# Bridging the P2P and WWW Divide with DISCOVIR - DIStributed COntent-based Visual Information Retrieval

Ka Cheung Sia
kcsia@cse.cuhk.edu.hk

Cheuk Hang Ng
chng@cse.cuhk.edu.hk

Chi Hang Chan
chchan@cse.cuhk.edu.hk

Siu Kong Chan
skchan3@cse.cuhk.edu.hk

Lai Yin Ho
lyho@cse.cuhk.edu.hk

Irwin King
king@cse.cuhk.edu.hk

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong SAR

## ABSTRACT

In the light of image retrieval evolving from text annotation to content-based and from standalone applications to web-based search engines, we foresee the need for deploying content-based image retrieval (CBIR) into Peer-to-Peer (P2P) architecture. By doing so, we not only distribute the tasks of feature extraction, indexing and storage of image data into peers, we also introduce another aspect of searching in addition to filename-based method in prevalent P2P applications. Through the deployment of DISCOVIR, we introduce a model to improve query efficiency targeting on content-based search.

Current P2P applications require installing special purpose software and proprietary protocols for information retrieval, which limit the number of audience. To make use of the WWW to increase popularity of P2P, we propose *DISCOVIR Everywhere* to bridge the two different worlds, P2P and WWW. We outline the process of accessing DISCOVIR network through web browsers or mobile devices, by the coordination of a light weighted gateway, with reduced workload compared to existing methodology.

## Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software; H.4.3 [**Information Systems Applications**]: Communications Applications

## General Terms

Design

## Keywords

peer-to-peer; content-based image retrieval; web integration; mobility

## 1. INTRODUCTION

The appearance of Peer-to-Peer (P2P) applications in recent years have demonstrated the significance of distributed information sharing systems by offering the advantages of scalable storage space. Regarding to current content-based image retrieval (CBIR) systems, we envisage the potential use of P2P network in both scattering data storage and distributing workload of feature extraction and indexing. Through the realization of CBIR in P2P network, we can support enormous image collection without installing high-end database and hardware for the web server. Also, we make use of the computation power of peers in addition to data storage.

On the other hand, we revolutionize the way of searching in current P2P architecture. The queries in P2P are mostly based on text, entered by users to describe shared files. The accuracy of retrieval depends mainly on whether users can come up with a common description on a file. CBIR raise another aspect of searching in P2P other than filename-based search. We also formulate a query model that improves query efficiency under the content-based search architecture and reveal the research need for improving query efficiency. Moreover, to improve the availability of P2P, we need to bridge the P2P and WWW for P2P to grow and gain popularity.

To address the desirabilities of P2P application raised above, we introduce the DIStributed COntent-based Visual Information Retrieval (DISCOVIR) [5] system and *DISCOVIR Everywhere* with the following characteristics:

1. DISCOVIR extends current CBIR system into P2P fashion and achieve the utilization of both data storage and computation resource at the same time.
2. Queries in DISCOVIR are no longer based on filenames but on the content of images. The need for annotating shared files is waived, thus, query accuracy does not depend on subjective perception of keywords.
3. *DISCOVIR Everywhere* provides a gateway for web and mobile users to access the DISCOVIR network. This architecture get rid of problems exist in current web-based P2P service by tightly integrating the web and P2P.

*WWW2003*, May 20–24, 2003, Budapest, Hungary.
ACM xxx.

In the next section, we will detail the related works in CBIR and P2P, and justify the motivation and necessity of combining them. In Section 3, we introduce the architecture of DISCOVIR and the functionality of its components. In Section 4, *DISCOVIR Everywhere* is introduced as an extension to DISCOVIR. We describe details of current implementations and future works in Section 5. Lastly, we give concluding remarks in Section 6.

## 2. RELATED WORKS

In early image retrieval system, it requires human annotation and classification on the image collection, the query is thus performed using text-based information retrieval method. However, there are several limitations for such implementation, they are:

- **Human Intervention** - Human intervention is required to describe and annotate the content of images, which is tedious and potentially error-prone.
- **Non-Standard Description** - As the size of image database grows, limited keywords results in inadequacy for describing the image content. Moreover, the keywords used are subjective and not unique. Different users may use different keywords to annotate the same image.
- **Linguistic Barriers** - If the image database is to be shared globally around the world, the retrieval of images will be ineffective when different languages are used in the description. It is difficult to map semantically equivalent words across different languages.

In order to solve these problems, CBIR is proposed to pass such tedious task to computer. Since early 1990's, many CBIR systems have been proposed and developed, some of them are QBIC [6], WebSEEK [28], SIMPLIcity [30], MARS [17], Photobook [22], WALRUS [20] and other systems for some domain specific applications [13, 12]. These systems are not designed to be distributed across different computers in a network. One of the shortcomings is that the feature extraction, indexing and also the query processing are all done in a centralized fashion which is computationally intensive and difficult to scale up. As indicated by several researchers [24, 27], one of the promising future trends in CBIR includes the distribution of data collection, data processing and information retrieval. By extending the centralized system model, we not only can increase the size of image collections easily, but we also overcome the scalability bottleneck problem by distributing the process of feature extraction and retrieval.

