「マルチメディア，分散，協調とモバイル
(DICOMO2016)シンポジウム」 平成28年7月

Aerial Image Discovery System with Accelerated Candidate Selections in Multi-Scale City Image Database

Kaoru Uchida[1] Shuo Hu[1] Bingjie Ma[1] Rheza Harliman[1]

Abstract: Aerial image discovery is one of the key technologies in various services using GIS, in which a match is to be identified in a large image database with a given query image with unknown location, orientation, and scale. The task is more challenging in city areas because the views have higher degree of homogeneity compared to non-urban areas. We propose a highly efficient image discovery system to find a match in the multi-layered image database of city aerial images with multiple magnitudes. The system first pre-selects matching candidates based on comparison results of image profiles such as frequency spectrum, then search in the database using scale-invariant matcher based on local feature descriptors. The matching focuses on the appropriate scale layer to accelerate search in the large database. This paper outlines our proposed approach and discusses its feasibility and performance based on the preliminary experimental results from our ongoing research work.

1. Introduction

Aerial image database is an essential infrastructure for various services using GIS (Geographic Information Systems), such as geographic analysis, disaster relief, rescue, and city planning. In addition to those governmental, business, and professional uses, it is nowadays also available to public through Web-based map services such as Google Map.

Aerial image discovery is one of the key technologies indispensable in various services using GIS, in which a match is to be identified in a large aerial photo image (file image) database with a given aerial photo (query image), with unknown location, orientation, and scale.

In aerial image in city areas, we see mostly buildings, houses, and streets, and those elements have more or less similar appearances. On the other hand, images taken in non-urban areas have more varied and colorful looking with farmlands, woods, mountains, and rivers. Image discovery task is, therefore, more challenging in city areas with this higher degree of homogeneity compared to non-urban areas.

For macroscopic analysis of the ground structure, such as rivers, mountains, and railways, we need aerial images with lower magnitude, which we will call as "high-view" images in this paper as they are from a camera high up in the air. On the other hand, for microscopic data usage, such as street or building analysis, we need images with higher magnitude, which we will call "low-view" images as they are from a camera closer to the surface. We generally store these file images of different scales and magnitudes covering the same area in our database.

When we are to make a query of a given aerial image, we need to search through file images of different scales, because the scale of the query image to be located is unknown, which inevitably makes each query computation highly intensive.

Although there have been various research attempts made in this area, the performance of automated discovery is not sufficiently high. One reason is this database size problem. When we have to find a match with a query in the database, we have to look through the multi-layers consisting of different resolutions, each consisting of large number of data covering multiple areas.

For example, let us think of a case where we already know that the query photo is from somewhere in West Tokyo, the Capital of Japan. In our proposed system which we will describe later, the file database to cover this area consists of 5 layers of different magnitude, and is calculated to contain 450 images in the highest view layer, down to 4800 images in the lowest view layer, totaling 8950 images.

Thus, it is practically impossible to perform fine matching with all images in all layers. To solve this complexity problem and accelerate the discovery, we propose to introduce a pre-selection mechanism to select subset of candidates for fine matching, to reduce the number of fine matching operations.

In this paper, we approach this query efficiency problem to reduce computation time in the search, and propose a mechanism and a system for attaining timely discovery in the large, multi-layered aerial city image database.

This paper outlines the problems, our proposed approach, and presents the preliminary experimental results using publicly available datasets from our ongoing research work.

2. Related works

Research related to image discovery so far mostly focuses on content based approaches using color, texture and shape features. Liu Zhen and Zhou Shu-qiu have proposed a method using color information and texture features obtained from Fourier transform in image retrieval by using colors and texture features in images [1]. The system can output several results similar to the input images.

In aerial image discovery field, Antonios Gasteratos et al. proposed a fuzzy aerial image retrieval system based on texture energy combined with color in an intelligent system for aerial image retrieval and classification [2]. Their experimental results show color information plays dominant role in the image retrieval process.

