FAST GEOMETRIC RE-RANKING FOR IMAGE-BASED RETRIEVAL

Sam S. Tsai[1], *David Chen*[1], *Gabriel Takacs*[1], *Vijay Chandrasekhar*[1], *Ramakrishna Vedantham*[2], *Radek Grzeszczuk*[2], *and Bernd Girod*[1]

[1]Information Systems Laboratory, Stanford University, Stanford, CA 94305
[2]Nokia Research Center, Palo Alto, CA 94304

ABSTRACT

We present a fast and efficient geometric re-ranking method that can be incorporated in a feature based image-based retrieval system that utilizes a Vocabulary Tree (VT). We form feature pairs by comparing descriptor classification paths in the VT and calculate geometric similarity score of these pairs. We propose a location geometric similarity scoring method that is invariant to rotation, scale, and translation, and can be easily incorporated in mobile visual search and augmented reality systems. We compare the performance of the location geometric scoring scheme to orientation and scale geometric scoring schemes. We show in our experiments that re-ranking schemes can substantially improve recognition accuracy. We can also reduce the worst case server latency up to 1 sec and still improve the recognition performance.

***Index Terms*—** image-based retrieval, mobile visual search, robust features, geometric verification

1. INTRODUCTION

Mobile image matching applications have gained popular interest as phones become equipped with powerful computing resources and high resolution cameras. Users can hold up their camera-phone and take pictures of objects they would like to inquire, and connect to a mobile image matching system that is either on the phone or a remotely located server, and identify the object and find information of the object anywhere [1, 2, 3].

Most current image-based retrieval systems adopt the feature based image matching approach [4, 1, 5, 6, 3, 2]. By representing images or objects using sets of local features [7, 8, 9], recognition can be achieved by matching features between the query image and candidate database image. Fast large-scale image matching is enabled using a Vocabulary Tree (VT) [10]. Features are extracted from the database of images and a hierarchical k-means clustering algorithm is applied to all of these features to generate the VT. Descriptors of the query image are also classified through the VT and a histogram of the node visits on the tree nodes is generated. Candidate images are then sorted according to the similarity of the candidate database image histogram to the query image histogram.

Geometric Verification (GV) is applied after feature matching [3, 2] to eliminate false feature matches. In this process, features of the query object are matched with features of the database objects using nearest descriptor or the ratio test [7]. Then, a geometric transformation of the location of the features in the query object and the locations of the features in the database object is estimated using RANSAC [11]. A single GV comparison typically takes 30 milliseconds to compute, which prohibits the list of candidate images for verification to a small number.

A number of groups have investigated different ways to speed up the GV process. In [12, 13], the authors investigate how to optimize steps to speed up RANSAC. In [5], they use geometry detected from each local features to estimate the geometric transformation. The authors in [14] have investigated how to perform simple geometric checks by matching visual words. Small feature groups have also been proposed in to incorporate geometry comparison into VT matching yet at the cost of greater complexity [15, 16].

We aim to design a fast and efficient mobile visual search system, and to develop a geometric scoring scheme for large-scale image matching. We find matching feature pairs between a query and candidate image using the descriptor classification paths in the VT. Then, we transform location information of the feature pairs into pairwise distances and generate a geometric similarity score of two images. This approach enables us to incorporate the scoring method described in [14], which uses orientation and scale. However, the two types of information may not be available for certain feature descriptors, such as Rotational Invariant Fast Features (RIFF) [17]. Furthermore, we only re-rank a subset of images using the geometric similarity score. We show that by forming feature pairs using the classification paths in VT and the location geometric scoring method, we can reduce the total time needed and improve the recognition performance.

This paper is organized as the following. In Section 2, we give a brief overview the geometric re-ranking image matching pipeline. In Section 2.1, we present how to generate the matching feature pair list and describe the location geometric scoring scheme that we propose in Section 2.2. We present the experimental results in Section 3.