P2P network is a recently evolved paradigm for distributed computing. With the emerging P2P networks and various implementations such as Gnutella [8], Napster [19], LimeWire [14], YouServ [3], Freenet [7], Morpheus [18] and KaZaA [11], they focus on the retrieval of data based on filenames or metadata, such as ID3 tag of MP3, and offer the advantages of distributed resource [25], increased reliability [4] and comprehensiveness of information [15]. Figure 1 shows an example of a typical P2P network. In the example, different files are shared by different peers. When a peer initiates a search for a file, it broadcasts a query request to all its connecting peers. Its peers then propagate the request to their connected peers and this process continues. Unlike the client-server architecture of the web, the P2P network aims at allowing individual computer, which joins and leaves the network frequently, to share information directly with each other without the help of dedicated servers. In these networks, a peer can become a member of the network by establishing a connection with one or more peers in the current network. Messages are sent over multiple hops from one peer to another while each peer responds to queries for information it shares locally. Meanwhile, some current researches [29, 23] focus on improving query efficiency on filename-based search and we address this problem in the notion of content-based search [21, 26] in earlier works.

The motivation of proposing DISCOVIR is to migrate traditional CBIR to a P2P network as a step to introduce content-based search in P2P. With the advantages of P2P network, we utilize not only the distributed data storage, but also the computation power of each peer for the preprocessing and indexing of images. In order to improve the accessibility of P2P network, we further elaborate on current web-based P2P services and propose *DISCOVIR Everywhere* to provide web interface for users to carry out CBIR in P2P network.

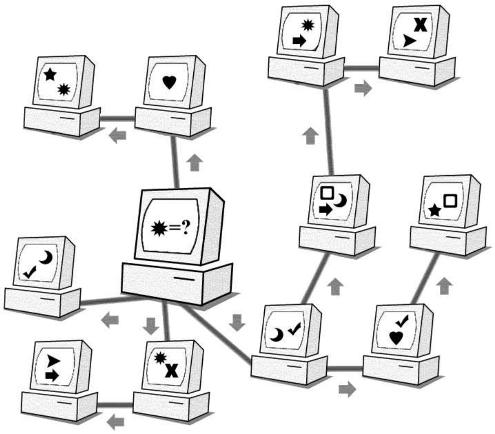


**Figure 1: P2P information retrieval**

## 3. ARCHITECTURE OF DISCOVIR

In this section, we will describe the architecture of a DISCOVIR client and the communication protocol in order to perform CBIR over a P2P network. Through the DISCOVIR program, users can share images among peers around the world. Each peer is responsible for extracting and indexing the feature of the shared images, by doing so, every peers can search for similar images based on image content, like color, texture and shape among images shared by all DISCOVIR peer in the network.

Figure 2 depicts the key components and their interaction in the architecture of a DISCOVIR client. As DISCOVIR is derived from LimeWire [14] open source project, the operations of Connection Manager, Packet Router and HTTP Agent remain more or less the same with additional functionality to improve the query mechanism used in original Gnutella network. Plug-in Manager, Feature Extractor and Image Indexer are introduced to support the CBIR task.

The User Interface is modified to incorporate the image search panel, Figure 3 shows a screen capture of DISCOVIR in the image search page. Here are the brief descriptions of each component:

- **Connection Manager** - It is responsible for setting up and managing the TCP connection between DISCOVIR clients.
- **Packet Router** - It masters the routing of message in DISCOVIR network between components and peers.
- **Plug-in Manager** - It coordinates the storage of different feature extraction modules and their interaction with Feature Extractor and Image Indexer.
- **HTTP Agent** - It is a tiny web-server that handles file download request from other DISCOVIR peers using HTTP protocol.
- **Feature Extractor** - It collaborates with the Plug-in Manager to perform various feature extraction and thumbnail generation of the shared image collection.
- **Image Indexer** - It indexes the image collection by content feature and carry out clustering to speed up the retrieval of images.
- **Bootstrap Server** - It maintains an update list of currently available DISCOVIR peers.

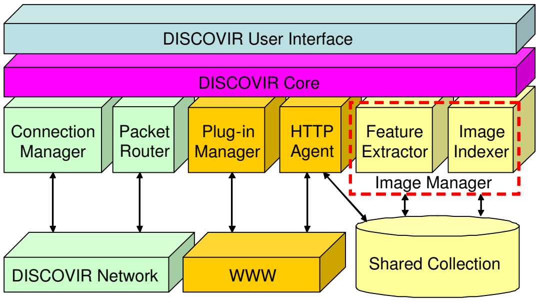
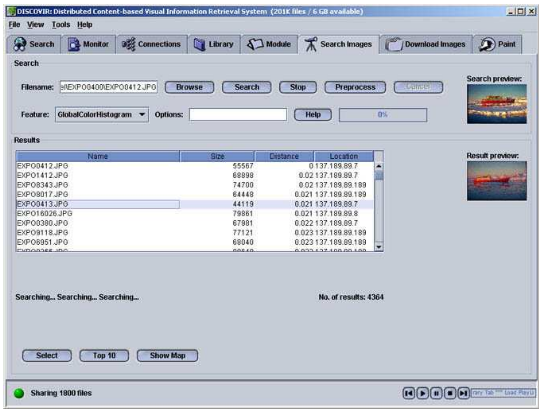