But aerial image match is easily affected by scale, rotation, noise and illumination, Color and texture features lack sufficient content expression ability for aerial images. The streets and buildings in different areas may have same color or texture

1 Graduate School of Computer and Information Sciences, Hosei University

histogram. In aerial image discovery, there are so many similarity images, if we want to find which area is the input image belonged to, color and texture feature is not enough to discover result accurately. Yan Ke et al. who proposed PCA-SIFT to reduce the calculation, apply PCA-SIFT to build a sub-image retrieval system [3]. They acclaimed that this system can satisfy the requirements of high recall and precision in efficient near-duplicate detection and sub-image retrieval. According to Image localization in satellite imagery with feature-based indexing, Changchang Wu et al. presented a method to do query image localization which has unknown scales and rotations [4]. The paper applied visual word combined SIFT features to make localization and tried GPU-based SIFT implementation to speed up. But the recall of their experiments result using Google color maps is only 13%.

In addition, most of the previous works have emphasis on images in non-urban areas. Thus, we need to establish a method to attain match accuracy, high degree of robustness, and computational speed, especially for city image database.

3. Proposed approach

3.1 Overview of the proposed system

Our system tries to find the best match of a given query image in the file database covering the target area, and returns the location corresponding to the query image on the map. Our primary challenge is to minimize the average time necessary in the image query.

Figure 1 shows the basic flow of the query operation. The query proceeds in following three stages:

1. Pre-selection --- pre-select file candidates (database subset) for macro matching
2. Macro matching --- Compare preprocessed query image with pre-selected subset
3. Fine matching --- Match the query image with macro matching results (only when necessary)

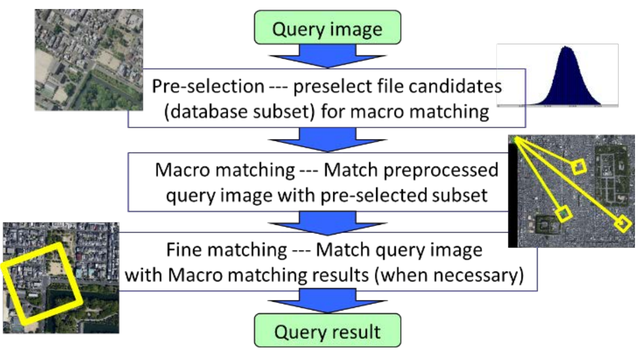

Figure 1: Flow diagram of image query

We construct our file database covering the area as multi-layered structure, each layer containing images of the same magnitude. Figure 2 shows a conceptual illustration of the aerial image database consisting of multi-scale layers.

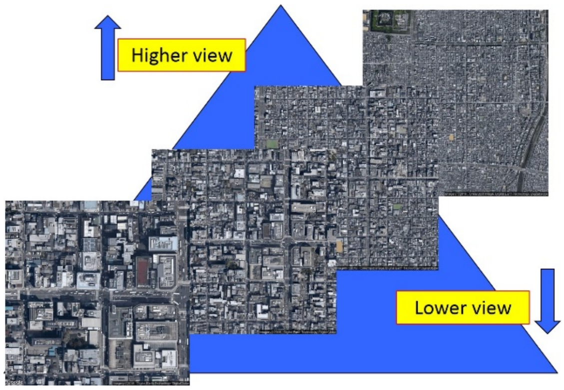

Figure 2: Conceptual illustration of aerial image database consisting of multi-scale layers

For each image in the database, the following profile data of the image are pre-computed and stored associated with the image:

(1) Frequency spectrum
(2) Color spectrum
(3) Surface structure descriptions
(4) Extracted key point features
(5) Actual coordinate of the image (latitude and longitude)

Frequency spectrum (1) is the frequency information mainly extracted from straight lines in the aerial photo, such as streets, railways, and large buildings. Color spectrum (2) is color distribution information of the aerial photo, which is typically different in cases of city areas and rural or mountainous areas. Surface structure descriptions (3) include information about the position and attributes of rivers, streets, railways, buildings and other structural elements on the surface.

Extracted key point features (4) are the features that are computed by image interest point detectors and descriptors that are used in the macro and fine matching stage. Actual coordinate of the image (5) is the latitude and longitude values of the photo, which is supposed to indicate the "answer" location corresponding to the query image on the map.

Figure 3 shows the block diagram of the proposed system configuration.

