IMAGE LOCALIZATION ON SATELLITE MAP WITH A NEW FEATURE-BASED INDEXING

N.N. for anonymity

Commission III/3

KEY WORDS: orthogonal steatite imagery , image localization, SIFT, map indexing, content-based image retrieval, vocabulary tree, visual word

ABSTRACT:

This paper presents a method to index ortho-map databases with image-based features and search a map database for regions that match query images of unknown scales and rotations. The proposed method uses image-based features on map to index the 2D locations. Image feature extractor normally generates features with location, orientation, shape, and descriptor of normalized image patch. In a map database, the geographical location, orientation and shape of image features can be recovered with a reasonable local planarity assumptions. Based on existing recognition techniques that use descriptor-based visual words, the proposed method extends the visual word with geographical dimensions, and uses the extended visual words to index a 2D location grid of map. An indexing-friendly scoring system is defined to measure the similarity of images, and the virtual database images that are made up of sets of unit tiles are also specifically handled. The implemented scoring algorithm can efficiently give the matching scores between a query image and all possible database images. Upon searching a new orthogonal image, a set of scaling and rotations are first selected, and the visual words are transformed and matched against the database. The best locations along with scales and rotations are determined from the set of query results of of the transformed visual words. Experiments show a high success rate and high speed in searching map databases for aerial images from different datasets.

1 INTRODUCTION

Nowadays, satellite imagery has become an important part of our information source. The amount of high resolution satellite imagery is growing rapidly, and many of them are now available to public through various map services, such as Google Maps, etc. In this paper, we are interested in the problem of searching for aerial images (with unknown scales and rotations) in a map database. Given a particular aerial image, we proposed a method to find the locations of similar map data along with the relative scales and rotations, and provide a confidence measurement for the similarity. Temporal changes, repetitive structures, varying illumination condition, and varying cameras lead to appearance variances that make the problem very difficult. With a proposed image feature-based indexing and searching, Our approach efficiently handles the challenges from the complexity and large scale of satellite imagery.

One of the main contributions of this paper is a proposed feature indexing for geo-located features on map. Our method adds geometric dimensions to existing visual word (Sivic and Zisserman, 2003, Nister and Stewenius, 2006), and uses the extended visual words on map to index 2D location grids. Unlike the general image retrieval problem, geographical size and geographical rotation of features in map database can be recovered. Visual words with sizes and rotations differentiate features at different scales and different orientations, which leads to an more efficient indexing and retrieval system.

A big difference between map database and image database of independent images is that unit images in map database are geographically connected. Satellite imagery can be viewed as a set of unit tile images and those tile images in map database can not be treated independently. Correspondingly, query image of different sizes in map database need to take varying number of unit images as group to match. This paper introduces a scoring scheme that can be applied images that are composed of a set of unit database images.

The remainder of the paper is organized as follows: Related work is discussed in Section 2. Afterwards Section 3 introduces our new visual word and the associated feature indexing system. An efficient image localization based on the indexing system is given in Section 4. Experimental results are presented in Section 5. Conclusions and future words are discussed in Section 6.

2 RELATED WORK

Our localization problem is essentially a content-based image retrieval problem. Compared with the problem of searching for images in image databases our goal is to look for a small part in a big image. In recent years, there has been an extensive investigation in local invariant image features, based on which many recognition systems have been developed. Lowe's Scale Invariant Feature Transform(SIFT) extracts distinctive similarity-invariant features from the scale space of images, and describes image patches of varying sizes by SIFT descriptors, which is a 128D vector of gradient histogram. Lowe also demonstrates a promising recognition system built on SIFT feature database. SIFT descriptor is also exhibiting its power of describing image patches in other types of local region features (Mikolajczyk et al., 2005). In this paper, we will use the feature-based approach to build an image-based localization system for map databases.