**Figure 2: Architecture of DISCOVIR**



DISCOVIR: Distributed Content-based Visual Information Retrieval System (201K files / 6 GB available)

Search

Filename: e\EXPO0400\EXPO0412.JPG

Feature: GlobalColorHistogram

Options:

0%

Results

| Name | Size | Distance | Location |
|---|---|---|---|
| EXPO0412.JPG | 55567 | 0 | 137.189.89.7 |
| EXPO1412.JPG | 68898 | 0.02 | 137.189.89.7 |
| EXPO8343.JPG | 74700 | 0.02 | 137.189.89.189 |
| EXPO8017.JPG | 64448 | 0.021 | 137.189.89.189 |
| EXPO0413.JPG | 44119 | 0.021 | 137.189.89.7 |
| EXPO16026.JPG | 79861 | 0.021 | 137.189.89.8 |
| EXPO0380.JPG | 67981 | 0.022 | 137.189.89.7 |
| EXPO9118.JPG | 77121 | 0.023 | 137.189.89.189 |
| EXPO6951.JPG | 68040 | 0.023 | 137.189.89.189 |

Searching... Searching... Searching...

No. of results: 4364

Sharing 1800 files

**Figure 3: Screen Capture of DISCOVIR**

## 3.1 Flow of Operations and Functionality of DISCOVIR Components

The following is a scenario walk-through to demonstrate the interaction between components. When a user chooses to preprocess his image collection, the *Feature Extractor* collaborates with *Plug-in Manager* to extract content feature and generate thumbnails from all images in the shared directory. The result is passed to *Image Indexer* for indexing and clustering purposes. In case when a new DISCOVIR client wants to join the network, the *Connection Manager* asks a *Bootstrap server* for a list of currently available DISCOVIR clients in order to hook up to the network. Once a user initiates a query for similar images, the *Feature Extractor* extracts feature from the example image instantly, *Packet Router* is responsible for assembling an ImageQuery message and sending out to the DISCOVIR network. For instance, when an ImageQuery message is received from other peers, the *Packet Router* checks for any duplication and propagates to other peers through DISCOVIR network. Meanwhile, it passes the message to *Image Indexer* for searching similar images. Upon similar images are found, an ImageQueryHit message is assembled and passed to *Packet Router* for replying the initiating peer. When ImageQueryHit messages return to the initiating peer, its *HTTP Agent* downloads thumbnails or full size images from other peers upon receiving user request from the user interface.

### *3.1.1 Preprocessing*

Plug-in Manager is responsible for contacting web-site of DISCOVIR to inquire the list of available feature extraction modules, it will download and install selected modules upon user request. Currently, DISCOVIR supports various feature extraction methods in color and texture categories such as AverageRGB, GlobalColorHistogram, ColorMoment, Co-occurrence matrix, etc [10]. All feature extraction modules strictly follow a predefined interface in order to realize the polymorphic properties of switching between different plug-ins dynamically, see Fig. 7.

Feature Extractor will extract feature and generate thumbnails for all images in the shared collection by using a particular feature extraction module. Let $I$ represent a raw image data, $f$ be the feature extraction method, the feature extractor perform the task illustrated in Eq. 1,

$$f : I \times \theta \rightarrow \vec{I} \tag{1}$$

where $\theta$ is the feature extraction parameter and $\vec{I}$ is the extracted feature vector. Image Indexer will then index the image collection using the multi-dimensional feature vectors $\vec{I}$ in order to answer an incoming query. It also clusters the set of feature vector for the sake of improving query efficiency by acquiring statistical distribution information of the local image collection [26].

Compared with the centralized web-based CBIR approach, sharing the workload of this computational costly task among peers by allowing them to store and index their own image collection helps solving the bottle-neck problem by utilizing distributed computing resources.

### *3.1.2 Connection Establishment*

For a peer to join the DISCOVIR network, it connects to the Bootstrap Server using the Connection Manager. The Bootstrap Server is responsible for storing a finite list of IP address of peers currently in the DISCOVIR network and randomly picks an IP address to return to the peer. Once the IP address is received, the peer is able to hook up to the DISCOVIR network by connecting to the selected peer. In order to provide the Bootstrap Server with update list of IP address, the Connection Manager of each peer sends an alive message to the bootstrap server periodically after it has connected to the DISCOVIR network.

### *3.1.3 Query Message Routing*

After a peer joins the DISCOVIR network, it may initiate a query for similar images. The Feature Extractor processes the query image instantly and assemble an ImageQuery message to be sent out through Packet Router. Likewise, when other peers receive the ImageQuery messages, they need to perform two operations, *Query Message Propagation* and *Local Index Look Up*.

- **Query Message Propagation** - In order to prevent query messages from looping forever in the DISCOVIR network, two mechanisms are inherited from Gnutella, namely, the Gnutella replicated message checking rule and Time-To-Live (TTL) of messages. The replicated message checking rule can prevent a peer from propagating the same query message again. The TTL mechanism constrains the horizon of a query message able to reach. After these two checkings, the query message will be propagated to linked peers through Packet Router.

