Location-dependent Content-based Image Retrieval System Based on a P2P Mobile Agent Framework

Yusuke Ariyoshi
Faculty of Economics, Management and Information Science,
Onomichi City University
Onomichi, JAPAN
email: y-ariyoshi@onomichi-u.ac.jp

Junzo Kamahara, Naoki Tanaka, Katsutoshi Hirayama, Takashi Nagamatsu
Graduate School of Maritime Sciences
Kobe University
Kobe, JAPAN
email: {kamahara, ntanaka, hirayama}@maritime.kobe-u.ac.jp
nagamatu@kobe-u.ac.jp

Yuuichi Teranishi
Cyber Media Center,
Osaka University
Suita, JAPAN
email: teranisi@cmc.osaka-u.ac.jp

***Abstract*—In this article, we propose a geoconscious content-based image retrieval system based on a P2P mobile agent framework. This system retrieves similar photographs from an image database of location-dependent photographs (e.g., photos of buildings, landmarks, etc.), which use GPS positions for geo-tagging. The P2P mobile agent framework supports intelligent agents. This agent searches similar content image using a query photograph by traversing the P2P network, instead of the mobile device issuing the query. In this paper, we describe the design of the proposed system and a portion of its implementation. This prototype system produces a new peer and rearranges the placement of image agents among peers for workload balancing. Furthermore, we provide the experimental results of our implementation for managing location-dependent image agents, which are clustered with peers in a distributed Delaunay network.**

Keywords-component; P2P; Content-Based Image Retrieval; Geolocation; Geo-tagging; Geoconscious; Delaunay network

I. INTRODUCTION

Based on the success and growth of photo-sharing sites such as Flickr, a large number of photographs are being shared over the Internet. The number of uploaded images is increasing rapidly, and an even greater number of images will be shared in the future owing to the increasingly widespread use of mobile devices that have cameras, such as smartphones. Searching similar photographs from such a huge image database is a typical problem in content-based image retrieval (CBIR).

These photographs can be classified into a number of types, including portraits, landscapes, artistic, and documentary photographs. The target of our research is retrieving location-dependent photographs (e.g., photographs of buildings or landmarks). The location of a photograph is indicated by the GPS position, i.e., the latitude and longitude where the photographer has taken the picture. This embedding of locational information into a photograph is known as geo-tagging.

We studied a method for retrieving similar images efficiently by utilizing the geolocation of images. It is efficient to limit the geographic range when searching for a location-dependent image. Using a combination of the geographic distance and image distance (using image features) also increases the performance of CBIR. We call this type of process "geoconscious" CBIR [1]. In this paper, the term geoconscious means that the geolocation is the primary criterion used for the retrieval of images.

For image retrieval, calculating various image features and comparing their high-dimensional vectors requires a significantly high calculation cost. While the calculation cost of image pairing is relatively low, comparing a photograph with a vast image dataset to find a similar image requires a large amount of computation. Therefore, as such workload increases, the efficiency of our retrieval method has to increase in-scale with the computation power required.

Such a complex CBIR requires intelligent behavior for each entity used in calculating the similarities. The main concept underlying geoconsciousness is the management of intelligent photo agents based on their locations on the P2P network. As intelligent agents on the P2P network work in a scalable manner, finding similar images using image features and geolocations is a suitable method for a mobile agent-based P2P framework. Structured P2P networks exhibit good usage efficiency for a specific search range. Therefore, we adopted P2P Interactive Agent eXtension (PIAX) [2] as such a framework.

PIAX is a structured P2P framework with an enhanced location-oriented service. It can handle intelligent agents that move across a structured overlay network with multiple peers. Agents can be implemented to calculate various types

978-1-4244-9529-0/13/$31.00 ©2013 IEEE

of features. Furthermore, PIAX supports a geometrical range search for the agents on its network.

A. Related Works

For P2P multimedia searches, there have been short surveys [3-5] on CBIR from P2P networks. However, the goal of these studies was to create generic systems for retrieving images or multimedia data without regard to their geolocation.