978-1-4244-7994-8/10/$26.00 ©2010 IEEE

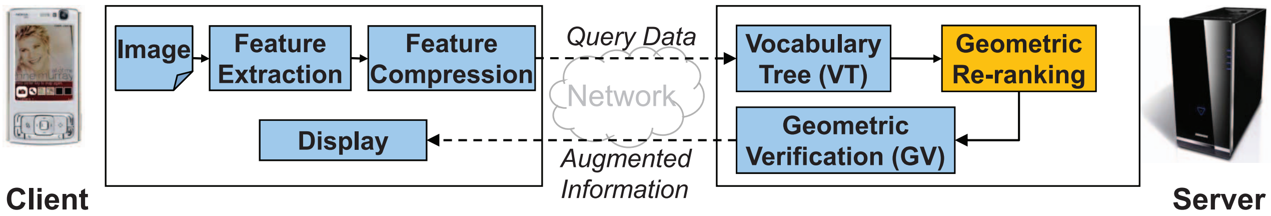

Fig. 1. A typical mobile visual search system is presented in the blue diagram. We propose to add a geometric re-ranking stage which speeds up the system and improves the overall recognition result.

2. GEOMETRIC RE-RANKING IN IMAGE MATCHING SYSTEM

We consider a mobile visual search system with geometric re-ranking as illustrated in Fig. 1. The mobile client takes a picture of a query object and sends the compressed features to a server where the image recognition takes place. On the server, query features are first quantized using a greedy search through the VT [6]. Then, the histogram of the quantized visual words is used to perform a similarity measure between a query image and a database image. We apply geometric re-ranking to a subset of the top matching candidates from the VT search. This improves the final list which is passed on to the GV stage, which typically considers a few images only.

In the next section, we describe how we generate a matching feature pair list M from the VT search and use the list to generate geometric similarity scores between a query feature set and a database feature set. We denote the query feature set as $F_q = \{l_{q,i}, o_{q,i}, s_{q,i}, d_{q,i}\}$, where the variables correspond to location, orientation, scale, and descriptor respectively, and i denotes the index within the feature set. The candidate database feature set is denoted as $F_d = \{l_{d,i}, o_{d,i}, s_{d,i}, d_{d,i}\}$.

2.1. Matching Feature Pairs using Vocabulary Tree

When a descriptor is classified using a VT, the descriptor is compared with the children of a node and the most similar child is selected. The process starts from the root and is repeated until reaching the leaf node, thereby generating a path from the root to the leaf within the tree. In Fig. 2, we show the paths of two feature sets for a VT of depth 3 and branch factor 3.

Descriptors that are similar tend to be quantized along the same path. Thus, we generate a matching feature pair list M of a query feature set F_q and a candidate feature set F_d as follows. For each node within the VT, we examine if there is one and only one query descriptor $d_{q,m}$ that has been classified to that node. Similarly, we check if there is one and only one candidate descriptor $d_{d,n}$ that has been classified to the node as well. If both of these criteria is satisfied for $d_{q,m}$ and $d_{d,n}$, we add (m, n) to the matching feature pair list, M.

Since we also include interior nodes of the VT, there may be duplicate feature pairs in M. Two descriptors that have more than one single-occupancy node in common, tend to be more discriminative. Hence, by allowing duplicate feature pairs in M, we emphasize the effect of more reliable feature matches.

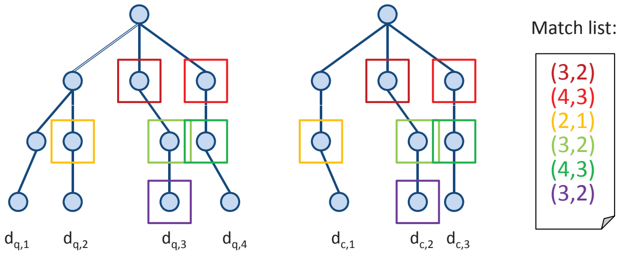

Fig. 2. We show the intersecting feature paths for two sets of features, using a tree of depth 3 and branch factor 3. We start from the root node and go through the nodes breadth first in the tree to match descriptors with one another. The square blocks indicate a node that has only one descriptor are in the feature path for both the query image and candidate image. For each square block, the pair of matching descriptors is added to the list, where the square block's color corresponds to the color of the added pair in the list.