- **Local Index Look Up** - The peer searches its local index of shared files for similar images using the Image Indexer. Once similar images are retrieved, the peer delivers an ImageQueryHit message back to the requester through Packet Router. Since images indexing is performed on the peer in preprocessing stage, the searching time can be speeded up.

### *3.1.4 Query Result Display*

When an ImageQueryHit message returns to the requester, user will obtain a list detailing the location and size of matched images. In order to retrieve the query result, the HTTP Agent will download thumbnails or full size image from the peer using HTTP protocol. On the other hand, HTTP Agent in other peers will serve as a web server to deliver the requested images. This HTTP Agent is essential for integration to WWW which will be described later in detail.

## 3.2 Query Efficiency

To address the efficiency problem in query messages routing, we propose the use of Firework Query Model (FQM) [21, 26]. In FQM, peers sharing similar content will be clustered together like a Yellow pages. When a peer initiates or receives a query message, the query is routed selectively according to the content of the query. Once it reaches its designated cluster, the query message is broadcasted by peers inside the cluster much like an exploding firework as shown in Fig. 4. This strategy aims to minimize the number of messages passing through the network, reduce the workload of each computer and maximize the ability of retrieving relevant data from the P2P network.

### *3.2.1 Peer Clustering*

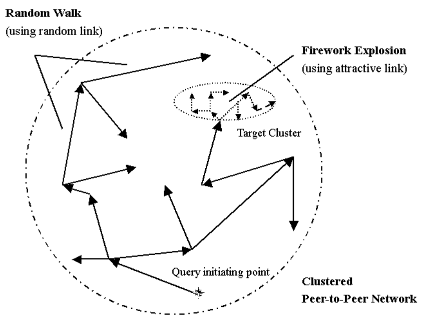


**Figure 4: Illustration of firework query.**

In order to help building the DISCOVIR as a self-organized network oriented in content affinity, we perform peer clustering at the level of overlaying network topology. The peer clustering algorithm introduced in FQM consists of three steps:

1. **Signature Value Calculation**–Every peer preprocess its data collection as described in Section *3.1.1* and divide the whole data set into sub-clusters by a clustering algorithm, e.g. $k$-means, competitive learning, and expectation maximization. A signature value $sig_v$ will be calculated for each sub-cluster to characterize its data properties. Since the selection of clustering algorithm and its parameters can result in different number of sub-clusters. The number of signature values is variable and is a trade-off between data characteristic resolution and computational cost.

2. **Neighborhood Discovery**–After a peer joined the DISCOVIR network with the procedure described in Section *3.1.2*, it broadcasts a signature query message, similar to that of ping-pong messages in Gnutella, to reveal the location and data characteristic of its neighborhood. This task is not only done when a peer first joins the network, it repeats every certain interval in order to maintain the latest information of other peers.

3. **Attractive Connection Establishment**–By acquiring the signature values of other peers, one can reveal the peer with highest data affinity (similarity) to itself, and make an attractive connection to link them up. When an existing attractive connection breaks, a peer should check its host cache, which contains signature values of other peers found in the neighborhood discovery stage, and reestablish the attractive connection using peer clustering algorithm again.

Having all peers joining the DISCOVIR network perform the three tasks described above, one can envision a P2P network with self-organizing ability to be constructed. The detail steps of peer clustering is illustrated in Algorithm 1.

### *3.2.2 Selective Query Message Routing*

The main goal of query message routing algorithm in P2P network is to minimize the number of message passing, while maximize the ability of retrieving relevant data. Here, we introduce the algorithm used in FQM to determine when and how a query message is propagated.

The query message routing algorithm used in FQM is different from the one described in Section *3.1.3* in how to

**Algorithm 1** Algorithm for peer clustering

```
Peer-Clustering(peer v, integer ttl)
for all signature value $sig_v$ in peer $v$ do
  for all peer $w$ in discovered neighbors do
    for all signature value $sig_w$ in peer $w$ do
      Compute $Distance(sig_v, sig_w)$
    end for
  end for
  establish attractive connection between $v$ and $w$ having $min(Distance(sig_v, sig_w))$
end for
```

select a peer for the query message to route to. In FQM, the distance between the signature value of a peer and the query, $Distance(sig_v, q)$, is measured. If it is smaller than a preset threshold, $\theta$, the peer will propagate the query to its neighbors through normal connections. Otherwise, if one or more $Distance(sig_v, q)$ is within the threshold, it implies the query has reached its target cluster. Therefore, the query will be propagated through corresponding attractive connections much like an exploding firework. The detail steps of the algorithm is illustrated in Algorithm 2.

