SIXTH FRAMEWORK PROGRAMME
PRIORITY 2
"Information Society Technologies"

Deliverable D4.2
<Design of the Multimedia Indexing Technique>
January 2008

Project acronym: SAPIR
Project full title: Search on Audio-visual content using Peer-to-peer Information Retrieval
Contract no.: 45128

Deliverable type: Report
Classification: R
Work package and task: WP4, T4.2
Responsibility: MPG

Editor: Mouna Kacimi

Contributors: Mouna Kacimi, Matteo Mordacchini , Fausto Rabitti, Gerhard Weikum

Internal Reviewer: Raffaele Perego

EXECUTIVE SUMMARY

This report presents the activities conducted within Task T4.2 of the SAPIR project. It discusses the state-of-the-art of peer-to-peer (P2P) technology for indexing and searching multimedia content including images, text, and user-provided tags. The focus is on peer-centric overlay networks of autonomous peers where each peer builds a local index from its own personal data, focused Web crawls, or data gathered from external sources. To facilitate global search, indexes contain peer-summary information regarding features that can be used for similarity search and ranking algorithms. The main goal is to support efficient routing of queries to a small number of judiciously chosen peers that can deliver high-quality results at acceptable execution cost. The report contains preliminary results achieved in Task T4.2.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 OBJECTIVES OF WP4

In this work package, we aim to provide methodologies for building a peer-to-peer (P2P) network that crawls, indexes, and searches large volumes of multimedia data. We focus on ***peer-centric overlay networks*** of autonomous peers where each peer builds a local index from its own personal data, focused Web crawls, or data gathered from external sources. Such networks are adequate for P2P applications that emphasize the content ownership and autonomy of the individual peers, whereas other applications may be based on network-centric overlays with peers dedicating their storage and computational resources to a jointly maintained global indexing and similarity-search infrastructure. The SAPIR architecture supports both peer-centric and network-centric overlays; the latter class is discussed in Deliverable D4.1. For peer-centric overlays as discussed here, a key issue is how to route queries to a small number of most promising peers based on peer-summary information and other statistics. In order to improve the performance of the query execution, caching mechanisms are pursued for keeping the results of frequent queries in many (or all) peers. For content dissemination, a novel distributed push-based crawling model is investigated, where content providers publish and “push” information to the P2P indexing nodes instead of being visited by crawling agents.

1.2 OBJECTIVES OF DELIVERABLE D4.2

The objective of this deliverable is to define a P2P infrastructure for indexing and searching audio-visual content in a peer-centric overlay network of peers with autonomous contents. The overlay network forms a logically global but physically distributed search engine, where each peer has its own local index for its private repository built from the peer’s personal data, focused Web crawls, or other kinds of imported content sources. The key mechanism for retrieving data is query routing: identifying the best peers for executing a query in order to return high-quality results at acceptable execution cost. The goal is to define appropriate strategies for query-routing decisions based on peer summaries and other kinds of statistics about data and workload.

In general, the design of the P2P multimedia indexing and query routing infrastructure should have the following properties:

- **Decentralization**: Data structures should be distributed among peers, for load sharing, resilience to failures, and high availability.
- **Load balancing**: The cost of maintaining indexes should be equally divided between peers. Additionally, the index structure should be able to sustain load peeks caused by skewed access to data items. When a particular item becomes popular, the load of some peers and their network neighborhood increases; appropriate mechanisms such as caching and replication need to accommodate such load shifts.
- **Scalability**: The P2P indexes should be able to handle increasing data volumes and access load. Particularly, scalability of indexing refers to the capability to increase total throughput as the numbers of data items and peers grow.
- **Dynamic maintenance**: The autonomy of peers may result in a high rate of new peers joining and existing peers leaving the overlay network, the so-called churn phenomenon. This poses challenging issues related to index maintenance and adaptive self-configuration.

2. OVERVIEW OF P2P INDEXING

2.1 ARCHITECTURE

In this report, we address *peer-centric* architectures where each peer comes with its own content and is willing to collaborate in an overlay network but without giving up its autonomy. Such architectures may be desired for dynamic federations of independent content providers, or for large-scale social networks that connect millions of users based on the users' personal computers for privacy and autonomy. A peer's content could be built from the peer's own crawls or imported from external sources and tailored to the user's thematic interest profile. Every peer in a peer-centric network creates a repository index that describes the peer's own content and supports local search. A peer may share its content (or specifically chosen parts of it) by posting meta-information and associated statistics to the P2P network. This meta-information contains compact statistics and quality-of-service information, and effectively forms a global index.

A key aspect of peer-centric approaches is query routing that aims to select the most promising peers for a given query. This problem is known in distributed information retrieval as database selection problem [Callan2000, Meng2002, Losee2004]. A query posed by a user can first be executed on the user's own peer, but can be additionally forwarded to a selected set of peers for better result quality. A peer uses the global index to identify candidate peers that are most likely able to provide good query results. The best peers for a given query can be selected according to different criteria including their availability, their network communication cost, the thematic relevance of their content to the query, and the novelty of their results. The results provided by the peers to which the query is forwarded should not highly overlap but rather complement each other.

Peer-centric approaches emphasize the peers' autonomy and require dynamic self-organization mechanisms. Like all forms of distributed indexing, these approaches are potentially susceptible to bottlenecks caused by load imbalance and load surges which could affect the system's scalability. For example, when a particular peer contains some popular data, it might become overloaded and form a bottleneck.

2.2 DISTRIBUTED DATA STRUCTURES FOR PEER-CENTRIC INDEXING

P2P systems are based on various forms of distributed data structures that support mapping keys to content object locations in a decentralized manner so that routing scales efficiently over a high number of peers [Steinmetz2005].