2.2. Geometric Similarity Scoring

We wish to confirm the matching pairs in M using geometry information. In the GV stage, a rigorous validation requires estimating a geometric transformation between the query image and the database image. The estimation of multiple parameters of the geometric transformation renders the process complex and time consuming; thus, we aim to estimate a single parameter instead.

The simplest approach uses only orientation and scale information. If we assume a global rotation between the query image and the candidate matching image, then, matching feature pairs should have a consistent orientation difference. Similarly, matching feature pairs should have a consistent scale difference, corresponding to the global scale change. We describe how to use these two types of information to perform scoring in Sec. 2.2.3.

Using location information of features for geometric re-ranking can be advantageous for several reasons. First, for the

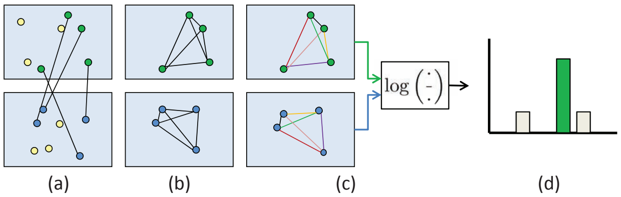

Fig. 3. The process of generating the location geometric score can be shown as the following steps: (a) features of two images are matched according to the descriptor paths, (b) distance of features within image are calculated, (c) log distance ratios of the corresponding pairs (denoted by color) are calculated , and (d) histogram of log distance ratios is formed. The maximum value of the histogram is the geometric similarity score.

client server model shown in Fig. 1, we would only need to send the location information of features, which can be compressed efficiently [18]. Second, as GV typically uses only the location information of features for finding a geometric transformation, the location information is already available for geometric similarity scoring. Furthermore, it is compatible with systems that use features that are rotation invariant, such as Rotation Invariant Fast Features[17], which do not yield orientation information.

However, using location information is not intuitive when the geometric transformation involves translation, scaling, and rotation. We show by using distance between feature locations, we are able to perform single parameter estimation using location information.

2.2.1. *Location geometric similarity scoring*

We propose transforming the location information into distance ratios to measure the geometric similarity, Fig. 3. We generate a set of log of distance ratios from the list M:

$$S_{LDR} = \left\{ \log \left(\frac{dist(l_{q,i}, l_{q,m})}{dist(l_{d,j}, l_{d,n})} \right) \mid (i,j),(m,n) \in M \right\}, \tag{1}$$

where $dist(\cdot\,,\cdot\,)$ corresponds to the Euclidean distance of two points in the image (Fig. 3 (a)-(c)). For two true matching pairs, the value corresponds to the scale ratio between the query and database image. We then estimate the number of features that have similar scale ratio as follows:

$$C_{LDR}(\alpha) = \sum_{z \in S_{LDR}} I\left(\frac{\alpha}{c} \leq z < \frac{\alpha+1}{c} \right), \tag{2}$$

where $I(\cdot\,)$ is the indicator function, and α/c corresponds to the scale ratio difference. c is a tolerance factor that is experimentally determined. In practice, for speed and simplicity, we implement (2) as a histogram with soft bin assignment with α as the histogram bin index. The geometric similarity score of the two feature sets is then given by:

$$Score_{LDR} = \max_{\alpha} C_{LDR}(\alpha) \tag{3}$$

Using log distance ratio enables us to perform single parameter estimation, estimating the scale ratio between the query and database image. Distances are invariant to rotation, scale, and translation. Distance histograms have been used to match point sets [19]. We extend this idea and use distance ratios, while still preserving robustness against similarity transforms.