**Algorithm 2** Algorithm for the Firework Query Model

```
Firework-query-routing (peer v, query q)
for all signature value $sig_v$ in peer $v$ do
  if $Distance(sig_v, q) < \theta$ (threshold) then
    if $q_{ttl} > 0$ then
      propagate $q$ using attractive links
    end if
  end if
end for
if Not forwarding to attractive link then
  if $q_{ttl} > 0$ then
    forward $q$ using normal links
  end if
end if
$q_{ttl} = q_{ttl} - 1$
```

## 3.3 Gnutella Message Modification

The DISCOVIR system is built compatible to the Gnutella network. In order to support the image query functionalities mentioned above, two types of messages are added. They are:

- **ImageQuery** - Special type of the Query message. It is to carry name of feature extraction method and feature vector of query image, see Fig. 5
- **ImageQueryHit** - Special type of the QueryHit message. It is to respond to the ImageQuery message, it contains the location, filename and size of similar images retrieved, and their similarity measure to the query. Besides, the location information of corresponding thumbnails are added for the purpose of previewing result set in a faster speed, see Fig. 6

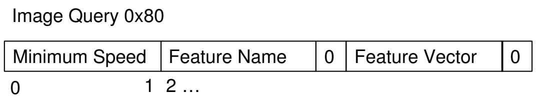


Image Query 0x80

| Minimum Speed | Feature Name | 0 | Feature Vector | 0 |
|---|---|---|---|---|
| 0 1 | 2 ... | | | |

**Figure 5: ImageQuery message format**

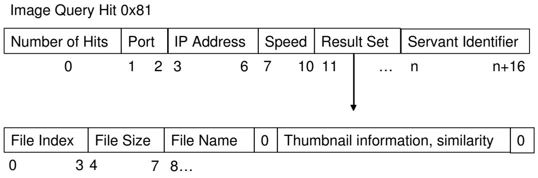
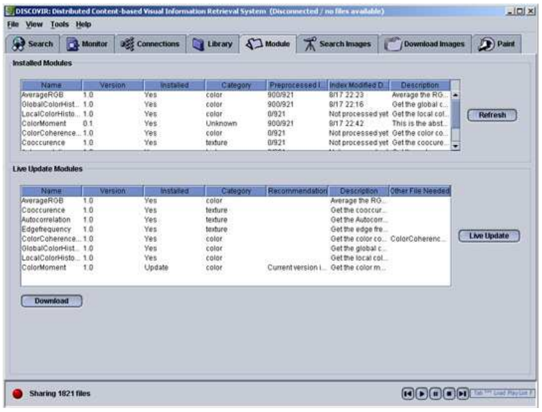


Image Query Hit 0x81

| Number of Hits | Port | IP Address | Speed | Result Set | Servant Identifier |
|---|---|---|---|---|---|
| 0 | 1 2 | 3 6 | 7 10 | 11 ... | n n+16 |

| File Index | File Size | File Name | 0 | Thumbnail information, similarity | 0 |
|---|---|---|---|---|---|
| 0 3 | 4 7 | 8... | | | |

**Figure 6: ImageQueryHit message format**



DISCOVIR: Distributed Content-based Visual Information Retrieval System (Disconnected / no files available)

Installed Modules

| Name | Version | Installed | Category | Preprocessed I... | Index Modified D... | Description |
|---|---|---|---|---|---|---|
| AverageRGB | 1.0 | Yes | color | 900/921 | 8/17 22:23 | Average the RG... |
| GlobalColorHist... | 1.0 | Yes | color | 900/921 | 8/17 22:16 | Get the global c... |
| LocalColorHisto... | 1.0 | Yes | color | 0/921 | Not processed yet | Get the local col... |
| ColorMoment | 0.1 | Yes | Unknown | 900/921 | 8/17 22:42 | This is the abst... |
| ColorCoherence... | 1.0 | Yes | color | 0/921 | Not processed yet | Get the color co... |
| Cooccurence | 1.0 | Yes | texture | 0/921 | Not processed yet | Get the coocure... |

Live Update Modules

| Name | Version | Installed | Category | Recommendation | Description | Other File Needed |
|---|---|---|---|---|---|---|
| AverageRGB | 1.0 | Yes | color | | Average the RG... | |
| Cooccurence | 1.0 | Yes | texture | | Get the cooccur... | |
| Autocorrelation | 1.0 | Yes | texture | | Get the Autocorr... | |
| Edgefrequency | 1.0 | Yes | texture | | Get the edge fre... | |
| ColorCoherence... | 1.0 | Yes | color | | Get the color co... | ColorCoherenc... |
| GlobalColorHist... | 1.0 | Yes | color | | Get the global c... | |
| LocalColorHisto... | 1.0 | Yes | color | | Get the local col... | |
| ColorMoment | 1.0 | Update | color | Current version i... | Get the color m... | |

Sharing 1821 files

**Figure 7: Downloading plug-in module for DISCOVIR**

## 4. DISCOVIR EVERYWHERE

Although migrating CBIR on P2P network has many advantages as aforementioned, this system still encounters limitations like the requirement of installing client software and the low accessibility compared to WWW. These limitations always introduce inconveniences for the users when they are not using their own computers. For example, when the users are in libraries or cyber-cafes, installing personal software is usually prohibited, therefore, they cannot use the services. For this reason, some web-based P2P applications have been developed. With these services, user can access the P2P network through a web browser while the web server serves as one of the peer in the P2P network. Some examples of web-based P2P applications are **AsiaYeah** [1], **Gnutellait** [9], **LinkGrinder** [16] and **AudioFind** [2].