Many proposals for peer-centric approaches have been made in the literature. A first generation of peer-centric approaches was build on top of unstructured networks that organize peers in a flat random graph, where all participant peers have the same functionalities [Gnutella2003]. Each peer holds a local index of its content. The data structure of the local index varies according to the type of the data (text, image, audio, video) and might take various forms such us inverted lists, R-trees, etc. Additionally, each peer holds a set of pointers to its neighbors in the random graph. In such networks, queries are solved using flooding techniques, also known as epidemic messaging [Chawathe2003, Ganesan2003, Ratsnasamy2002].

To limit flooding by sending queries to a selected set of peers, several approaches compute binary relations among peers. In other words, they create routing indices with peer summaries from local neighborhoods using different properties such as network related information [Agrawal2003], application needs [Marti2004, Wang2004], peer characteristics

[Loeser2003], and similarities between peers contents or interests [Hang2002, Nejdl2003, Sripanidkulchai2003, Tang2003, Gupta2004, Crespo2004 , Mislove2006, Gennaro2007, Raftopoulou2008]. Gennaro et al. propose a scalable P2P indexing structure, based on a Routing Index (RI) [Crespo2002], which is able to cope with a huge number of indexed objects and queries. In this context no global indexing phase is required, because each peer node is responsible for storing and indexing its own objects. This network forms a distributed Peer-to-Peer search engine for similarity search on metric space based on the paradigm of Routing Index called MRoute [Gennaro2007]. 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. Sripanidkulchai et al. [Sripanidkulchai2003] take into account query traces over the P2P network to organize peers. Hang et al. [Hang2002] generate, for each peer, signature vectors based on the low level features (color, texture, shape) of its contained images. Other approaches [Crespo2002, Nejdl2003, Gupta2004] associate peers with semantic descriptions that can be simple keyword-based annotations, schema or ontologies.

Many other proposals for peer-centric approaches are based on structured networks. These approaches build peer summaries based on statistical synopses such as Bloom filters or hash sketches maintained in a directory based on distributed hash tables (DHT) [Bender2005a, Michel2006b, Ramabhadran2004, Cuenca-Acuna2003, Mueller2005, Podnar2007]. The *Minerva* system for P2P search [Bender2007] combines local index structures of autonomous peers with a global directory based on a distributed hash table as an overlay network. The data collection of a peer is locally indexed using inverted lists, where for each key (e.g., keyword or term) is associated the list of the corresponding URLs of Web pages or other files. Minerva maintains a global index for peer-granularity meta-information that holds compact, aggregated summaries of the peers' repository indexes. The global index implementation is based on Pastry [Rowstron2001], a distributed hash table (DHT).

2.3 SAPIR INDEXES

We have defined multiple types of indexes in the SAPIR architecture [Kacimi2008]. These indexes provide a Key $\rightarrow$ TargetObject mapping and a management mechanism, i.e. methods like lookup of Keys, insert, and delete. We have defined specific properties for these indexes to highlight the differences between them. Table1 summarizes the different types of indexes.

The Global Index is a conceptually global structure but physically distributed among peers. The Local Index stores the fraction of the Global Index for which a peer is currently responsible. The Shortcut Index is an additional index that stores beneficial (key, value) pairs to speed up information retrieval. The Repository Index is private to each peer and is used to index Content Objects in the Local Repository.

2.3.1 Index Granularity

An index can have either Peer granularity or Content-Object granularity. In the first case, the index captures information about peers' identifiers and possibly a set of statistics about the peers with regard to the associated Key (e.g., the Content Object frequency of the Key at the given peer). In the second case, the index captures information about Content Objects URI and possibly a set of Content Object specific statistics with regard to the given Key (e.g., the frequency of the key in the Content Object).

Properties / Index types	Granularity		Scope		Completeness		Freshness	
	Peer	Content Object	Local	Global	Total	Partial	Always Fresh	Possibly Stale
Global Index	×	×		×	×		×	
Local Index	×	×		×	×		×	
Shortcut Index	×	×		×		×		×
Repository Index		×	×			×	×	

Table 1- Index Types and their Properties

2.3.2 Index Scope

An index can contain information about the local data of a peer. In this case the Index has a Local property. In case the index contains information about remote peers or remote Content Objects, it has a Global property.

2.3.3 Index Completeness

An index can contain the complete information about all objects (Content Objects or Peers) for the keys that it is responsible for. In this case the Index has a Total property. In case the index contains only a subset of the objects that it could potentially know about, it has a Partial property. The latter typically holds for a Shortcut Index, but could also be useful in randomized indexing strategies.

2.3.4 Index Freshness

The information stored in an index can be fresh by always updating the system with the insertions and deletions of Content Objects and the departures and arrivals of Peers. Alternatively, an index may also contain stale information when dealing with Shortcut Indexes or when immediate updates are too expensive.