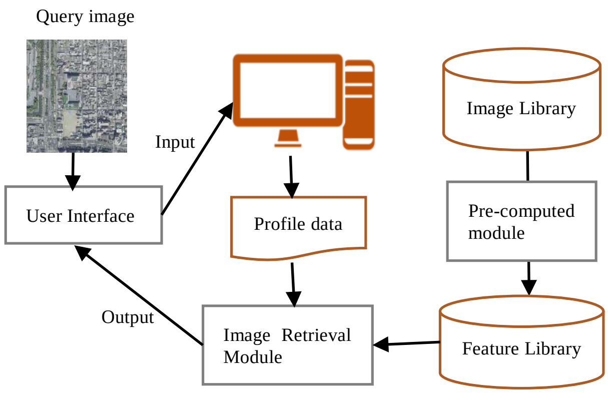

Figure 3: Block diagram of the proposed system configuration

3.2 Flow of image query

As we have seen in Figure 1, following is the basic flow of the query operation.

Given a query image, first we compute its profile data corresponding to the above (1)—(4) from the query image. Using the computed profile data (1)—(3), we first conduct pre-selection, the purpose of which is to pre-select candidates for the matching in the later stages. We will discuss the details of pre-selection in the next subsection.

Then we carry out what we call macro matching, to find a narrower candidate set for more detailed matching, after which fine matching follows, if necessary.

3.3 Pre-selection based on query profiles

The purpose of pre-selection is to select subset of candidates to be processed in the macro matching so that the matching computation time be minimized. In pre-selection, file image profiles are compared with that of the query image one by one, and the file image is "accepted" if its profile similarity is greater than the pre-fixed threshold [5]. Figure 4 shows the block diagram of the pre-selection scheme.

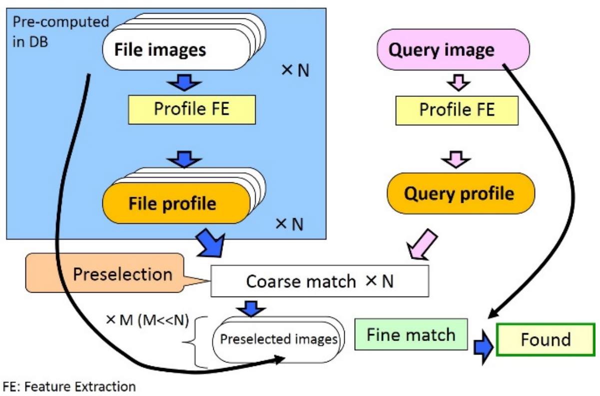

Figure 4: Block diagram of the pre-selection scheme

3.4 Macro and fine matching

In macro matching, the query image is compared with images in the pre-selected candidate set, which have been "accepted" in the pre-selection, one by one. To accelerate the query, we sort the candidate list in the order of similarity, and conduct matching in this order from the most similar ones.

In the macro matching, we first compute the corresponding interest feature points in the query image, then compare them with the pre-computed interest feature points of the file images. In comparison, we verify each of corresponding point pair combinations with geometric consistency between the two. Finally, we calculate the matching score to decide if the images match or not.

The query aerial image is generally taken at a different time and in a different way from the file images of the same location in the database, thus there is brightness change due to time difference and camera view variation. To absorb these variations, we use the image matching technique based on local image descriptors to achieve robust identification against geometric transformation between two images.

The image matching based on local image descriptors is widely used in various applications in computer vision. Typically, extensive object identification studies have been done using SIFT (Scale-Invariant Feature Transform) descriptor [6] because its features are invariant to scale, rotation, and intensity. Due to the high computational cost of SIFT, SIFT-like descriptors have been proposed that compute local features faster, such as SURF (Speeded Up Robust Features) [7], BRISK (Binary Robust Invariant Scalable Keypoints) [8], ORB (Oriented-BRIEF; Binary Robust Independent Elementary Features) [9], KAZE [10], and AKAZE (Accelerated KAZE) [11].

The corresponding point pairs obtained from the interest point detector may include some outliers (the data points that do not fit the model). Hence, we verify each corresponding point pair with geometric consistency between the query image and the file image. For this purpose, we use RANSAC (RANdom SAmple Consensus) algorithm [12] to verify using homography whether corresponding point pair is an inlier (data point that can be described by the model) or not.

In macro matching, we pre-process the query image so that the image can be compared with images in a higher view (preferably the top) layer, because there are smaller number of images in higher-view layers. Typical pre-processing is down-sampling, to shrink the image data size in order to speed up the comparison.