For image retrieval, the geo-tagging of photographic images has emerged as a research field over the past decade. A survey in [6] introduced the concepts of geolocation and landmark recognition. The geolocation of a photograph does not directly represent the landmark in the image, because the GPS position is not that of the landmark itself but rather the position from where the photograph was captured. Moreover, the content of a photograph might not include only a landmark but may also include various other objects, such as people, food, and souvenirs. Therefore, landmark recognition uses keypoint (local feature) matching, because landmark recognition needs local feature matching of the object in the image and not global feature matching of the images. [7] classified landmarks in large-scale image collections without the use of geolocations, and [8] constructed a 3D model using the GPS positions and camera parameters for previously known landmarks.

The remainder of this article is organized as follows. The next section outlines the concept of geoconscious P2P CBIR. In section 3, we describe the suitable agents used in our system. In section 4, we describe some of the noted behavior of our system, and in section 5, we discuss the results of a preliminary experiment. Finally, we give some concluding remarks regarding this study and describe potential future research in section 6.

II. GEOCONSCIOUS P2P MOBILE AGENT CONTENT-BASED IMAGE RETRIEVAL

In this section, we give an overview of our P2P system architecture. Our system utilizes two layers. The lower layer is a P2P network layer, which consists of peers and an overlay network. The peers provide agent functions, such as agent creation, to the upper layer. The overlay network provides a location-dependent query function, such as range query. The upper layer is the agent layer. In this layer, several types of mobile agents work together to realize the application services. Our geoconscious CBIR system includes an image agent (IA), which manages the image date and its meta data; a cluster agent (ClA), which manages the geographical region and IAs within the region; a crawling agent (CrA), which crawls to collect geo-tagged images on the Internet; a web agent (WA), which is the web user interface of our system; and a query agent (QA), which executes a similar image retrieval query from the user. Fig. 1 shows a diagram of the relationships among these agents.

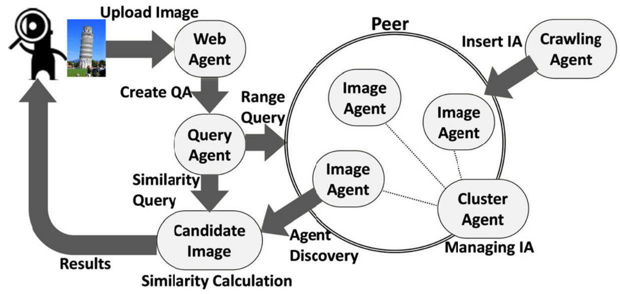

Figure 1. Relationships between different agents in our system

III. AGENT DESIGNS

A. Crawling Agent

The CrA crawls to collect shared geo-tagged photographs on the Internet. To obtain the photographic images, the CrA uses Flickr, Panoramio (http://www.panoramio.com), and other sites. The CrA then produces a new IA to manage the image and other textual or geo-tagged information. The generated IA will move to the proper peer within its location.

B. Image Agent

The IA has functions for calculating several image features and managing the geo-tag and textual information when present. Each IA corresponds to one image of the photograph. Naturally, the number of IAs in a P2P network is the same as the number of images that can be searched.

The functions of the IA are as follows:

1) Calculate the image features

The IA should contain low-level features of the image.

2) Find and move to the peer that manages the locations near the geolocation of the image

Each ClA manages a different geographic region. The IA asks the PIAX for the proper ClA using the geographic position of the image.

3) Calculate the similarities based on the queried features

The IA is requested to calculate the degree of similarity between the queried image and the managed image using a formula that specifies how to assemble the image features and geolocation into a similarity score.

4) Store and recover the features to/from storage

PIAX has a persistent function to store and recover the features to and from the node storage.

The IA should only include simple functions because the discovery of agents and the ranking of images are performed by other agents. In our architecture, IAs that have similar images should be located in the same peer to reduce the amount of network traffic for query messaging because the QA moves to a peer containing the candidate IAs before the retrieval.

C. Cluster Agent

The ClA manages the geometric region of the IAs in the peer as a Voronoi region. In a Voronoi region, any point is guaranteed to be closer to the kernel point of that Voronoi region than to any other kernel point [9]. Fig. 2 shows an example of a Voronoi diagram.