The analogy to text retrieval by indexing images with visual words from features in (Sivic and Zisserman, 2003) and (Nister and Stewenius, 2006) is crucial for achieving scalable recognition. By training from feature databases, a set of visual words are selected as vocabulary to represent the image as documents. Each visual word represents a small portion of the entire feature space, and the features that lie in its portion. By indexing images with their visual words, an inverted file is generated for each visual word to give the set of images that contain the visual words. As a result, feature matching becomes an efficient lookup of the inverted files of visual words. It is then easy to evaluate the matching scores for database images by combining the scores from the visual words

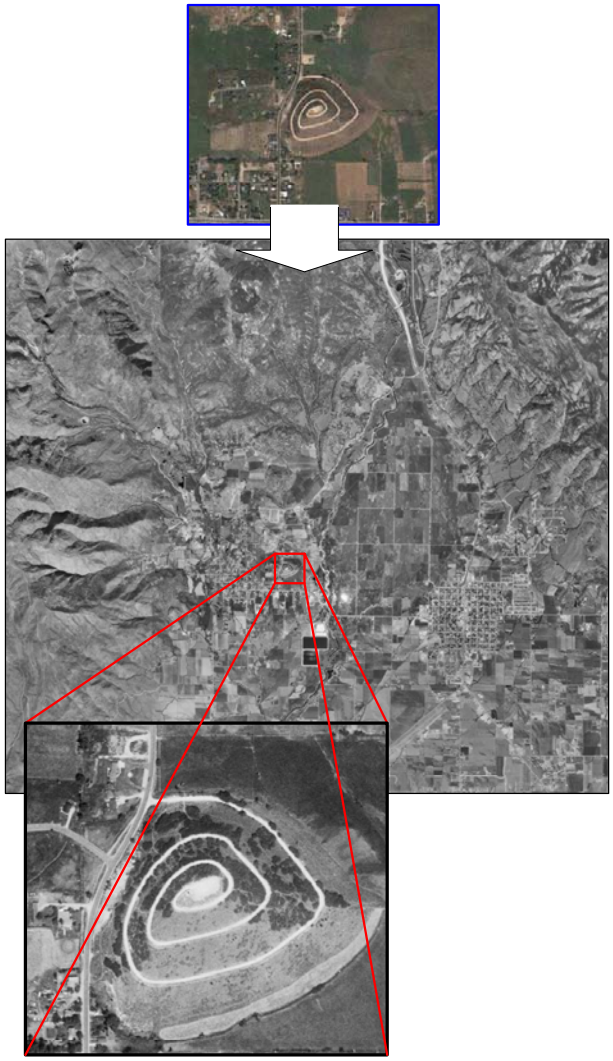

Figure 1: Illustration of the image localization problem. The background is a zoomed view of part of the satellite map in Park City, UT, USA, and the smaller image on the left bottom and the small red box in the map gives a detailed view of a small area. The color image at the top is an example of what we want to search in the map.

they share with query images. In this paper, we will take advantage of the simplicity of the camera pose in satellite imagery and build a more efficiently feature indexing system with an improved visual word for image localization.

The vocabulary tree method (Nister and Stewenius, 2006) is used in this paper to employ the SIFT descriptors for indexing. With a tree of feature cluster centers, a feature can follow the path of its closest centers to reach a unique leaf of the tree. High dimensional SIFT descriptors are then quantized into one dimensional integers by assigning the integer ID of the leaf to the feature. This hierarchical scheme is very important for selecting a large vocabulary and for the fast quantization of new features descriptors. On one hand, it is hard to select a large number of cluster centers from large number of input image features(e.g. get 10000 centers from 100000 features.) On the other hand, computation spent at each level in quantization is linear to the branching factor of the tree and more levels give higher quantization speed when total number of leaves are the same. The hierarchical scheme makes the method scalable for large vocabulary and large number of features, they Nister and Stewenius demonstrated efficient high quality recognition.

3 MAP DATABASE INDEXING

A good indexing of features needs a vocabulary of discriminative visual words. Although quantization through vocabulary tree makes SIFT descriptors easily indexable, pure descriptor-based indexing still results in a lot of ambiguity. The reason for this is that the feature descriptor itself does not carry much information about the size, orientation and shape of the real 3D geometry. Invariant features detectors can transform different image patches of a same 3D structure to similar normalized patch. On the other hand, it may also generate similar normalized patches for different 3D surfaces. Accordingly, a same visual word is very likely to represent a set of features from different 3D structures with different projective transformations.