If there are multiple match results found in the macro matching, we conduct fine matching, in which we compared the original (non-down sampled) query image with images in a lower view layer one by one in the descending order of the similarity score obtained in macro matching.

4. Prototype implementation

4.1 Prototype system

To prove feasibility and effectiveness of our approach, we implemented a prototype and conducted performance evaluation on this prototype.

Our prototype has two (2) aerial photo database sets, one for Kyoto and the other for West part of Tokyo. Each consists of 5 layers of different scales, and two kinds of images, one is an aerial image, and the other is extracted surface structure description layout map.

Table 1 summarizes the number of images in each layer set.

Table 1: Number of images in each layer set

	L1 (high view)	L2	L3	L5 (Low view)
Kyoto	120	360	918	2100
West Tokyo	450	1000	2700	4800

4.2 File data and query test data collection

To be used as as file data in image database and also for evaluating the performance, we need a large scale collections of pairs of aerial photos and corresponding structural feature labels.

Although there are several publicly available aerial photo databases, they are collected abroad, and most of them consist of photos taken in rural areas, thus there is not sufficient dataset especially for cities.

For this reason, we developed an automatic data collection system which collects and downloads publicly available data via WebAPI. We used Google Map public WebAPI to collect sample data for high view and low view satellite land surface image as shown in Figure 5 below.

Figure 5: High View Image (left) and Low View Image (right) from Google Map WebAPI

In addition, to evaluate the performance robustness against the variety of image sources, we collected aerial photos from the Geospatial Information Authority of Japan (GSI) database. Figure 6 is an example of this GSI high view image.

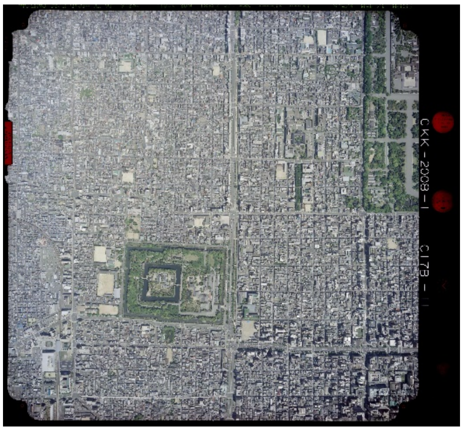

Figure 6: GSI High View Image

GSI provides images in several timestamp and we picked up some from collections starting from 2005. Image in Figure 6 was taken from Kyoto area around 2008. Based on this high view image, we scale, clip and add some rotation in a random degree to produce the low view test image that has similar size with an image obtained from Google WebAPI. We also labeled with its latitude and longitude of the location, to verify the query match correctness.

Figure 7 shows example of query data used in the experiments.

Figure 7: Examples of query data:
Google Map Data (Left) and GSI Map Data (Right)

5. Experiments and the results

5.1 Frequency based pre-selection

As we discussed in Section 3.3, the purpose of pre-selection is to pre-select subset of candidates to be processed in the macro matching, so that the computation time of matching be reduced. In order to prove the feasibility and effectiveness of the pre-selection, we implemented frequency-based pre-selection and conducted its performance evaluation.

Basic idea is that aerial views with different scale have different frequency distributions. We assumed that the larger the image magnitude is (lower view), the frequency distributes more on the lower frequency range, because the average width and average length between streets are larger. On the other hand, the smaller the image magnitude is (higher view), the frequency distribute more on the higher frequency range, because the average width and average length between streets are smaller. Figure 8 illustrates its basic concept; the left hand side is lower view case and the right hand side is higher view case, shown with the extracted streets (upper row) in the corresponding image (lower row).

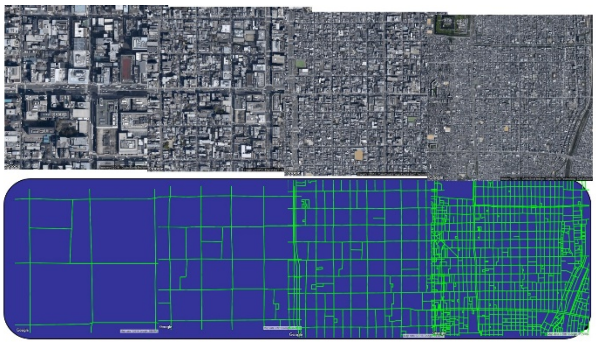

Figure 8: Difference of scales and street distributions

We make a hypothesis that generally the frequency distribution among images in the same magnitude layer is consistent, and the frequency distribution pattern differ from layer to layer. If this hypothesis is true, we can use the frequency distribution peak information of the query image as input to pre-selection, because we can select those file images having the similar frequency distribution as the query image, thus reducing the size of pre-selected candidate subset. Figure 9 illustrates this scheme.