3. MULTIMEDIA DATA

Multimedia data such as images, audio, and video are indexed in such a way so as to facilitate content-based queries for retrieving similar objects. The idea is to have: (1) a text description that may include annotations such as “social tags”, (2) an MPEG-7 description that contains multimedia feature representations such as color, texture, and contours for images, phonemes for speech, etc. In this report, we focus on how we build indexes on top of these descriptions, for both text and multimedia features. The description of a content object is an XML file in a format derived from MPEG-7 (ISO/IEC 15938). An example of this description is given in Figure 1. This is the specific description format adopted by SAPIR; more details about the representation of audio-visual content is given in Deliverable D3.1 [Kaplan2007].


```
<SapirMMObject>
        <MediaLocator>
                <MediaUri>46030169</MediaUri>
        </MediaLocator>
        <photo id="46030169" secret="ee60cce333" server="31" farm="1"
        dateuploaded="1127547199" isfavorite="0" license="2" rotation="0"
        originalsecret="ee60cce333" originalformat="jpg">
        <owner nsid="24007924@N00" username="Ko(char *)hook" realname="Nick
        Kocharhook" location="San Francisco, CA, USA"/>
                <title>Me Carrying a Beam</title>
                <tags>
                  <tag id="876598-46030169-731" author="24007924@N00"
                   raw="me" machine_tag="0">me</tag>
             </tags>
           < comments photo_id="46030169">
                   <comment id="876598-46030169-12605547" author="79261828@N00"
              authorname="jmissig" datecreate="1127576815">The expression on your
               face is great.</comment>
           </comments>
        </photo>
<Mpeg7>
        <Description type="ContentEntityType">
            <MultimediaContent type="ImageType">
                <Image>
                        <VisualDescriptor type="ScalableColorType"
                        numOfBitplanesDiscarded="0" numOfCoeff="64">
                                <Coeff> 84 -25 -56 61 -12 -9 3 27 -10 -9 24 44 -19 1 9
                                26 7 -2 -3 12 -3 -1 0 8 -15 3 -3 8 -15 5 -10 18 3 -1 -2 -2
                                0 -1 1 2 5 -5 -5 3 1 2 4 9 6 -3 2 3 5 7 6 3 -4 0 0 -2 10 0
                                -3 1 </Coeff>
                        </VisualDescriptor>
                        <VisualDescriptor type="ColorStructureType" colorQuant="2">
                                <Values> 106 0 0 0 88 7 0 119 150 135 7 0 0 0 0 0 17
                                48 0 25 0 2 91 68 130 146 119 107 42 55 25 6 21 46
                                25 32 40 48 27 42 139 111 62 85 111 86 44 47 46 37
                                30 56 28 16 16 54 145 151 89 58 47 35 93 103
                                </Values>
                        </VisualDescriptor>
                </Image>
            </MultimediaContent>
        </Description>
</Mpeg7>
</SapirMMObject>
```


Figure 1: Image Description in MPEG-7-style XML

4. INDEXING TEXT

We use a Minerva-style approach [Bender2005b] for indexing text information related to audio-visual Content Objects. The text part of the XML files describing a peer's content is parsed (optionally using stemming and/or stopword elimination) to build a text repository index on each peer. A repository index takes the form of inverted index lists [Zobel2006]. All terms that appear in the peer's collection form a tree-like index structure (often a B+-tree) where the leaves contain term-specific lists of unique content-object identifiers for all content objects that contain the corresponding term. This is illustrated in Figure 2. For query processing, these inverted lists can be combined by intersection or union for all query terms to find candidate content objects for a given query. Depending on the query execution strategy, the lists of content-object identifiers can be ordered by identifiers (for efficient list merging) or by score values (for efficient pruning). The per-term scores themselves are precomputed using a wide variety of information-retrieval models (e.g., tf*idf-based vector-space IR, probabilistic IR, or statistical language models), to support ranked-retrieval queries.

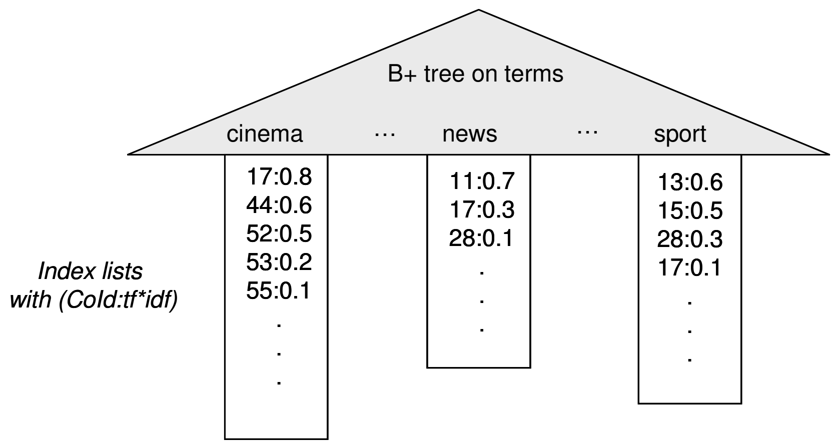

Figure 2: B+ Tree of Inverted Index Lists (Repository Index)

tf*idf scoring is one of the popular schemas that is often used by search engines to score and rank an object's relevance to a given query. The number of occurrences of a term t in a document d is called term frequency and typically denoted as tf. The number of documents in a collection that contain a term t is called document frequency and denoted df. To compute the weight (term-specific score) wi,f of the ith term in the jth document we use the tf*idf measure:

$$w_{i,j}= (tf_{i,j}/max\{tf_{t,j}\}) * log(N/df_i)$$

where N is the total number of documents in the collection. Note that we can use different scoring schemas such as Okapi BM25 (based on probabilistic IR) [Manning2007] to compute the relevance of a document to a given query term.

Every peer publishes a summary (Post) for every term in its repository index to the global index (Figure3), which is routed to the peer currently responsible for this term (using the DHT). A hash function is applied to the term in order to determine the peer currently responsible for this term. This peer maintains a PeerList of all postings for this term from peers across the network. Posts contain contact information about the peer who posted the summary together with statistics to calculate IR-style measures for a term (e.g., the size of the inverted list for the term, the maximum average score among the term's inverted list entries, and further statistical measures). These statistics are used to support the peer selection process, i.e., for determining the most promising peers for a particular query.