In the context of satellite imagery, local 3D structure corresponding to image features can be approximately recovered by assuming the corresponding part of the ground to be plane. This assumption is reasonable because the variation of elevation is much smaller than its distance to the camera. Then every image feature can be seen as a geographic image feature. Described by the geographical size and orientation which are discriminative properties of image features, they resolve the ambiguity between many different 3d structures on ground. Therefore, the geographical size and orientation can be used along with the descriptor for indexing. Additionally the geographic properties of affine covariant regions can also be recovered, the proposed method can be extended to use these features, which is not done in this paper that only uses the SIFT features, the size and the orientation of the features.

Satellite imagery can be ortho-rectified with an affine transformation because the cameras are very far away from ground. Then, the only remaining unknown image transform between images taken by different camera or camera at different time is approximately a 2D similarity transformation. Considering the similarity transformations between our query and database, it is correct to choose similarity-invariant SIFT feature because they are highly repeatable under similarity transformations.

The visual word stored in our database is a triplet consisting of SIFT descriptor, geographical size and geographical orientation. Feature descriptors are quantized with a vocabulary tree that is trained from millions of features extracted from the satellite map. Then the geographical orientation and the logarithm of geographical size are also discretized to integers for convenient indexing. Hence, each visual word is a 3D integer vector. Like other indexing-based techniques, inverted files are constructed for the visual words of each map database.

In detail, for a feature $(x,\ y,\ \zeta,\ \theta,\ sift)$, where $(x,\ y)$ denotes its UTM coordinate in meter, ζ the geographical size in meter, θ the angle in the ground plane in degrees, and $sift$ the SIFT descriptor, it is transformed to an indexing pointer as below:

$$(\mathrm{f_s}\lfloor \log_2 \zeta \rfloor,\ \lfloor \theta * N_\theta/360 \rfloor,\ \mathrm{f_v}(sift)) \Rightarrow (\lfloor x/W_t \rfloor,\ \lfloor y/H_t \rfloor)$$

where function $\mathrm{f_s}$ maps the logarithm of feature size to a smaller set of N_s integers, N_θ is the number of rotation to use, function $\mathrm{f_v}$ uses a pre-trained vocabulary tree to quantize 128D SIFT descriptors to integers, $W_t \times H_t$ is the tile size for the database. $(\mathrm{f_s}\lfloor \log_2 \zeta \rfloor,\ \lfloor \theta * N_\theta/360 \rfloor,\ \mathrm{f_v}(sift))$ on the left is the visual word from feature, and on the right side$(\lfloor x/W_t \rfloor,\ \lfloor y/H_t \rfloor)$ is a location (visual document of a $W_t \times H_t$ tile) to index. Our

proposed method is indexing a set of 2D location instead of dependent images in other image retrieval problems. The n database are stored on disk as tile images with associated f tures corresponding each indexed locations. Tile size 200×2 is chosen in our experiment because the original image data downloaded from TerraUSA (MicroSoft, n.d.) server are or nized as 200×200 tiles.

An advantage of this new visual word is that scaling and rotat can be applied to the visual words in vocabulary, and the tra formed visual words will still be in the vocabulary. Simila scaling and rotation can be applied to any query image to g new group of visual words for query. For convenience, give function g of some scaling plus some rotation, the transformat of visual word of t is denoted as $g(t)$, and the transformation visual word group of $\{t\}$ as $\{g(t)\}$.

Given some scaling and rotation threshold, the set of visual words that have similar scale, similar rotation and same descriptor with a visual word is easily recoverable. Then the inverted files of those similar visual words can also be used in matching. This enable us to match query images to database image with some transformation threshold. For convenience, given a threshold T of some scaling and some rotation, the set of visual words in range with a visual word t is called $T(t)$.

It is important to weight the visual words in the vocabulary so that visual words are treated differently. For example, if some features show up in all most all the locations, a small number of such feature won't be able to provide much information for localization, which means a small weight should be assigned. A standard IDF (inverse document frequency) weighting is used in this paper to weight visual words. Such an IDF weight for a visual word t_i is defined

$$w_i = \text{idf}_i = \log \frac{|XY|}{|xy : t_i \in d_{xy}|}$$

where $|XY|$ is the total number of locations to index and $|xy : t_i \in d_{xy}|$ is the number of locations that have this visual word. IDF is set to zero when the number of occurrence is zero. IDF weighting of the new visual words now gives different weights to features that have different scales and orientations. A a result, the large features now become more important than the smaller ones since there are less of them. This is also consistent with our intuition about the features. Without the utilization of the geographical information, the IDF weighing will lose a powerful dimension for discriminating features. However, due to the ambiguity of the real 3d structures, the proposed visual world will not be applicable to general image retrieval problem except for the cases where all 3D dense structure and relative sizes can be recovered.