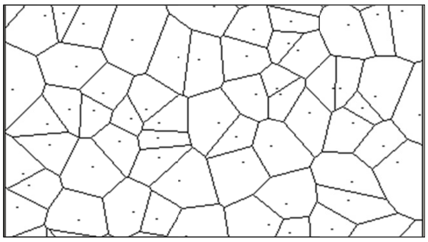

Figure 2. Example of a Voronoi diagram

PIAX supports a Delaunay overlay network. Delaunay triangulation describes the dual structure of a Voronoi diagram. The ClA calculates the Voronoi region by acquiring the locations of its own kernel point and neighbor kernel points from the peer.

Furthermore, the ClAs should optimize the workload balance of the peers. When a new IA is added to a peer, if the number of IAs in the peer is above a certain threshold, the ClA will split into two ClAs and produce a new peer that manages half of the existing IAs. The details of this are described in a later section. In future research, we will approach this problem as a generalized mutual assignment problem [10].

D. Web Agent

The WA is a web user interface on a P2P network. The WA can exist on multiple peers, and users can access the WA and input a photograph through a web page. After users upload photographs, a QA is created.

The WA will display the ranked image results on the web page when they are returned from the QA. The web page may be implemented using Ajax to show the image results asynchronously.

As there is no central management of the agents, we are not limited to the use of a single WA. The WAs must join only the P2P network of our system (See Fig. 3). This ensures the scalability of the image search and retrieval used in our architecture, because the QAs produced by a WA can query each IA of different peers in parallel.

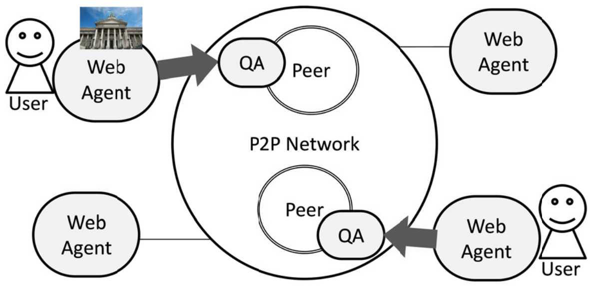

Figure 3. Scalability of multiple user search

E. Query Agent

The QA executes a similar image retrieval query from the user. The functions of the QA are as follows:

1) Calculate image features

The QA also has to contain features of the image.

2) Find and move to the peer that manages the locations near the geolocation of the image

This function is the same as that in the IA.

3) Form the range query for agent retrieval

The QA forms and sends a range query using the geolocation of the image, and receives the IAs for the retrieved candidates.

4) Obtain the similarity score from each retrieved IA

The QA issues the query message to the candidate IAs to calculate the similarity score between the images of the QA and IAs.

5) Rank the IAs based on their similarity scores

To return the similarity score results to the WA, the QA must order the images based on their score.

A QA is a subclass of an IA. The functions of a QA appear very similar to those of an IA. However, a QA has the functions to obtain the similarities and calculate the ranking of images. The QA will be dismissed after returning the results to the WA.

.

IV. Implementing System Behavior

A. Constructing the Image Database

Initially, we need to form an image database on a structured P2P network. This database contains several image features extracted by analyzing the images. Furthermore, clustering the images based on their geo-tags allows the images to be collected based on their neighboring order. Each cluster is represented as a ClA according to its geolocation. The ClAs manage the geometric region. The ClAs also assign the IAs to a specific peer for balancing the workload.

The CrA collects geo-tagged photographs and produces new IAs to manage the images and other information. When an IA is created, the IA should calculate the image features to determine the similarities to other images. After calculating the image features, the IA will request the ClAs to find a proper peer according to the geolocation of the image. One ClA exists on each peer. Therefore, the geographically corresponding ClA will answer the IA, which then moves into that peer. PIAX supports mobile agents. The workflow of this is shown in Fig. 4.