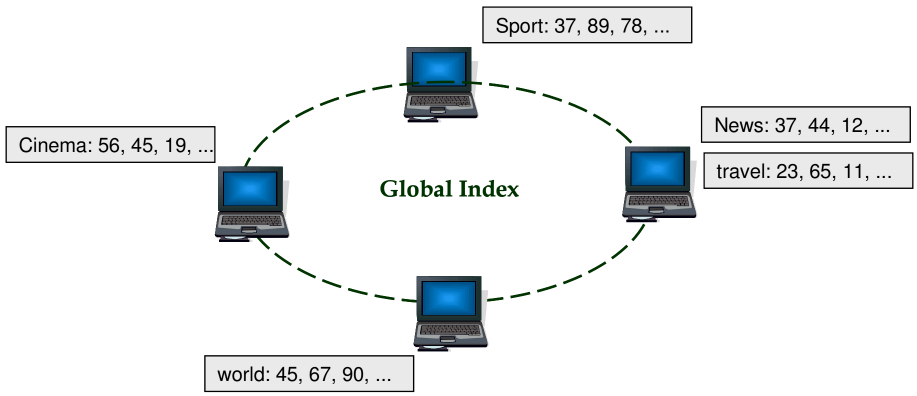

Figure 3: Global Index

Query processing for a multi-term query works as follows: first, the query is executed locally using the peer's repository index. If the result is considered unsatisfactory by the user, the querying peer retrieves a list of potentially useful peers by issuing a PeerList request for each query term to the global index. Using advanced forms of database selection methods from distributed IR and metasearch [Callan2000, Meng2002, Bender2005a, Bender2006b, Michel2006a, Michel2006b, Nottelmann2004, Nottelmann2006,], a number of promising peers for the complete query is computed from these PeerLists. This step is referred to as peer selection. Subsequently, the query is forwarded to these peers and executed based on their repository indexes. Note that this communication is done in a pairwise point-to-point manner between the peers, allowing efficient communication and limiting the load on the global directory. Finally, the results from the various peers are combined at the querying peer into a single result list; this step is referred to as result merging.

4.1 Advanced Query Routing

The query-routing decisions can use a variety of strategies based on statistical models; one of the best performing methods is CORI [Callan1995]. However, selecting the peers to which a query should be forwarded solely by CORI-like peer-quality measures does not consider the potential overlap among the target peers' contents. If there are several high-quality peers for a given term or topic that periodically mirror each other's content, they would all be ranked high for the given query and may end up being chosen as query targets. This wastes resources and is bound to achieve suboptimal recall. To overcome these problems that are typical for a highly diverse P2P network (and would not arise in this form in a federation of digital libraries or metasearch environment), we have developed means for estimating the overlap of two peers' contents. These estimates are then factored into the query routing decision by using a weighted combination of peer quality and overlap as ranking and decision criterion. The approach not only considers overlap between target peers and the local content of the query originator, but also among different target peers. An integrated quality-novelty selection algorithm chooses beneficial peers in an incremental manner for an overall execution plan with very good benefit/cost ratio [Bender2005a, Michel2006a]. Experiments with the Minerva system have shown major gains over CORI in terms of relative recall for a given number of target peers. Typically, the overlap-aware methods achieve 50 to 100 percent higher relative recall, or can achieve the same recall with half of the target peers or less thus saving precious network bandwidth.

Like CORI, the above new methods consider quality and overlap measures for individual terms and aggregate these precomputed measures for a given query. This ignores correlation among terms - a situation that is often unavoidable for tractability and usually accepted in IR, but becomes aggravated in P2P IR because of the coarser peer granularity of the directory. For example, for a query with two terms x and y, the query routing method considers the best peers for x and the best peers for y, and then identifies the best overall peers by combining these two rankings. But the best peer for the combination of x and y may actually be rather poor for each individual term and may not be considered at all. It turns out that the lack of correlation awareness is not a severe issue for highly correlated terms, because good peers for one term are then likely to have at least decent quality for the other term as well. But for uncorrelated or negatively correlated terms, the situation can be critical. Therefore, the solution that we developed performs on-demand distributed mining for uncorrelated or anti-correlated term sets, using ideas from frequent itemset mining. To make this scalable, the mining is restricted to interesting candidates which are terms that appear sufficiently often in query logs. Also, the distributed solution makes very clever use of the underlying DHT infrastructure and of hash sketches for probabilistic counting. This solution is both very elegant and of high practical value. Correlation-aware query routing [Bender2006b, Michel2006b] improves the benefit/cost ratio of other methods; it is orthogonal to the overlap awareness and can be combined with it. The gains from correlation awareness are significant.

5. INDEXING SOCIAL TAGS AND ATTRIBUTES

In the spirit of social tagging communities, users can provide annotations that describe their content objects. Currently, Minerva handles attribute-value-style annotations. For example, users might rate documents with annotations like “rating=5”. Additional annotations can automatically be generated from the XML files, such as “author=weikum” or “location=Germany”. These annotations are also indexed and become part of the global index, so that users can explicitly query for content objects with “rating=5” and even combine this with regular query terms. In addition to indexing the full text of the XML files, we index all the annotations using attribute-value style. For the XML description of flickr images shown in figure 1, we support the following XML tags as the most important ones for this setting (other settings require different tags, and the system supports arbitrary XML tags and attributes): Note that these tags are a subset of those defined in the SAPIR format [Kaplan2007].