Figure 2 demonstrates the change of IDF distribution when feature sizes are used in location indexing. In this proposed scheme, more visual words in the vocabulary have zero occurrences, which is more efficient for lookup of inverted files because more lookups are skipped. The proposed visual words also emphasize the importance of feature sizes.

4 MAP DATABASE SEARCH

This sections will explain the image localization algorithm with the proposed feature-based indexing. The definition of out scoring system and the associated implementation is first introduced, which is a main contribution of the paper. Specific handling of querying sets of database tiles are discussed afterwards. The

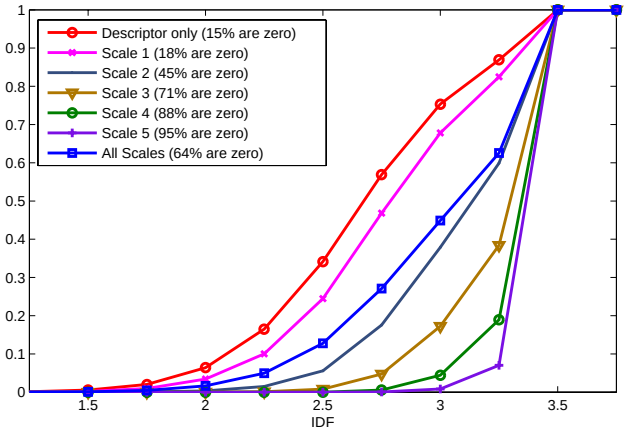

Figure 2: CDF plot of non-zero IDF for a database. In this experiment, 5 scale levels are used, rotation is ignored. The first red curve is the IDF for the descriptor-only visual word. The rest are the curves for the proposed visual words at five different scales and at all scales. first curve is the one of descriptor only case. With scales taken into account, much larger portion of visual words have zero occurrence in the database. visual words of larger scales will have more zeros occurrences and larger IDF weights because of less occurrence.

querying algorithm for recovering locations along with scales and rotations are given in the end.

Let the vocabulary be $\mathbb{N}$, the visual words at each location d is a multiset over the vocabulary as $d = < \mathbb{N}, n_d >$ where function n_d gives the number of occupance for each visual word. The correlation between any two multisets is defined as

$$\begin{aligned} \text{corr}(q, d) &= \sum_{i \in \mathbb{N}} n_q(i) n_d(i) w_i^2 \\ &= \sum_{i \in q} \sum_{j \in d} \rho(i, j) w_j^2 \end{aligned}$$

where $\rho(i, j)$ is a function that returns 1 when i and j are equal, and returns 0 otherwise. Please remember that multisetS may have more than one occurrences of one visual word, and $i \in q$ and $j \in d$ enumerates all of them.

The second equation above does not require counting of occurrences of visual words, and it leads to an easy implementation of correlation computation with inverted files as follows:

```
function ComputeCorrelation(q)
1 set corr(*, q) to zero
2 for each i ∈ q
3   for each d in the inverted file of i
4     corr(d, q) += w_i^2
```

When some threshold is provided, a set of visual words need to be checked for each visual word in the query document. The computation will be as follows:

```
function ComputeCorrelationWithThreshold(q, T)
1 set corr(*, q) to zero
2 for each i ∈ q
3   for each j ∈ T(i)
4     for each d in the inverted file of j
5       corr(d, q) += w_j^2
```

The final normalized matching score between any two multisets of visual words are defined as

$$\text{score}(q,\ d) = \frac{\text{corr}(q,\ d)}{\sqrt{\text{corr}(q,\ q)\text{corr}(d,\ d)}}$$

which measures confidence of two images being the same scene.