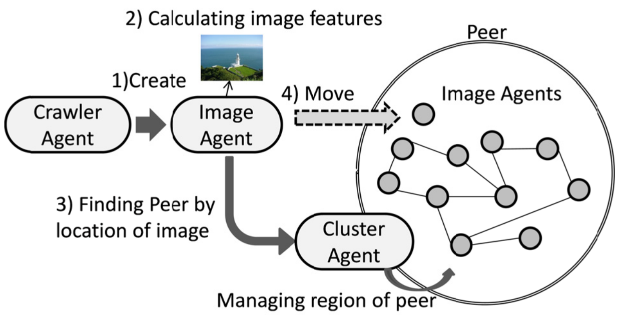

Figure 4. Workflow of image database construction

B. *Splitting region for load-balancing*

When a new IA is to be added to a peer, the ClA checks the number of IAs already existing in the peer. If the number of IAs, which indicates the capacity of the peer, is over a certain threshold, the ClA will split into two ClAs and produce a new peer for load balancing (Fig. 5).

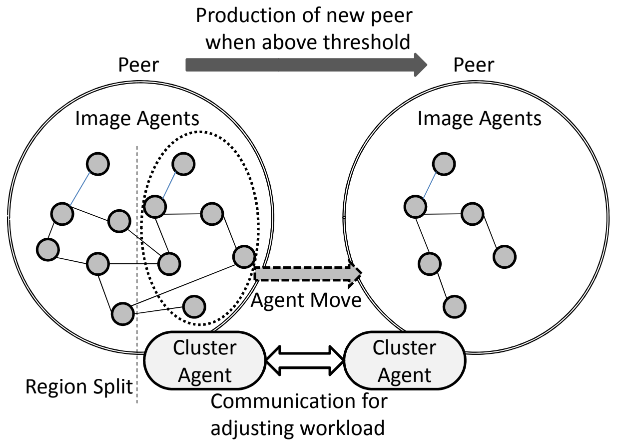

Figure 5. Balancing the workload of peers by moving the IAs

When a new peer is produced by splitting an overloaded peer, a method for dividing the region must be chosen. First, the gravity center of the IAs in the region will be calculated. The gravity center may differ from the kernel point. Therefore, we can obtain a new kernel point as the reflection of the existing kernel point through the gravity center. A new peer will be created with this new kernel point. The Voronoi region will then be divided mid-perpendicular of the two kernel points. The new boundary of the two Voronoi regions will be created. The region that has the old kernel point may not be changed without this new boundary. The region that has the new kernel point will be changed based on the new relationship with the other kernel point. Therefore, recalculating the Voronoi region of the new peer will cause a rearrangement of the image agents that belong in the neighbor peers (Fig. 6).

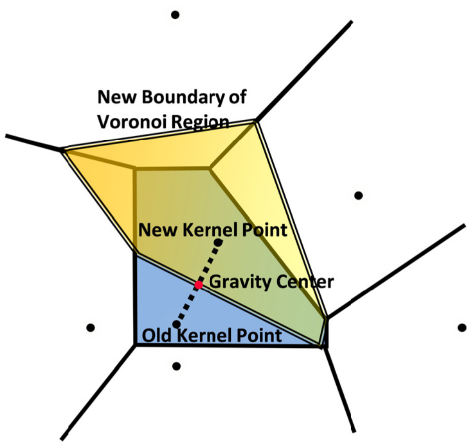

Figure 6. Relation between new and old kernel points

C. *Distributed Delaunay Network Generation*

A Delaunay overlay network is generated at the P2P network layer by an autonomous distributive algorithm [11]. Fig. 7 shows a summary of this algorithm. The Delaunay network is made by repeating the following steps on each peer:

```
1.  List NodeList
2.  List NeighborNodeList
3.  List NonNeighborNodeList
4.  while(true)
5.     clearAll NonNeighborNodeList
6.     addAll NeighborNodeList to NodeList
7.     clearAll NeighborNodeList
8.     sortAsClockwise NodeList
9.     localDelaunayTriangulation(NodeList,
         NeighborNodeList, NonNeighborNodeList)
10.    delegation(NeighborNodeList,
         NonNeighborNodeList)
11.    notificationOfTriangulation (NeighborNodeList)
```

Figure 7. Algorithm of distributed Delaunay network generation

1) *Local Delaunay Triangulation:* A peer calculates the proper Delaunay network around itself based on the location information of the other peers that this peer holds. Other peers are divided into neighbors and non-neighbors from the calculation results.
2) *Delegation:* A peer does not require information of its non-neighbors. Information of a non-neighbor is passed to the nearest neighbor from the non-neighbor. The peer then deletes information of the non-neighbors.
3) *Notification of Triangulation:* During this procedure, each neighbor is told of its adjacent peers, as recognized from step 1, to assure communication between them.