- ***<tag></tag>***: Tags are annotation keywords that the user assigns to an image describing its content (e.g., as in Flickr). We consider tags as an important case in the indexing process as described later in this section. Note that in some cases tags are not carefully chosen by the user, thus, they might generate noisy information.
- ***<title></title>***: The title of an image is given by the user. It contains a high level description of the image. It does not always reflect the image content but can indicate the location of the image, the date when it was taken, or any other related information.
- **<description></description>**: This is a complementary information to the title and the tags provided by the user. It might contain a description about the content of the image, or any other related information such as the trip on which the picture was taken, related events, weather conditions, etc.

- **<comment></comment>**: This usually describes opinions of other users on the image.
- ***<owner></owner>*** : This tag contains information about the owner of the image, e.g., nickname or author, and location of the image owner
- ***<date></date>***: This is either the date when the picture was taken or the date when the picture was posted to the network.
- ***<location></location>: The indicates the location of where the image was taken (e.g., in terms of GPS coordinates or a location name).***

Terms under the tags described above are indexed using N-grams for tag-term combinations. We employ this technique for each XML tag content considering terms as the basic indexing unit. For example for the text “<description> Paris monument </description>”, the following 2-grams are generated: “<description>=Paris”, “<description>=monument”. Additionally, we consider the complete text as an N-gram for “<tag>” attributes. For example, the text “<tag> snow mountain </tag>” generates two 2-grams and one 3-gram: “<tag>=snow”, “<tag>=mountain”, and “<tag>=snow mountain”.

The benefit of using N-gram indexing is that we can handle XML Fragment queries of the form “<attribute>term</attribute>”. N-grams are considered as indexing keys and are stored in the Global Index.

6. INDEXING MULTIMEDIA CONTENT FEATURES

We propose two techniques for indexing Multimedia Content Features. The first technique consists of an inverted-list approach, and the second technique is the MRoute method [Gennaro2007] which supports similarity search on metric spaces based on the paradigm of a Routing Index (RI). An RI is a peer-granularity Local Index in the terminology of the SAPIR architecture [Kacimi2008].

6.1 TEXT-BASED APPROCH FOR CONTENT-BASED IMAGE RETREIVAL

Because of the computational costs of metric similarity search, it would actually be desirable to support image-content search based on a notion of inverted lists, in the spirit of information retrieval for text documents. To this end, various approaches have been pursued in the literature, including [Asm2007,Pm2002,Wilkins2005]. In the following, we focus on one of the simplest techniques from this line of research, namely, the method by [Wilkins2005], which is particularly light-weight and very easy to implement. The main goal of this approach is to search large image collections based on image-image similarity, without the need for clustering or calculation of a similarity matrix [Qmtmc2004]. This method uses the MPEG-7 XML descriptions to extract text-style features from images and provide retrieval facilities using these features. Two indexing approaches are considered to create an inverted index representation: a frequency-based approach and a positional-based approach. For experimentation with these approaches, we have implemented the method of [Wilkins2005] in the Minerva platform for retrieval with combined text and image-content features. In the following we provide implementation details of the method, and discuss some preliminary findings. It is planned to further study this family of methods in the coming year.

The first step in the creation of any inverted index is term identification for the current image being processed. If we look at data for the Color Structure feature, we can observe that the possible value for any term will be within the range of 0-255, providing only 256 unique terms and leading to poor selectivity of query filters. For Edge Histogram data this problem is even worse with the possible value for a given term being within the range of 0-8 providing only 9 unique terms. This means that the lexicon of unique terms is severely limited if we take terms as being integer values delineated by white space. To compare this to text, we would expect a similarly sized text index to have a lexicon ranging into the hundreds of thousands of unique terms. Therefore Wilkins et al. [Wilkins2005] create a greater number of unique terms. Their idea is to create what is called Term Grams (TGrams). TGrams are similar to N-grams since they are used to analyze the text and identify new terms. TGrams work on the token (word or term) level. A TGram is a new term that is the concatenation of adjacent terms to the current term being processed. The number of terms that will be concatenated is determined by the length of TGram. We give the following example of a snippet of a Color Structure feature representation:

10 8 0 0 128 10 0 0 255 166

To create TGrams, we first extract standard single terms, which would provide an identical list as above. If we want to create length-2 TGrams, they will result in the following terms:

10_8, 8_0, 0_0, 0_128, 128_10, 10_0, 0_0, 0_255, 255_166

It should be noted that the created TGrams are overlapping, and that an underscore character is inserted between the concatenated terms. The reason for this is that it is important to differentiate between the integer value `108' and the length-2 TGram `10_8'. TGrams with different lengths can be created, for example, TGrams of length 3 through 6, meaning that the earlier example would produce the following output:

Length 3:
10_8_0, 8_0_0, 0_0_128, 0_128_ 10, 128_10_0, 10_0_0, 0_0_255, 0_255_166

Length 4:

10_8_0_0, 8_0_0_128, 0_0_128_10, 0_128_10_0, 128_10_0_0, 10_0_0_255, 0_0_255_166

Length 5:

10_8_0_0_128, 8_0_0_128_10, 0_0_128_10_0, 0_128_10_0_0, 128_10_0_0_255, 10_0_0_255_166

Length 6:
10_8_0_0_128_10, 8_0_0_128_10_0, 0_0_128_10_0_0, 0_128_10_0_0_255, 128_10_0_0_255_166

This process creates more complex and varied terms, and adds a degree of size to the lexicon that improves selectivity of thus retrieval efficiency. However another increase in size can be generated if we take into account either positional or frequency information depending on index type. If the index is a frequency-based index, we take for each multimedia feature description its unique terms and count the number of occurrences of that term within the multimedia feature description. Taking our initial example, we obtain the frequency data shown in Table 2.