When querying an image q, its matching score $\text{score}(q,\ d)$ with all d in the database are computed as above, then the best matches can be obtained by comparing the scores to determine the largest ones. To realize efficient query, the self-correlation of database image$\text{corr}(d,\ d)$ is should be pre-computed when building the database. $\text{corr}(q,\ q)$, self-correlation of query image, only need to be computed once for each possible transformation on a query image, and it is constant for all database documents.

Our problem is to locate images with unknown scale and rotation in a map database. It needs to compute not only the best possible locations, but also the corresponding scales and rotations. Although the actual stored images in our map database are 200×200 tiles, it is not enough only computing the matching score between the query image and database tiles, because the corresponding size of the query image in the map is unknown. We need to match query images with any map images with some reasonable size of $W_t \times H_t$ tiles, where W_t and H_t are the number of tiles along x and y direction.

The correlation between any query image q and any tile group D can be obtained from the query's correlation with the tile units as

$$\text{corr}(q, D) = \sum_{d\ \in\ D} corr(q, d)$$

After computing the correlation of a query with all the database tiles, the correlation between the query and any tile groups can obtained by summing up the correlation of all the tiles in the group.

The self-correlation of the tile groups in the map database needs to be specifically handled. For any group of tiles $D = \{d\}$, its self-correlation is

$$\begin{aligned} \text{corr}(D,\ D) &= \sum_{i\ \in\ \mathbb{N}} n_D(i) n_D(i) w_i^2 \\ &= \sum_{i\ \in\ \mathbb{N}} w_i^2 (\sum_{d\ \in\ D} n_d(i))^2 \\ &= \sum_{d1\ \in\ D} \sum_{d2\ \in\ D} \text{corr}(d1, d2) \end{aligned}$$

The self-correlation of a tile group is the sum of the correlations of any two tiles in the tile group. Therefore, the correlations between any two tiles need to be pre-computed for fast computation of self-correlations.

To recover the scale and rotation, we first choose a set of reasonable geometric transformations with corresponding thresholds to cover the gaps between them. The corresponding map sizes of a query image for the transformations are then determined, and the matching scores between a query image and all possible tile groups are then computed. Finally, by finding the maximum matching scores, scale, and rotation are recovered along with the location. To avoid locations that are too close are chosen, each time when a maximum score is selected, its neighboring locations will be suppressed as non-maximum.

```
function Query(q)
1 Get the set of geometric transformation G to test
2 for each g ∈ G
3   Compute the tile group size W_g × H_g
4   Compute correlation of g(q) with database tiles
5   Compute the summed correlation for tile groups
6   Compute the self correlation for tile groups
7   Compute the matching scores
8 Select tile groups with the N_q largest scores
9 Verify 2D transformation for selected tile groups
```

The final optional step of matching is the geometric verification of some top scoring images as tile groups. Putative feature matches can be established form the inverted files of visual words. A simple histogram method is then used to get the inliers for 2D similarity transformation. Since each feature contains information about scale, rotation and translation, each putative match can establish a 2D similarity transformation. A histogram of number of supporting matches can be constructed, and the largest number corresponds to the best possible geometric transformation for this image. Then the best matches among all candidate images are the ones with largest number of inliers. Geometric verification not only filters out false positives, but also recovers more accurate scale, rotation, and location.

5 EXPERIMENTS

This section first explains how the feature extraction is adapted for map database, then talks about the database and query images that are used in the experiments. Experiment of querying map database for images that are from different datasets are presented.

To extract SIFT features for satellite map database, the huge images need to be divided into small pieces to run feature detector. However, It is infeasible to detect features for the entire map at once. Enough overlap between sub-images are very important for keeping the features that are close the boundary of division. Otherwise, features of large scale on the boundary will be lost. In our experiments, a 800 meter overlap is used. It means the feature detection can approximately keep all features of sizes up to 800. Additionally, GPU-based SIFT implementation is used to speed up the processing (Sinha et al., 2006, Wu, 2007). The fast processing with GPU is also another reason that we choose SIFT.

In our experiment, a gray-level satellite map of 16000 meters by 12000 meters in Park city of Utah is used in our experiments. The background image in Fig 1 is a small part of our database, where both mountains and cities are covered. It contains 4800 200×200 tile units that are downloaded form TerraServer, and the dataset was taken in 1997. With our GPU-based SIFT, 831084 features are detected from the entire map. The experiments for larger data is not done in this paper due to the time limit.