When users submit queries through the web page, the server helps distributing the query and collecting the results from the P2P network. Such kind of search engine for file retrieval through P2P network provides an alternative source of files in addition to the documents on WWW. This provides a more comprehensive and larger file database when the number of users in the P2P network is large enough. However, there are drawbacks concerning these models:

1. **Centralization** - The web server is public to everyone who is able to access the web page. When many users use this service, the server has to handle huge amount of queries and collection of results. The problem remains the same as prevalent search engine. Moreover, the web server will generate lots of traffic to its neighboring peers, which skews the workload in P2P network.
2. **CBIR functionality** - All the web-based P2P applications mentioned above are based on text search.

When adapting to CBIR approach, it will incur lots of penalty when feature extraction of images is done by the server.

## 4.1 Design Goal of DISCOVIR Everywhere

To account for the two problems raised above, *DISCOVIR Everywhere* overcomes them by distributing the heavy workload to peers evenly while keeping its accessibility through web. Unlike other web-based P2P applications, the web server does not act like a peer in the P2P network. Instead, it coordinates the forwarding of queries and returning of result between web clients and peers. Preprocessing of query image, initiation of query and collection of result are all done in a DISCOVIR peer assigned by the web server. Even there are huge number of user, this architecture is scalable because the web server is only responsible for distributing workload to DISCOVIR client. It allows users to perform CBIR in P2P network through a web browser or a WAP phone. Since the mobile network, the Internet and DISCOVIR are three separate networks running on their own protocols. The idea of *DISCOVIR Everywhere* is also to bridge three networks together using light weighted access gateways.

## 4.2 Components in DISCOVIR Everywhere

Referring to Fig. 8, we identify the four main components in the *DISCOVIR Everywhere* design. They are:

- **Web Browsers and Mobile Devices** - It is a device running a web browser with network access to the WWW. The mobile devices can be WAP phone or PDAs with wireless access to the *DISCOVIR Everywhere* Gateway, they can access the web page either by WML or XML.
- **DISCOVIR Peers** - They are interconnected computers running the DISCOVIR client program in the Internet. In addition to P2P query service, the HTTP Agent of each peer will accept GET and POST HTTP requests to provide seamless integration with the WWW. Moreover, every DISCOVIR peer is required to send 'heart-beat' message to the Bootstrap Server periodically to indicate their availability in DISCOVIR network.
- **DISCOVIR Bootstrap Server** - It is originally the host cache of P2P network in LimeWire. In *DISCOVIR Everywhere*, it is responsible for maintaining an updated list of accessible DISCOVIR peers and their availability for providing HTTP access, if it cannot receive the 'heart-beat' message from a peer for a certain period of time, its record will be removed and considered off-line.
- ***DISCOVIR Everywhere* Gateway** - It is a server program providing users with web-based searching interface. It contacts the Bootstrap Server for the list of IP address of available DISCOVIR Peers and coordinates the redirection of users' query request to different peers.

## 4.3 Procedure for Searching

Referring to Fig. 9, the process of query in *DISCOVIR Everywhere* consists of six steps. Details of each step are shown in the following:

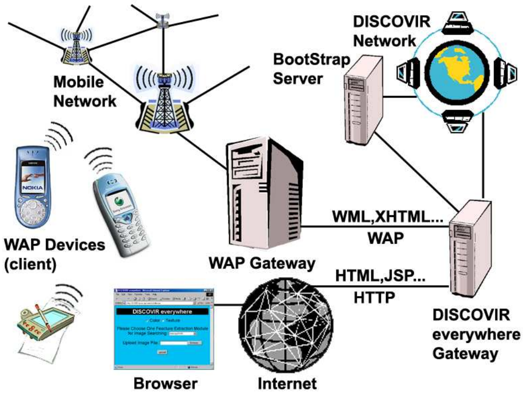


**Figure 8: Architecture of DISCOVIR everywhere**

1. A user initiates a query request for similar images through the web interface provided by *DISCOVIR Everywhere* Gateway. In addition, he also sends the type of feature extraction method he intends to use to the gateway. If the user access the web interface by a WAP phone, he should access it through the WAP Gateway and the medium of communication is WML instead of HTML.

2. The *DISCOVIR Everywhere* Gateway receives the request and inquire the bootstrap server about the IP address and port number of a DISCOVIR peer capable of handling this query.

3. Upon receiving the inquiry from *DISCOVIR Everywhere* Gateway, the DISCOVIR Bootstrap Server picks one of the available peers from the list in a round robin manner in order to distribute the workload evenly. This is similar to the techniques used by DNS servers to distribute workload among web servers.

4. Once knowing the IP address of DISCOVIR peer capable of handling the query, the gateway generates a HTML page instantly for users to upload his query image and intended feature extraction method to the selected peers using HTML form submission procedure.

5. User uploads the query image to selected DISCOVIR peer through a HTTP POST request. Meanwhile, the *HTTP Agent* of that selected peer stores the image and requests its *Feature Extractor* to extract feature and assemble an ImageQuery message to be sent out through *Packet Router*, which is analogous to the processing of initiating query using the DISCOVIR client program. The web browser keeps this HTTP connection open until results return from the DISCOVIR peer.

6. Once the selected DISCOVIR peer accumulates up to a certain number of results or reaches a time limit, it packages the result in HTML format and sends back to user through *HTTP Agent*. With the inherent support of HTTP defined in Gnutella protocol, web users are able to download thumbnails or full size images directly from DISCOVIR peers without the help of gateway.