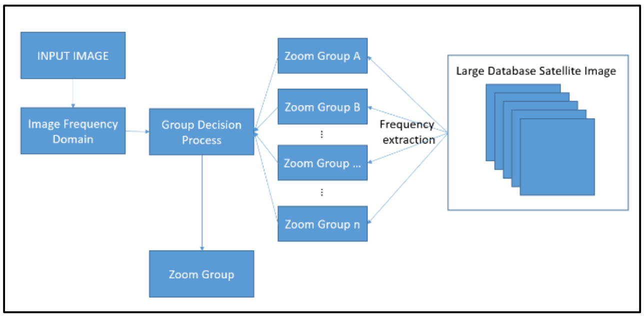

Figure 9: Pre-selection scheme based on frequency spectrum distribution

To prove the validity of this hypothesis, we conducted an evaluation. We applied Fourier Transform to obtain the frequency spectrum of an aerial image. Using FFT, we extract the frequency spectrum from the input aerial images in respective layers and compared their distribution.

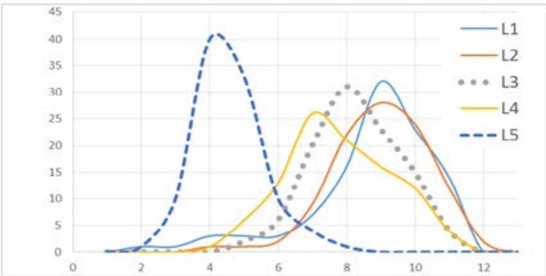

Figure 10: Relative frequency histogram of aerial images from different scale layers

Based on Figure 10, we can observe that samples from higher magnitude layer, in this case is L5, likely to appear in lower range frequency and vice versa, as we expected. With this result, the pre-selection can be expected to work effectively for selectively filtering candidate file images having similar frequency range, and will reduce computation time by eliminating unnecessary matching with images with inherently different spectrum profiles.

Through this experiment, we have proved that use of frequency spectrum profile is effective for pre-selection.

5.2 Macro matching performance

To prove feasibility and effectiveness of the proposed fine matching mechanism, and also to observe the trade-off tendency between matching performance (accuracy) and processing speed, we conducted a test to observe the matcher behavior with different down-sampling size (one side in pixels) of query images.

We prepared 100 query images by clipping randomly selected L5 (low-view) layer aerial images from Kyoto area. Then each is further modified to have random scale variation from 0.5 to 1.5 times the original scale, rotation at a random angle, and finally cropped to a square image. In the evaluation test, these 100 query test images were matched with all the file images in a L1 (high-view) layer using SIFT [6] as interest point detector.

In this experiment, we used OpenCV 3.1 on Arch-Linux ver. 4.5.1-1 on a personal computer with Intel Core i7-4710MQ CPU (2.5 GHz clock speed) and 4 GB memory.

Table 2 summarizes the results of this matching trial with different query image sizes (one side in pixels). "Time" means the computation time required to perform high-view of one query with all 120 L1 datasets (Graph plot is in Figure 12 for computation time). FRR and FAR are calculated as follows to measure matching accuracy:

FRR = ratio of failed queries in 100 cases

FAR = ratio of false match cases in 100 x 120 comparisons

Table 2: Macro matching performance using SIFT for different query image sizes

Query img size	FRR	FAR	TIME (sec)
400	0	0	156.42
300	0	0	88.44
200	0	0	42.30
150	0	0	27.22
125	0	0	19.19
100	5%	0	10.33
75	34%	0	8.25

It is observed that we can reduce the macro matching processing time by reducing the query image size by simple down sampling, without any penalty of accuracy degradation, if the image size is over 100.