Term	Frequency
10_8	1
8_0	1
0_0	2
0_128	1

Table 2: Frequency-Based Index

This data is stored in the index for this multimedia feature description. To enlarge the lexicon size, new terms can be created as the aggregation of the TGram and the frequency data. We use a hyphen to distinguish between the two types of data so that the term is not confused as being a TGram of larger length (e.g. differentiate between the length-3 TGram 10_8_1 and the length-2 TGram 10_8-1 which occurs once). If we take the above example we would then have created the following new terms:

10 8-1, 8 0-1, 0 0-2, 0 128-1

A similar approach is taken with the positional index, except that instead of storing the frequency of the term, we store the position in which it occurs within the document. Again taking the earlier TGram of length 2, we would get the data as shown in Table 3.

Term	Position
10_8	0
8_0	1
0_0	2
0_128	3

Table 3: Position-Based Index

This would generate the following terms to be added to the lexicon:

10 8-0, 8 0-1, 0 0-2, 0 128-3

The TGrams describing the multimedia features of images are considered as standard terms and are indexed using a Minerva style approach as described in the previous section.

6.2 PEER-TO-PEER ROUTING INDEX APPROACH FOR MULIMEDIA SEARCH

6.2.1 Network Organization and Data Indexing

MRoute is based on an unstructured P2P network. Each peer P in the network maintains both an index of its local resources and a table for every neighbor, summarizing the objects that are reachable from it. The set of data objects owned by P, called local repository, is denoted with Data(P). 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. For each feature F of O, we maintain a set of m indices built using the metric distance of O^F from a set of m known reference points. The exploitation of distances from predefined reference points and the use of the triangle inequality are well-known methods to evaluate the placement of an object in a metric space. This information is then used at query-resolution time in order to prune uninteresting objects without the need of calculating the actual distance of the query from the objects.

In our structure we exploit reference points' distances for building Routing Indices (RIs), a particular form of peer-granularity Local Indexes which together constitute the Global Index. An RI consists of a vector with k elements. Each element of the vector represents a distance interval from a reference object, as illustrated in the example of Figure 4.

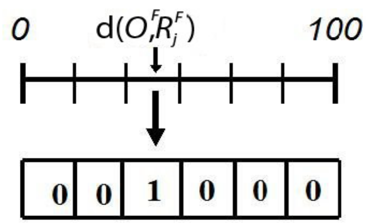

Figure 4: Example of Reference Object in Routing Index

As can be seen, the vector is filled in with 0s, except for the entry corresponding to the distance interval where the real distance of the object from the reference object lies. Thus, if a feature *F* has an associated set of reference points $R^F = \{R_1^F, \cdots, R_m^F\}$, and given $k+1$ division points $a_0 < \cdots < a_k$, the index of O^F wrt a generic reference point R_j^F, is a vector

$$DataIdx(O^F)_{R_j^F} = (b_0, \cdots, b_{k-1})$$

where $b_i \neq 0$ iff $d\left(O^F, R_j^F\right) \in [a_i, a_{i+1})$.

P exploits these indices in order to obtain a cumulative index for *F*. It represents the histogram of the distribution of all its object values into the division intervals. It is obtained as a sum of the single data indices, and defined as:

$$NodeIdx(P, F)_{R_j^F} = \sum_{O \in Data(P)} DataIdx(O^F)_{R_j^F}$$

This index is related to a single reference point of feature *F*. The complete representation of *F* is given by the set of *m* indices related to its set of reference points R^F.

In order to allow the sharing of information among peers, each single peer index is used to build routing indices. Let us denote with *Nb (P)* the set of other peers directly connected to *P*. Given $P_i \subset N(P)$, *P* maintains information about the values of the features of the resources that can be found by following the link $P \to P_i$. This information can then be used at query-resolution time to decide whether to route the query to P_i (since there are potential matchings) or not. For each reference point R_j^F of a feature *F*, the index is calculated recursively in the following way:

$$LinkIdx(P \to P_i, F)_{R_j^F} \equiv NodeIdx(P_i, F)_{R_j^F} + \sum_{p \in Nb(P_i) - P} LinkIdx(P_i \to p, F)_{R_j^F}$$

Since the network is unstructured, it may contain loops. The presence of loops is a well-known problem for Routing Indices, since they may led to the duplication of the information contained in an index. To avoid this problem, we use a node caching mechanism to avoid repeated updates of our indices. As a consequence, an update related to a given node is associated with only one path. 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 $P \to P_i$ represents the value of all the objects placed in a *logical* sub-tree rooted at P_i.

6.2.2 Query Processing

RIs are exploited to solve and route queries in the system. A query is defined as a set of sub-queries over a sub-set of the features defined for the system's objects. Given a feature *F*, let Q^F be the sub-query related to *F* and *r* its radius. Q^F is mapped in the index space in a similar way as for data objects. Thus, given a reference point R_j^F, there is an index $QueryIdx(Q^F)_{R_j^F} \equiv (q_1, \cdots, q_k)$, where $q_i = 1$ if $[a_i, a_{i+1}) \cap \left[d\left(Q^F, R_j^F\right) - r, d\left(Q^F, R_j^F\right) + r\right] \neq \emptyset$. The index can then be used to check whether the logical sub-tree rooted on a peer contains potential matchings or not. The process can be repeated to also see if a given peer possibly stores matches for the query. This could be done by comparing the query index and an RI or a peer index, respectively. Given a query index $QueryIdx(Q^F)_{R_j^F} \equiv (q_1, \cdots, q_k)$ and an RI $LinkIdx(P \to P_i, F)_{R_j^F} = (l_1, \cdots, l_k)$, there exist potential matchings if $\exists i: q_i \neq 0 \wedge l_i \neq 0$. The procedure is similar for a single peer's repository index. In order to have a match for the query Q as a whole, all the indices of all the features must match with the corresponding RIs (or the peers' repository indices). In this way, a query is forwarded only toward zones where potentially matching objects could be found. Since indices are an approximate representation of the objects values, a positive index matching does not necessarily mean that the corresponding network zone really contains matches for the query. But in any case, uninteresting peers or even uninteresting zones of the network are excluded from the query forwarding process, thus greatly reducing the number of involved peers.

