of MRoute, that lead to the conclusion that it fulfill the operation requirements of the NeP4B projects, and in general the requirements of an index to support large scale multimedia similarity based searching applications.

In the context of the NeP4B project, we intend to pursue this research by integrating our Multimedia Routing Index with a Semantic Routing Index. In fact, a Semantic Routing Index is based on the semantic mappings between peers, exploiting the mapping between the single elements of the domain ontologies of each peer. A query on a NeP4B semantic peer can be propagated to the other peers through these semantic mappings. Integrating the two mechanisms (Multimedia Routing Index and Semantic Routing Index) would allow the P2P system to try to answer user queries also in these cases, quite common in the real world, where the sought complex objects are poorly described in the peer ontologies or the the semantic mapping are imprecise or incomplete.

References