Figure 11 (left) is an example of down-sampled macro match input image (100 pixel case), and Figure 11 (right) is an example illustrating a successful match. As you can see, the macro matcher successfully finds out a correct position from a considerably large, busy-looking, high-view image consisting of structures of similar appearances, using only a small, low quality image clip. This demonstrates the robustness of our approach against rotation and scale changes.

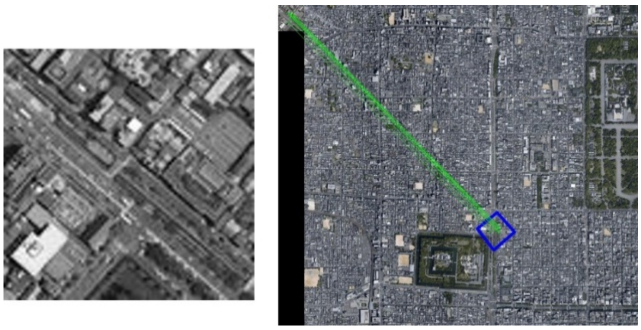

Figure 11: Macro match input (left) and a result (right); interest point correspondences between the down-sampled query image (upper left) and high view file image (right) are shown by green lines, and the location in the high-view image found by the macro matcher is shown by a blue rectangle

5.3 Macro match algorithm comparison

As we have discussed in Section 3.4, several feature detectors, descriptors, and matchers have been proposed and are used besides SIFT. Among them, we tried AKAZE (Accelerated KAZE) [11] in OpenCV version 3 for a comparison. Table 3 summarizes the results of a test using AKAZE instead of SIFT.

Table 3: Macro matching performance using AKAZE for different query image sizes

Query img size	FRR	FAR	AKAZE (sec)
400	0%	66.658%	38.42
300	9%	18.833%	19.90
200	38%	0.333%	6.89
100	78%	0.000%	0.60

Figure 12 shows how average matching time (Y-axis) changes with the down sample size of the query image (X-axis), in cases of SIFT and AKAZE. As we can see, AKAZE runs much faster than SIFT, approximately 5 times faster. On the other hand, AKAZE matching performance is much lower than SIFT.

Figure 13 is the graph plot of FRR and FAR in case of AKAZE (Note that FRR and FAR errors are almost zero for all image sizes for SIFT).

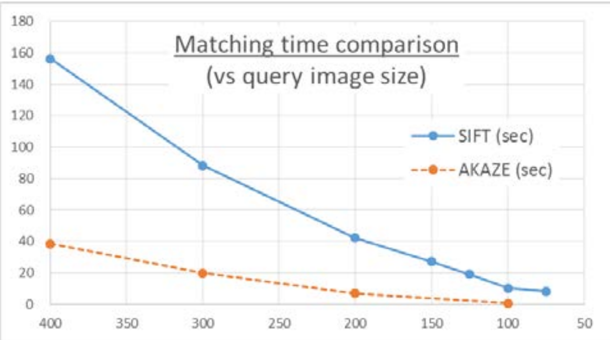

Figure 12: Macro matching computation time for different query image sizes (SIFT vs AKAZE)

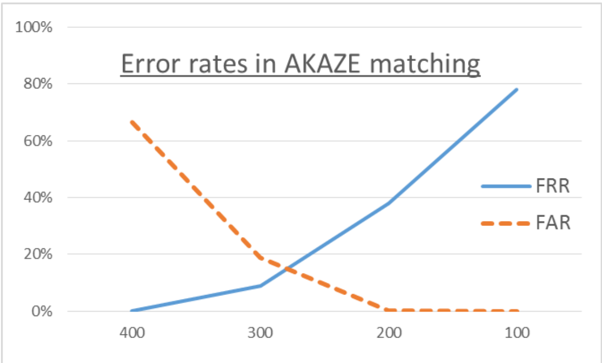

Figure 13: Macro matching error rates (FRR and FAR) using AKAZE for different query image sizes

5.4 Macro matching robustness against query image source variation

In the evaluations so far, query aerial images from Google map data are searched in the Google map file database. Although they are not exactly the same data, they are basically from the same source, and thus likely to have higher degree of similarity with each other.

To see the robustness of our approach, we also conducted experiments using another query dataset generated from GSI database (see Section 4.2). Here, query aerial images generated using GSI data are searched in the Google map file database, same as the previous experiments.