2.2.2. *Orientation geometric similarity scoring*

Similar to what was described in the previous section, the orientation geometric scoring is formed as follow:

$$\begin{aligned} S_{OD} &= \{(o_{q,i} - o_{d,j}) | (i,j) \in M\}, & (4)\\ C_{OD}(\alpha) &= \sum_{z \in S_{OD}} I\left(\frac{2\cdot \pi \cdot \alpha}{c} \leq z < \frac{2\cdot \pi \cdot (\alpha+1)}{c} \right) & (5)\\ Score_{OD} &= \max_{\alpha} C_{OD}(\alpha) & (6) \end{aligned}$$

Intuitively, this orientation difference corresponds to the global rotation angle between the query image and the database image.

2.2.3. *Scale geometric similarity scoring*

Scale can also be compared by simply using the feature pairs in M. The scale geometric scoring is formed as follow:

$$\begin{aligned} S_{LSR} &= \left\{ \log \left(\frac{s_{q,i}}{s_{d,j}} \right) | (i,j) \in M \right\}, & (7)\\ C_{LSR}(\alpha) &= \sum_{z \in S_{LSR}} I\left(\frac{\alpha}{c} \leq z < \frac{\alpha+1}{c} \right), & (8)\\ Score_{LSR} &= \max_{\alpha} C_{LSR}(\alpha) & (9) \end{aligned}$$

In this case, the log scale difference indicates the scale difference between the query image and the database image.

3. EXPERIMENTAL RESULTS

To demonstrate the performance of our proposed algorithm, we implement and integrate it in the mobile visual search system [3, 2]. We collect 1M images of CD, DVD and book covers and remove similar product images based on appearance. SURF [8] features are extracted from the database images to train the VT. We use a tree configuration of depth 6 and branch factor 10. The feature paths of each database image is generated by classifying the descriptors down the VT and loaded during the geometric re-ranking stage. We pick the the highest scoring 250 images from the VT search and rerank them based on the computed geometric similarity score. We test the recognition performance using a thousand query images[1].

[1] http://msw3.stanford.edu/∼dchen/CDD/index.html

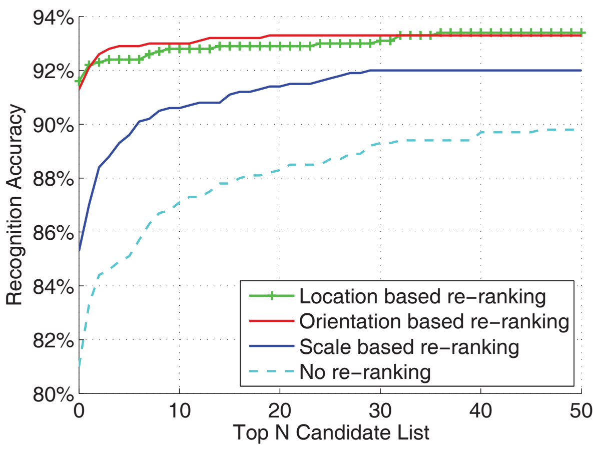

Fig. 4. Performance comparison of different geometric re-ranking schemes. Correct match is declared if the true match is within the top N candidates.

We show the performance of the different geometric scoring methods and the results without re-ranking in Fig. 4. We declare a match to be correct if the corresponding image is within the top N candidate in the list. By incorporating a geometric re-ranking method, the recognition result can be boosted significantly. The location geometric scoring and the orientation geometric scoring method perform similarly and has a clear advantage over the scale based method. For both methods, the accuracy of the top matching result is improved from 81% to 91% compared to that without re-ranking.

We show the average time for the three different geometric scoring methods along with the time required for GV in Tab. 1. We see that the location geometric similarity scoring takes longer than the orientation and scale geometric scoring methods. This is because the total number of calculations for the distance is O(n^2). All three are only a small fraction of the time of one single GV comparison, which is 30 ms.

Table 1. Processing time for each geometric comparison scheme.