D. *Retrieve Similar Image.*

When a user searches for a photograph, he/she should input a query image into the WA. The WA then creates a QA from the query image. The QA constructs the query message for PIAX in the form of a range query based on the location. PIAX will answer with the agents that exist within the query range. The value of the query range radius affects the search performance, which we will discuss later.

When the QA receives the list of agents (i.e., candidate images managed by the agents), the procedure used for calculating the similarities between the query image and the agent is performed for each agent on the list. There are several methods for calculating an image similarity. The QA issues a query with a formula for calculating a similarity using image features and the geographic distance into the IA. The QA then sorts the similarity information from each IA and presents the results to the user. A diagram of the relationship between agents is shown in Fig. 8.

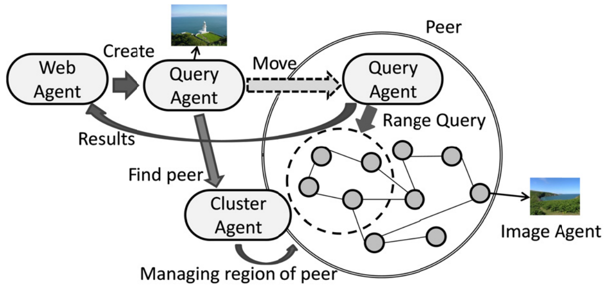

Figure 8. Workflow for retrieving a similar image

E. Ranking Function

Considering the distance between query image and target image improves the performance of the location-dependent image retrieval. Our research [12] shows that our ranking function, which combines the image distance with the geographic distance, has a better performance than a ranking function that uses only the image distance or the geographic distance. The goal of our image retrieval is to find images "orthologous" to the query image. We define an orthologous image as a similar image that captures the same object, such as a building or landmark, contained in the query image. We propose the use of a conjunctive ranking function, i.e., an orthologous identity function (OIF), to calculate the orthologous identity distance, which represents the degree of similarity between the query image and the candidate image. The definitions of OIFs are shown below:

$$OIF(P_q, P_c) = IMGDist(P_q, P_c) + \exp(GEODist(P_q, P_c)) - 1 \quad (1)$$

$$OIF'(P_q, P_c) = 0.5IMGDist(P_q, P_c) + 0.5GEODist(P_q, P_c) \quad (2)$$

GEODist(P_q, P_c) is defined as follows:

$$GEODist(P_q, P_c) = \frac{d(P_q, P_c)}{R} \quad (3)$$

,

where $d(P_q, P_c)$ is the physical geographic distance between P_q and P_c, and R represents the radius of retrieval specified by the user.

V. PRELIMINARY EXPERIMENT

We developed a mobile agent P2P geoconscious image search system. Thus far, we have implemented an image database construction function using PIAX, image feature calculations, and a comparison function.

A. Image Database Construction

For an image database construction test using PIAX, we used the 64-bit edition of Windows 7 on VMware Player with an Intel CPU i7-3667U 2.0 GHz 2Core/4Thread and 4 GB of memory. The IA created from the geolocation data searches for a proper peer according to its geolocation and moves to the peer. The number of IAs is then checked. If the number of IAs is over the threshold, the peer is divided and the IAs re-allocated. In this test, the IA did not process the image feature calculation or a comparison, and only geolocation data were used to check the performance around the PIAX framework. For this, we used 100,000 geolocation datasets from Flickr. The initial number of peers was 3 × 3 = 9. The location of the initial peers was changed by adding a random number to every trial. The threshold used to divide a peer for load balancing was 500 AIs.