7. OUTLOOK

In this report we have presented the SAPIR approach to indexing in peer-centric overlay networks. Essentially, our mechanism for indexing is a Peer-granularity Global Index, and we use intelligent query routing for search. The developed techniques can be applied to the full spectrum of features, ranging from text and tags to audio-visual features such as colors, texture, and contours. In the second year of SAPIR we will further refine and extend this approach, and we will study the scalability and performance properties of our methods.

8. REFERENCES

[Agrawal2003]	A. Agrawal and H. Casanova. Clustering hosts in p2p and global computing platforms. International Symposium on Cluster Computing and the Grid (CCGRID), 2003.
[Asm2007]	G. Amato, P. Savino, V. Magionami. Image Indexing and Retrieval Using Visual Terms and Text-Like Weighting. DELOS Conference 2007: 11-21
[Bender2005a]	M. Bender, S. Michel, P. Triantafillou, G. Weikum, C. Zimmer: Improving Collection Selection with Overlap Awareness in P2P Search Engines. International ACM SIGIR Conference, 2005.

[Bender2005b] M. Bender, S. Michel, P. Triantafillou, and G. Weikum. Minerva: Collaborative p2p search. International Conference on Very Large Data Bases (VLDB), 2005

[Bender2006a] M. Bender, S. Michel, C. Zimmer, P. Triantafillou, G. Weikum.: P2P Content Search: Give the Web Back to the People. International Workshop on Peer-to-Peer Systems (IPTPS), 2006.

[Bender2006b] M. Bender, S. Michel, G. Weikum: P2P Directories for Distributed Web Search: From Each According to His Ability, to Each According to His Needs. International Workshop on Web IR and Integration (WIRI), 2006.

[Bender2007] M. Bender, S. Michel, J. Xavier Parreira, T. Crecelius.: P2P Search: Make It Light, Make It Fly, International Conference on Innovative Data System Research (CIDR), 2007.

[Callan1995] J. Lu, J. Callan: Federated Search of Text-Based Digital Libraries in Hierarchical Peer-to-Peer Networks. European Conference on Information Retrieval (ECIR), 2005.

[Callan2000] J. Callan. Distributed Information Retrieval. Advances in information retrieval, Kluwer Academic Publishers, 2000.

[Callan2003] J. Lu, J. P. Callan:. Content-based retrieval in hybrid peer-to-peer networks. International Conference on Information and Knowledge Management (CIKM), 2003.

[Chawathe2003] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making gnutella-like p2p systems scalable. International ACM SIGCOMM Conference, 2003.

[Crespo2002] A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. International Conference on Distributed Computing Systems (ICDCS), 2002.

[Crespo2004] A. Crespo, H. Garcia-Molina: Semantic Overlay Networks for P2P Systems. International Workshop on Agents and Peer-to-Peer Computing (AP2PC), 2004.

[Cuenca-Acuna2003] F. M. Cuenca-Acuna, C. Peery, R. P. Martin, T. D. Nguyen: PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities. The International Symposium on High Performance Distributed Computing (HPDC), 2003.

[Ganesan2003] P. Ganesan, Q. Sun, and H. Garcia-Molina. Yappers: A peer-to-peer lookup service over arbitrary topology. IEEE Conference on Computer Communications (INFOCOM), 2003.

[Gennaro2007] C. Gennaro, M. Mordacchini, S. Orlando, F. Rabitti. MRoute: A Peer-to-Peer Routing Index for Similarity Search in Metric Spaces. International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P), 2007.

[Gnutella2003] Gnutella. http://www.gnutella.com, 2003.

[Gravano1999] L. Gravano, H. Garcia-Molina, A. Tomasic. GlOSS: Text-Source Discovery over the Internet. ACM Transaction on Database Systems, 1999.

[Gupta2004] A. Gupta, O. D. Sahin, D. Agrawal, and A. El Abbadi. Meghdoot: Content-based publish/subscribe over p2p networks. ACM/IFIP/USENIX International conference on Middleware, 2004.

[Hang2002] C. Hang Ng and K. Cheung Sia. Peer clustering and network query model. World Wide Web Conference (WWW), 2002.

[Kacimi2008] M. Kacimi, Y. Mass, M. Shmueli-Scheuer, G. Weikum. Deliverable D2.1: Defining the Main Components of the SAPIR Software, their Functionality, and their Interfaces. January 2008

[Kalnis2006] P. Kalnis, W. Siong Ng, B. Chin Ooi, K. L. Tan. Answering similarity queries in peer-to-peer networks. Information System, Elsevier, 2006.

[Kaplan2007] A. Kaplan, C. Hagège, F. Falchi, W. Allasia, F. Gallo, J. Mamou, Y. Mass, R. Miotto, N. Orio. SAPIR Deliverable D3.1: Common Schema for Feature Extraction, 2007.