Comparison scheme	Time per comparison (ms)
Orientation geometric scoring	0.1
Scale geometric scoring	0.12
Location geometric scoring	0.46
Geometric Verification (GV)	30

A typical system performs GV for the top 50 images with preemptive stop, yielding a worst case scenario of $30 \cdot 50 = 1500$ ms latency with a recognition rate of ∼90%. For a geometric re-ranking system using location geometric scoring, we need only retain the top 5 images for GV while providing a system that has a recognition performance of ∼92%. In this case the worst case time is only $(30 \cdot 5 + 115) = 265$ ms, which is only <18% of the typical system and 1 second faster.

4. CONCLUSIONS

We develop a new method of incorporating geometric similarity re-ranking for mobile image matching systems. Based on the classification feature paths in the VT, a list of matching database and query feature pairs is computed. We use geometric similarity scoring to re-rank candidate matching images given by the tree search. We develop a location geometric scoring that is invariant to similarity transform, compatible with rotational invariant features, and can be conveniently integrated in a mobile visual search system. We improve the recognition accuracy from 81% to 91% for the top matching result, and can reduce the overall latency by 1 sec.

5. REFERENCES

[1] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *ACM International Conference on Multimedia Information Retrieval*, Vancouver, Canada, October 2008.

[2] S. S. Tsai, D. Chen, J. Singh, and B. Girod, "Rate-efficient, real-time CD cover recognition on a camera-phone," in *ACM International Conference on Multimedia*, Vancouver, Canada, October 2008.

[3] D. Chen, S. S. Tsai, R. Vedantham, R. Grzeszczuk, and B. Girod, "Streaming mobile augmented reality on mobile phones," in *International Symposium on Mixed and Augmented Reality*, Orlando, FL, USA, October 2009.

[4] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," in *International Conference on Computer Vision*, 2003, vol. 2, pp. 1470–1477.

[5] J. Philbin, O. Chum, M Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[6] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, June 2007, pp. 1–7.

[7] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.

[8] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: speeded up robust features," in *European Conference on Computer Vision*, Graz, Austria, May 2006, pp. 404–417.

[9] V. Chandrasekhar, G. Takacs, D. Chen, S. S. Tsai, R. Grzeszczuk, and B. Girod, "CHoG: Compressed Histogram of Gradients," in *In Proceedings of Conference on Computer Vision and Pattern Recognition*, 2009.

[10] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, June 2006, pp. 2161–2168.

[11] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cryptography," *Communications of ACM*, vol. 24, no. 1, pp. 381–395, 1981.

[12] O. Chum, J. Matas, and J. V. Kittler, "Locally optimized RANSAC," in *Proceedings of DAGM*, 2003, pp. 236–243.

[13] O. Chum, T. Werner, and J. Matas, "Epipolar geometry estimation via ransac benefits from the oriented epipolar constraint," in *International Conference on Pattern Recognition*, Washington, DC, USA, 2004, pp. 112–115, IEEE Computer Society.

[14] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European Conference on Computer Vision*, 2008, pp. I: 304–317.

[15] Z. Wu, Q. Ke, M. Isard, and J. Sun, "Bundling features for large scale partial-duplicate web image search," in *Conference on Computer Vision and Pattern Recognition*, 2009, pp. 25–32.

[16] O. Chum, M. Perdoch, and J. Matas, "Geometric min-hashing: Finding a (thick) needle in a haystack," in *Conference on Computer Vision and Pattern Recognition*. 2009, pp. 17–24, IEEE.

[17] G. Takacs, V. Chandrasekhar, S. S. Tsai, D. M. Chen, R. Vedantham, R. Grzeszczuk, and B. Girod, "Unified real-time tracking and recognition with rotation-invariant fast features," in *Conference on Computer Vision and Pattern Recognition*, 2010, p. submitted.

[18] S. S. Tsai, D. M. Chen, G. Takacs, V. Chandrasekhar, R. Vedantham, R. Grzeszczuk, and B. Girod, "Location coding for mobile image retrieval," in *Proc. 5th International Mobile Multimedia Communications Conference*, 2009.

[19] M. Boutin and M. Comer, "Faithful shape representation for 2D gaussian mixtures," in *International Conference on Image Processing*, 2007, pp. VI: 369–372.