For an average of ten trials, it took 1,062 s to process 100,000 geolocations. After the trial, the number of peers was 441.7, and the average number of IAs per peer was 226.6. When a peer was divided, 87.9 IAs moved to new peers from the neighbor peer, with the exception of the peer that was divided. Fig. 9 shows the results from our image database construction algorithm. The blue lines represent a Delaunay overlay network, which connects peers, and each gray polygon indicates a Voronoi region managed by a ClA.

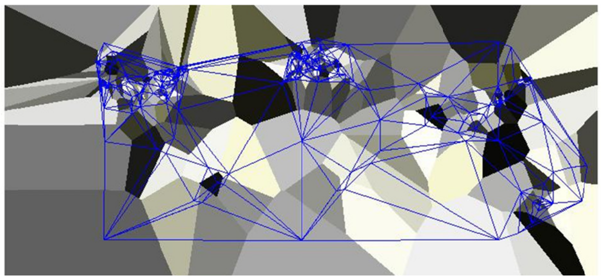

Figure 9. Voronoi regions and Delaunay overlay network

B. Image Feature Calculation and Comparison

For this test, we used the 64-bit edition of Windows 7 Professional on VMware Player with an Intel CPU i7-3960X 3.3 GHz 2Core/4Thread and 8 GB of memory. The test program read the URL list of Flickr and produced 92,948 IAs on a peer. The sequential time taken to read the images and calculate the MPEG-7 image features was about 7 h. This equates to 0.271 s per image. We used the following five image features for the test: color layout, color structure, dominant color, edge histogram, and scalable color.

It took 5.12 s to compare a query with the 92,948 images. Therefore, one IA processed the query in 0.055 ms. A peer may have only 18,000 IAs within a 1 s processing time. Therefore, we must test multiple peers for query processing.

C. Search Radius vs. Retrieval Performance

A location-dependent image in a certain search radius may be retrieved based on its own GPS position. If there are no errors in the captured GPS data, the search radius can be as small as possible. However, a GPS error is unavoidable. Therefore, the search radius needs to cover a certain extent of the location within the estimated GPS accuracy. Otherwise, a small radius loses correct images that are out of range. On the other hand, a large search radius will decrease the performance of the image retrieval because several unrelated images will appear from a large search radius. In this way, there is a tradeoff between the search radius and image retrieval performance.

Fig. 10 shows the results of our experiment for comparing the performance of OIFs against the geographic distance and color histogram (image distance). OIF(1) uses (1) while OIF(2) uses (2). The performance results are shown as the mean reciprocal rank (MRR) [13]. The MRR of the image distance is a constant value on the radius because the image distance is unrelated with the geographic distance. Obviously, the performance of the OIFs overcomes both the geographic and image distances. The best radius in this experiment is 113 m for OIF(1) and 130 m for OIF(2). The slope of the OIFs declines after the best radius.

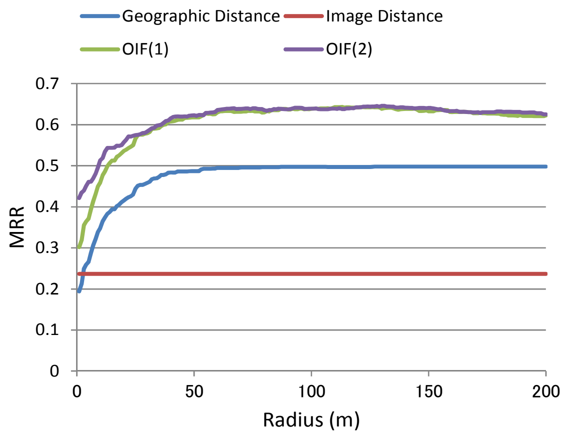

Figure 10. Radius of range search vs. MRR

VI. CONCLUSION

We proposed a design for a geoconscious CBIR system based on a P2P mobile agent framework. Geoconscious CBIR will enhance the image retrieval accuracy for location-dependent photographs by limiting the geographic range for finding the image and using a combination of geographic and image distances. The P2P mobile agent framework will accelerate the performance and scalability of the retrieval.

We used the PIAX P2P mobile agent framework, as it adapts an overlay network and supports a structured P2P architecture. We proposed the design of different agents for this framework, as well as a protocol for constructing an image database and for CBIR. Furthermore, we showed the results of preliminary experiments using the implementation of an image database construction function and a geoconscious image search function.