[Loeser2003] A. Loeser, W. Nejdl, M. Wolpers, and W. Siberski. Information integration in schema-based peer-to-peer networks. International Conference On Advanced Information Systems Engineering (CAISE), 2003.

[Loeser2005] A. Loeser, C. Tempich, B. Quilitz, W. T. Balke, S. Staab, W> Nejdl: Searching Dynamic Communities with Personal Indexes. International Semantic Web Conference 2005: 491-505

[Losee2004] R. M. Losee and L. C. Jr. Information retrieval with distributed databases: Analytic models of performance. IEEE Transactions On Parallel and Distributed Systems, 15(1):18 -27, January 2004.

[Manning2007] C. Manning, P. Raghavan, H. Schütze: An Introduction to Information Retrieval, Cambridge University Press, 2007

[Marti2004] S. Marti, P. Ganesan, and H. Garcia-Molina. Sprout: P2P routing with social networks. International Conference on Extending Database Technology (EDBT), 2004.

[Meng2002] W. Meng, Clement T. Yu, King-L. Liu. Building efficient and effective metasearch engines. ACM Computing Surveys. 34(1), 2002

[Michel2006a] S. Michel, M. Bender, P. Triantafillou, G. Weikum. IQN Routing: Integrating Quality and Novelty for Web Search. International Conference on Extending Database Technology (EDBT), 2006

[Michel2006b] S. Michel, M. Bender, N. Ntarmos, P. Triantafillou, G. Weikum, C. Zimmer. Discovering and Exploiting Keyword and Attribute-Value Co-Occurrences to Improve P2P Routing Indices. International Conference on Information and Knowledge Management (CIKM), 2006.

[Mislove2006] A. Mislove, K. P. Gummadi, P. Druschel. Exploiting Social Networks for Internet Search. International Workshop on Hot Topics in Networks (HotNets), 2006

[Mueller2005] W. Mueller, M. Eisenhardt, A. Henrich. Scalable summary based retrieval in P2P networks. International Conference on Information and Knowledge Management (CIKM), 2005.

[Nejdl2003] W. Nejdl, M. Wolpers, W. Siberski, A. Loser, I. Bruckhorst, M. Schlosser, and C. Schmitz. Semantic overlay clusters within super-peer networks. International World Wide Web Conference (WWW), 2003.

[Nottelmann2004] H. Nottelmann, N. Fuhr. Combining CORI and the Decision-Theoretic Approach for Advanced Resource Selection. European Conference on Information Retrieval (ECIR). 2004.

[Nottelmann2006] H. Nottelmann, N. Fuhr. Comparing Different Architectures for Query Routing in Peer-to-Peer Networks. European Conference on Information Retrieval (ECIR), 2006.

[Parreira2007] J. Xavier Parreira, S. Michel, G. Weikum. p2pDating: Real Life Inspired Semantic Overlay Networks for Web Search. Journal of Information Processing and Management, Special Issue on Heterogeneous Distributed Systems, 43(3): 643-664, 2007.

[Pm2002] T. Pfund, S. Marchand-Maillet. A Dynamic Multimedia Annotation Tool. In Proccedings of {SPIE} Photonics West, Electronic Imaging 2002, Internet Imaging {III}, San Jose, USA, 2002.

[Podnar2007] I. Podnar, M. Rajman, T. Luu, F. Klemm, K. Aberer: Scalable Peer-to-Peer Web Retrieval with Highly Discriminative Keys. International Conference on Data Engineering (ICDE), 2007.

[Qmtmc2004] T. Quack, U. MÄonich, L. Thiele, and B. Manjunath. Cortina: A system for large-scale, content-based web image retrieval. In Proceedings of ACM Multimedia, 2004.

[Raftopoulou2008] P. Raftopoulou, E. G.M. Petrakis. iCluster: a Self-Organising Overlay Network for P2P Information Retrieval, European Conference on Information Retrieval (ECIR) Conference, 2008.

[Ramabhadran2004] S. Ramabhadran, S. Ratnasamy, J. M. Hellerstein, and S. Shenker. Brief announcement: prefix hash tree. International Symposium on Principles of Distributed (PODC), 2004.

[Ratsnasamy2002] Q. Lv, S. Ratsnasamy, and S. Shenker. Can heterogeneity make gnutella scalable? International Workshop on Peer-to-Peer Systems (IPTPS), 2002.

[Rowstron2001] A. I. T. Rowstron, P. Druschel.: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. Middleware, 2001.

[Sripanidkulchai2003] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems citation. IEEE Conference on Computer Communications (INFOCOM), 2003.

[Steinmetz2005] R. Steinmetz, K. Wehrle. Peer-to-Peer Systems and Applications. Springer, 2005.

[Tang2003] C. Tang, Z. Xu, S. Dwarkadas. Peer-to-peer information retrieval using self-organizing semantic overlay networks. International ACM SIGCOMM Conference, 2003.

[Wang2004] H. J. Wang, Y. Hu, C. Yuan, Z. Zhang, and Y. Wang. Friends troubleshooting network: Towards privacy-preserving, automatic troubleshooting. International Workshop on Peer-to-Peer Systems (IPTPS), 2004.

[Wilkins2005] P. Wilkins, P. Ferguson, E.F Smeaton, C. Gurrin. Text Based Approaches for Content-Based Image Retrieval on Large Image Collections. European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies (EWIMT), 2005.

[Zobel2006] J. Zobel, A. Moffat.: Inverted files for text search engines. ACM Computing Surveys. 38(2): 2006.