In Table 4, we first compare the performance results on two sets of data when we use SIFT algorithm. Here, the error rate (FRR) is by far larger in GSI database case than Google map case (Gmap), where the processing time is generally comparable.

Table 4: Macro matching performance using SIFT: Gmap vs GSI query test data (against Gmap DB)

Query img size	Gmap		GSI	
	FRR	Time (sec)	FRR	Time (sec)
400	0	156.42	62.0%	230.45
300	0	88.44	65.0%	130.81
200	0	42.30	77.0%	53.48

Table 5 is for the case of AKAZE instead of SIFT.

Although the accuracy is not as high as the Google map query case, taking the fact into consideration that we have not done sufficient optimization for GSI query case, we can conclude that our approach works on different query images as well, demonstrating the possible applicability of our proposed approach for more general query cases with images from different sources.

Table 5: Macro matching performance using AKAZE: Gmap vs GSI query test data (against Gmap DB)

Query img size	Gmap			GSI		
	FRR	FAR	Time (sec)	FRR	FAR	Time (sec)
400	0%	66.658%	38.42	7%	27%	34.04
300	9%	18.833%	19.90	61%	3%	15.81
200	38%	0.333%	6.89	97%	0%	4.02

6. Discussion

As we have seen in the previous section, we have proved that use of frequency spectrum profile is effective in pre-selection.

Based on the experimental results on section 5, we can conclude that by using pre-selection and macro matching, we can improve the computation speed without noticeable degradation of discovery accuracy as tradeoff. Pre-selection based on frequency spectrum profiles obtained from Fourier Transform greatly contributes in computation time reduction by quantifying image magnitude so we can minimalize unnecessary comparisons.

Based on the findings in the previous section, following is an example of the presumed operational scenario when a query aerial photo in Kyoto with unknown location, orientation, and scale is given, provided that we perform the macro match with 120 pieces of L1 dataset using SIFT in our proposed architecture and method:

1. Apply FFT to obtain frequency spectrum profile to estimate its magnitude, for pre-selecting candidate images based on correspondence between frequency distribution and image magnitude
2. Down-sample it to appropriate size, so that FRR free macro

match is possible --- For example, if the query image is estimated to be in L5 layer, the appropriate one side size is 125 pixels to compare with 120 pieces of L1 dataset, as observed from Table 2.
3. Repeat macro match till it finds a match
4. Note that no fine match process is expected to be necessary because there are likely to be no false accepts in macro match, i.e. macro match can come up with a unique query answer

The expected total computation time for the above is approximately 10 sec (assuming that time for step 1 and 2 is negligible and the average match count is 60, half of the L1 dataset size)

Let us compare this with the most naïve scenario of a brute-force search. Suppose we are to compare the query image with all the lowest view (L5) images in the database (2100 pieces in Kyoto case), the expected total computation would take:

$1.1 \times 2100 / 2 = 1155$ sec = 19.3 min

Therefore, with the introduction of our pre-selection and macro matching scheme, we could reduce the average discovery time from 1155 sec to 10 sec in the Kyoto case.

On another scenario, if we are to allow 5% query error (meaning the system cannot find the match once in 20 query trials), the total time is expected to be 5 sec (half of 10.33 sec).

On the other hand, if we are to allow 9% query error and to use AKAZE instead, the appropriate down-samples size is 300. In this case, as there is 18.8% FAR, we have to perform low-view fine match in these false accept cases. Supposing the match is expected be found averagely after 918/2 comparisons with L3 images (from our tests, we found that each AKAZE match takes 1.1 sec if the query image is not down-sampled), the expected total computation time in this scenario is:

$19.9+1.1 \times 918/2 \times 18.8\% = 115$ sec

Thus, with this performance, we cannot employ AKAZE instead of SIFT, unless there is considerable improvements.

7. Conclusion and future work

We have addressed the task of aerial city image discovery, the difficulty of which is the database size, because we have to look through the multi-layers consisting of different resolutions. To realize reasonably efficient query, we propose an approach with pre-selection and macro matching prior to fine matching. The pre-selection works to select subset of candidates to be processed in the macro matching. In macro matching, we pre-process the query image so that the image can be compared with images in a higher view (preferably the top) layer, because there are smaller number of images in higher-view layers.