## 4.4 Advantages of DISCOVIR Everywhere over Prevalent Web-based Search Engine

Compared to existing search engines and web-based P2P services, *DISCOVIR Everywhere* exhibits the following advantages:

1. **Comprehensiveness** - By utilizing the storage capacity and individual contribution of peers in the network, we increase the comprehensiveness of data archive for searching. Besides, the web-based interface provide a handy access compared to using pre-installed P2P client programs.

2. **Query Richness** - *DISCOVIR Everywhere* possess CBIR functionality beyond existing text based retrieval, while eliminates the need for preprocessing, storage and indexing in existing CBIR search engines by delegating them to peers in DISCOVIR network.

3. **Scalability** - Compared to existing web-based P2P service, the *DISCOVIR Everywhere* Gateway is much more light weighted. Instead of serving as a proxy for web users to access the P2P network, it takes the role of coordinator between web users and DISCOVIR peers. Apart from reducing the workload for initiation of query and collection of results, this also avoids perverted usage of P2P network by distributing query requests among peers evenly.

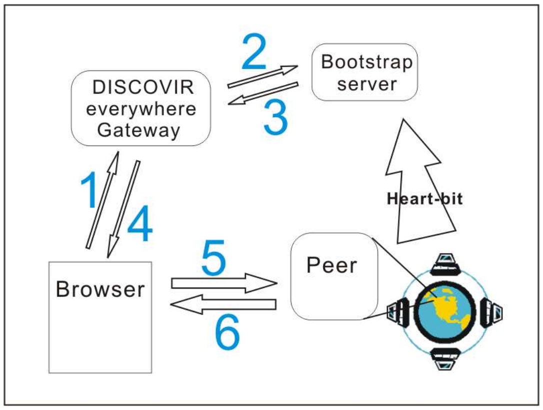


**Figure 9: Query Procedure of DISCOVIR Everywhere**

# 5. IMPLEMENTATIONS AND FUTURE WORKS

We have built a working implementation of DISCOVIR. The client is implemented in Java and makes use of the LimeWire open source project. LimeWire is a Java implementation of Gnutella Network and its key components are under GNU Public License. We have released DISCOVIR as a free software and intend to release the source code in near future.

Figure 10 illustrates the top ten results panel, it makes use of the generated thumbnails to avoid heavy traffic caused by downloading full size image directly, while users can preview lower quality images. Figure 11 shows the drawpad functionality of DISCOVIR. Besides supplying an example image for query, users may draw their own sketch and search. Figure 7 shows the plug-in module download page of DISCOVIR. Currently, DISCOVIR supports AverageRGB, GlobalColorHistogram, LocalColorHistogram, ColorMoment and ColorCoherenceVector on color-based feature, Co-occurrence matrix, Auto correlation, Edge frequency and Primitive length on texture-based feature. We are working on shape-based feature and encourage contributions from the community by implementing plug-ins.

*DISCOVIR Everywhere* Gateway is hosted on a Tomcat server. The HTTP Agent of DISCOVIR client program is modified to support HTTP POST request from web client to accept incoming images. One of the shortcomings in current design depends on trustworthiness of web users. A malicious web user may attack a specific peer with burst of query messages, a key agreement protocol between DISCOVIR peers, Bootstrap server and *DISCOVIR Everywhere* gateway should be used to safeguard the peers.

With the recent advances in web service technologies, one may envision the marriage of web service and DISCOVIR peer, *DISCOVIR Everywhere* may adopt the service-oriented architecture (SOA), while the communications between gateway, DISCOVIR peers and web clients all standardized to use SOAP.

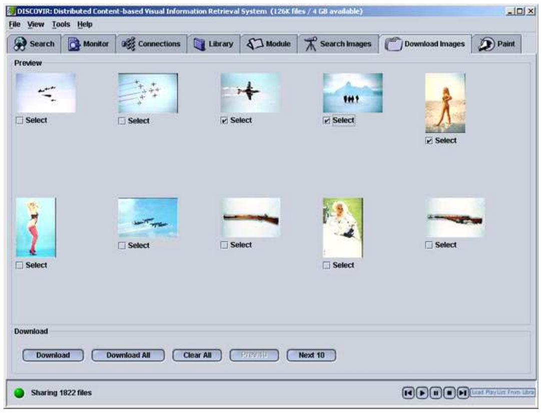


**Figure 10: Top 10 result of retrieval**

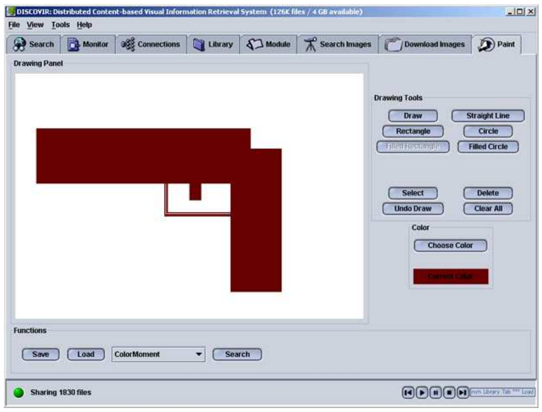


**Figure 11: Drawpad functionality of DISCOVIR**

# 6. CONCLUSION

In this paper, we marriage the traditional CBIR with the P2P network to distribute storage capacity and workload among peers and provide content-based search in P2P network. We illustrate the design and implementation of DISCOVIR, in order to exhibit the key components required in a P2P based CBIR system. We also illustrate how its accessibility can be improved by the use of World Wide Web.