After feature extraction, a cluster center tree is trained off-line for the quantization of SIFT descriptors. With this tree and some manually chosen scale set and rotation set, image query tasks can be performed after loading features from disk, convecting them to visual words, and adding location pointers to inverted files of visual words. Specifically, A tree of clusters of 5 levels and reasonably 100000 leaves together with 100 feature sizes and 360 feature rotations selected to establish the vocabulary. A set of geometric transformations made up of 5 scale change and 1 rotation are used in the query. The reason that only 1 rotation is chosen is that the query image in our experiments does not have rotation against the database.

To get new query image, we use web browser to visit map databases and take screen shots. we first took a set of 20 screen shots

in the DOQQ 1.0m B&W (west) data at Seamless USGS website (USGS, n.d.). Our query experiment of those images have a 100% success rate at the top match. It is later found out that those two datasets are actually the same except that they have different rectification, which is a small affine transformation between them. Experiments proves that the proposed visual word with SIFT features are working properly in the small non-similarity transformation.

As shown in Fig 3, a set of color query images from different dataset are selected in Google Maps. 40 images are partially-randomly chosen from Google Maps in the same area. Variances of scales in the query images are maintained for verifying how well the proposed visual word can handle scale changes. More pictures with at lease a few roads are chosen on purpose because those regions are close the places where people go to. Considering that the color image set in TerraServer is dated 2003, those query images might be taken no earlier than 2003, which will mean a 6 year gap from the 1997 database. The regions with too much temporal changes are avoided. Fig 4 demonstrates the experimental results, which shows a recognition rate of 90% for top 6 matches and 40% for the top match is obtained. Note that 6 is only .13% of the entire database. Although this is not a general success rate because the query set is not randomly sampled, the result is still promising because normally those parts with salient features are what we are interested in.

Figure 3: Thumbnails of some Sample query images in our experiments.

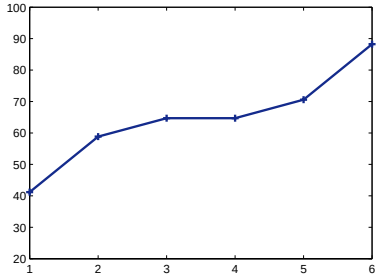

Figure 4: Percentage of ground truth locations that make in to the top X matches. This curve doesn't show a smooth increase because our sample is a relatively small to the database size.

The speed of query without geometric verification is 2hz on a machine with 3Ghz CPU and 1GB RAM. Geometric verification can take charge of finding the most likely location together with the scale and rotation from the top matches. This step takes about 1 to 4 seconds to verify the top 10 putative locations depending on the number of features.

6 CONCLUSION AND FUTURE WORK

The paper proposed a new visual word for indexing orthogonal satellite imagery and the associated method for image-based localization. The proposed visual word incorporates the geographic information of image features, and gives stronger discriminability for indexing images. A scoring implementation is designed to handle the problem of having large images that are made up of the unit database images, which is a significant difference with standard image retrieval. Scale and rotations are recovered together with location by matching the proposed visual words. The future work of this paper includes extending the work to other image features and experimenting on larger database.

REFERENCES

Lowe, D., 2004. Distinctive image features from scale-invariant keypoints. In: International Journal of Computer Vision, Vol. 20.

MicroSoft, n.d. Terraserver. http://terraserver.microsoft.com/.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T. and Van Gool, L., 2005. A comparison of affine region detectors. IJCV.

Nister, D. and Stewenius, H., 2006. Scalable recognition with a vocabulary tree. In: CVPR '06.

Sinha, S., Frahm, J.-M., Pollefeys, M. and Genc, Y., 2006. Gpu-based video feature tracking and matching. In: EDGE06.

Sivic, J. and Zisserman, A., 2003. Video google: A text retrieval approach to object matching in videos. ICCV2003 02, pp. 1470.

USGS, n.d. Seamless usgs. http://seamleass.usgs.gov.

Wu, C., 2007. Siftgpu: A gpu implementation of lowe's sift. In: http://cs.unc.edu/ ccwu/siftgpu.