In this paper, we outlined our proposed approach, the system configuration and how the discovery proceeds. We also presented the preliminary experimental results using publicly available datasets from our ongoing research work.

We have shown based on experiments that our approach can reduce the average discovery time from 1153 seconds in a conventional, naïve scenario to 10 seconds in the Kyoto case.

Summary of findings

Following is the summary of findings that are possibly the contribution to the research community:

1. We proved the effectiveness of pre-selection approach for candidate subset reduction.
2. We tried (rather straightforward) use of general local interest point descriptors for aerial image matching and showed its effectiveness.
3. We proposed and tried macro (high-view layer) matching approach, and showed the preliminary performance test results, with its trade-off of error rates and speed. Our approach can accelerate the discovery as much as 110 times compared to a conventional, brute-force query approach without pre-selection and macro matching.

Future work

As for pre-selection, we plan to add and evaluate various profile features, such as color spectrum and surface structure information. For reliable extraction of structural landmarks such as streets and buildings, we are trying deep machine learning approach using Convolutional Neural Network (CNN), trained with existing map and aerial photo dataset. One of its difficulties is how to detect roads accurately especially in urban areas because some high-rise structures such as tall buildings often occludes and/or cast shadows on other things on the surface such as streets, which makes the automated detection less reliable.

As for macro matching, we plan to try and evaluate other local interest point descriptors, parametric optimizations, and additional query image preprocessing techniques for better matching performance. For example, we observed that AKAZE, which is more than 5 times faster than SIFT, performs pretty well in accuracy on query images with comparatively higher quality. We believe therefore the hybrid use of different descriptors and matchers will enhance the overall performance. We plan to implement hybridization of multiple matchers based on the image quality assessment in the pre-selection stage.

To prove general robustness of our approach, we also plan to try experiments using various aerial images from other big cities like Tokyo, as well as other middle-sized cities, towns, and villages with more woods and mountainous areas.

We believe that GPU (Graphics Processing Unit), with its highly parallel architecture, is particularly well suited for the macro and fine matching in our approach, since they are basically multi-dimensional vector computations. We hope to demonstrate accelerated performance results through the use of GPGPU (General-purpose computing on graphics processing units) as well.

References

[1] Liuzhen, ZhouShuqiu. Image retrieval by using of colors and texture features in images[J]. Computer Engineering and Desig n, 2007, 28(20): 4952-4954.

[2] A. Gasteratos, P. Zafeiridis, I. Andreadis, An Intelligent Syste-m for Aerial Image Retrieval and Classification, in: G.A. Vou-ros, T. Panayiotopoulos (Eds.), Lecture Notes in Artificial Intel

ligence, vol.3025, Springer, Berlin, Heidelberg,2004, pp. 63–71.
[3] Ke Y, Sukthankar R, Huston L. Efficient Near-Duplicate Detec tion and Sub-image Retrieval [J]. ACM Multimedia, 2004:869--876.
[4] Wu C, Fraundorfer F, Frahm J M, et al. Image Localization in satellite imagery with feature-based indexing[J]. The Internat ional Archives of the Photogrammetry, Remote Sensing and Sp atial Information Sciences, 2008, XXXVII (B3a).
[5] R. McAfee. Coarse matching. Econometrica 70(5), 2025—2034 (2002)
[6] D. G. Lowe. Distinctive image features from scale-invariant k-eypoints. Intl. J. of Computer Vision, 60(2):91–110, 2004.
[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Spee ded up robust features. Computer Vision and Image Understan ding, 110(3):346–359, 2008.
[8] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In IEEE Intl. Conf. on Co mputer Vision (ICCV), pages 2548–2555, 2011.
[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In IEEE Intl. Conf. on Computer Vision (ICCV), pages 2564–2571, 2011.
[10] P. F. Alcantarilla, A. Bartoli, and A. J. Davison. KAZE featu res. In Eur. Conf. on Computer Vision (ECCV), pages 214–22 7, 2012.
[11] P. F. Alcantarilla, J. Nuevo, A. Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces [J]. IEEE Tr -ans. Patt. Anal. Mach. Intell, 2011, 34(7): 1281-1298.
[12] Fischler M A, Bolles R C. Random sample consensus: a par adigm for model fitting with applications to image analysis an d automated cartography [J]. Communications of the ACM, 19 81, 24(6): 381-395.