We are currently combining the database construction and image feature calculation functions together, and developing an additional image retrieval function.

ACKNOWLEDGMENT

This work was supported by KAKENHI (22300035).

REFERENCES

[1] J. Kamahara, N. Tanaka, K. Hirayama, T. Nagamatsu, Y. Teranishi, and Y. Ariyoshi, “Design of GeoConscious P2P Content-based Image Retrieval,” Proc. 7th Intl. Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA2012), Nov. 2012.

[2] Y. Teranishi, “PIAX: Toward a Framework for Sensor Overlay Network,” Proc. 6th Annual IEEE Consumer Communications & Networking Conference (CCNC 2009 Workshops), Jan. 2009, pp. 1-5, http://piax.org/en/.

[3] J. Li and G. Zhang, “The State of the Art in Content-Based Image Retrieval in P2P Networks,” Proc. ACM 2nd International Conference on Internet Multimedia Computing and Service (ICIMCS’10), Dec. 2010, pp. 143-146, doi: 978-1-4503-0460-3/10/12.

[4] M. Batko, V. Dohnal, D. Novak, and J. Sedmidubsky, “MUFIN: A Multi-Feature Indexing Network,” Proc. IEEE 2nd International Workshop on Similarity Search and Applications (SISAP’09), Aug. 2009, pp. 158-159, doi: 10.1109/SISAP.2009.24.

[5] X. Meng, C. Feng, and Y. Wang, “Efficient Image Retrieval in P2P Using Distributed TS-SOM and Relevance Feedback,” Proc. International Symposium on Distributed Computing and Applications to Business; Engineering and Science (DCABES2007), Aug. 2007, pp. 1066-1071.

[6] J. Luo, D. Joshi, J. Yu, and A. Gallagher, “Geotagging in Multimedia and Computer Vision—A Survey,” Multimedia and Tools Application, vol. 51, no. 1, Jan. 2011, pp. 187-211, doi:10.1007/s11042-010-0623-y.

[7] Y. Li, D. J. Crandall, and D. P. Huttenlocher, “Landmark Classification in Large-scale Image Collections,” Proc. IEEE 12th International Conference on Computer Vision (ICCV 2009), Sept. 2009, pp. 1957-1964, doi: 10.1109/ICCV.2009.5459432.

[8] X. Li, C. Wu, C. Zach, S. Lazebnik, and J. M. Frahm, “Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs,” Proc. ACM 10th European Conference on Computer Vision (ECCV ’08), Oct. 2008, pp.427-440, doi: 10.1007/978-3-540-88682-2_33.

[9] F. Aurenhammer, “Voronoi Diagrams — A Survey of a Fundamental Geometric Data Structure,” ACM Computing Surveys, vol. 23, no. 3, Sept. 1991, pp. 345-405, doi: 0360-0300/91/0900-0345.

[10] K. Hirayama, T. Matsui, and M. Yokoo, “Adaptive Price Update in Distributed Lagrangian Relaxation Protocol,” Proc. the 8th International Joint Conference on Autonomous Agents & Multi-Agent Systems (AAMAS-2009), May 2009, pp. 1033-1040.

[11] M. Ohnishi, Y. Minamoto, T. Eguchi, H. Kato, R. Nishide, and S. Ueshima, “Autonomous and Distributive Generation Algorithm of Delaunay Network for P2P Model Utilizing Node Location,” IPSJ Transaction on Database, vol. 47, Mar. 2006, pp. 51-64.

[12] J. Kamahara, T. Nagamatsu, and N. Tanaka, “Conjunctive Ranking Function using Geographic Distance and Image Distance for Geotagged Image Retrieval,” Proc. the ACM multimedia 2012 workshop on Geotagging and its Applications in Multimedia (GeoMM’12) , Nov. 2012, pp. 9-14.

[13] D. R. Radev, H. Qi, H. Wu, and W. Fan, “Evaluating Web-based Question Answering Systems.” Proc. of 3rd Int. Conf. on Language Resources and Evaluation (LREC 2002), May. 2002.