The DISCOVIR system contains the key components required in a P2P based CBIR system. Connection Manager and Packet Router are used to establish connection and handle query routing. Plug-in Manager, Feature Extractor and Image Indexer are used to extract feature vectors from images, index images and calculate similarly measures. HTTP Agent is used to retrieve images from other peers.

We proposed the *DISCOVIR Everywhere* protocol to deal with the accessibility problem occurring in most prevalent P2P applications. The *DISCOVIR Everywhere* gateway receives query messages using HTTP protocol and distribute it among different peers using bootstrap server. The DISCOVIR Everywhere protocol utilize the storage and processing power of each peer instead of the bootstrap server. Thus, the accessibility can be improved while maintaining the scalability of the system.

## Acknowledgments

DISCOVIR is built based on LimeWire open source project. This research work is supported in part by an Earmarked Grant (#2150317) from the Research Grants Council of Hong Kong, SAR.

## 7. REFERENCES

[1] AsiaYeah Gnutella Search. http://www.asiayeah.com/service/searchFile_c.jsp.

[2] AudioFind. http://www.audiofind.com/.

[3] R. J. J. Bayardo, R. Agrawal, D. Gruhl, and A. Somani. Youserv: A web-hosting and content sharing tool for the masses. In *Proceedings of 11th World Wide Web Conference*, May 2002.

[4] G. Coulouris, J. Dollimore, and T. Kindberg. *Distributed Systems Concepts and Design.* Addison-Wesley, third edition, 2001.

[5] DIStirbuted COntent-based Visual Information Retrieval. http://www.cse.cuhk.edu.hk/∼miplab/discovir.

[6] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, N. W., D. Petkovic, and W. Equitz. Efficient and Effective Querying by Image Content. *Journal of Intelligent Information Systems: Integrating Artificial Intelligence and Database Technologies*, 3(3-4):231–262, 1994.

[7] The Freenet homepage. http://freenet.sourceforge.net.

[8] The Gnutella homepage. http://www.gnutella.com.

[9] Gnutellait Web Search. http://gnutella.abctella.it/.

[10] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison Wesley, first edition, 1992.

[11] KaZaA file sharing network. http://www.kazaa.com/.

[12] I. King and Z. Jin. Relevance feedback content-based image retrieval using query distribution estimation based on maximum entropy principle. In *Proceedings to the International Conference on Neural Information Processing (ICONIP2001)*, 2001.

[13] T. K. Lau and I. King. Montage: An image database for the fashion, clothing, and textile industry in Hong Kong. In *Proceedings of the Third Asian Conference on Computer Vision (ACCV'98)*, Lecture Notes in Computer Science, January 4-7, 1998.

[14] The LimeWire homepage. http://www.limewire.com.

[15] Modern Peer-to-Peer File-Sharing over the Internet. http://www.limewire.com/index.jsp/p2p.

[16] LinkGrinder. http://www.gnutellagrinder.com/.

[17] S. Mehrotra, Y. Rui, M. Ortega, and T. Huang. Supporting Content-based Queries over Images in MARS. In *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, pages 632–633, 1997.

[18] The Morpheus homepage. http://www.musiccity.com.

[19] The Napster homepage. http://www.napster.com.

[20] A. Natsev, R. Rastogi, and K. Shim. WALRUS: A Similarity Retrieval Algorithm for Image Databases. In *Proc. SIGMOD, Philadelphia, PA*, 1999.

[21] C. H. Ng and K. C. Sia. Peer Clustering and Firework Query Model. In *Poster Proc. of The 11th International World Wide Web Conference*, May 2002.

[22] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Tools for Content-based Manipulation of Image Databases. In *Proc. SPIE*, volume 2185, pages 34–47, February 1994.

[23] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. In *In Proc. ACM SIGCOMM*, August 2001.

[24] Y. Rui, T. S. Huang, and S.-F. Chang. Image Retrieval: Current Techniques, Promising Directions and Open Issues. *Journal of Visual Communication and Image Representation*, 10:39–62, April 1999.

[25] The Search for Extraterrestrial Intelligence homepage. http://www.setiathome.ssl.berkeley.edu.

[26] K. C. Sia, C. H. Ng, C. H. Chan, and I. King. P2P Content-Based Query Routing Using Firework Query Model. In *The 2nd International Workshop on Peer-to-Peer Systems, submitted*, Feb 2003.

[27] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jai. Content-Based Image Retrieval at the End of the Early Years. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000.

[28] J. R. Smith and S. F. Chang. An Image and Video Search Engine for the World Wide Web. In *Proc. SPIE*, volume 3022, pages 84–95, 1997.

[29] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *Proc. of ACM SIGCOMM*, pages 149–160, August 2001.

[30] J. Z. Wang, G. Li, and G. Wiederhold. SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries. In *IEEE Trans. on pattern Analysis and Machine Intelligence*, volume 23, pages 947–963, 2001.