MOBILE 3D VISUAL SEARCH USING THE HELMERT TRANSFORMATION OF STEREO FEATURES

Haopeng Li and Markus Flierl

School of Electrical Engineering
KTH Royal Institute of Technology, Stockholm
{haopeng, mflierl}@kth.se

ABSTRACT

This paper presents a scheme for mobile 3D visual search that facilitates mobile recognition of 3D objects. We use a multi-view approach to extract the 3D geometric information of the query objects and integrate it into SIFT descriptors. To meet a given transmission bandwidth, we use a rate-constrained quad-tree representation for feature selection and encoding. With this approach, we are able to progressively match the query features against the stereo features in the database and implement a robust geometric verification with the Helmert transformation. The experimental results show that our proposed approach to mobile 3D visual search outperforms JPEG and single-view SIFT-based search.

Index Terms— Mobile visual search, 3D geometric verification, Helmert transformation.

1. INTRODUCTION

The advancement of wireless mobile devices and the desire for an augmented reality in a real-world environment have raised interest in applications of mobile visual search [1] [2]. Visual search allows users interactive and semantic access to real-world objects. With the integration of digital cameras into mobile devices, image-based information retrieval for mobile visual search is developing rapidly. A crucial problem is the efficient utilization of the information in the mobile images.

The challenges of mobile image retrieval are rooted in the bandwidth constraint and the limited computational capacity of mobile devices. To solve these problems, most of the mobile visual search algorithms use the so-called bag of features approach where only the salient image features are extracted and sent.

Therefore, the detection of features and the computation of descriptors play an important role in feature extraction. The well-known Scale Invariant Feature Transform (SIFT) [3] has been widely used in visual search applications. It is more robust than many other well-known features in the context of feature matching and recognition due to its invariance under rotation, scale change and affine transformation [4]. However, the direct transmission of SIFT descriptors is not practical due to the large data volume. In particular, the amount of SIFT data is usually larger than the size of the JPEG-compressed image itself [1]. Hence, several compression schemes have been proposed to solve this problem. An efficient approach is known as Compressed Histogram of Gradients (CHoG) [5], which can reduce the data rate by factor 20 when compared to that of the uncompressed SIFT descriptors. However at very low data rate, the recall rate decreases significantly as only a few features can be matched correctly.

Currently, mobile visual search uses only 2D image-based features for object recognition while ignoring the underlying 3D geometric information. Expanding feature descriptors to capture also 3D geometric information will improve the recall rate. In such a case, the assessment of a query will be based on the visual appearance of the object as well as the underlying 3D geometry. That is, in cases where the visual appearance of a query is very similar, the underlying 3D geometry can be used to discriminate objects.

In this paper, we propose an approach that uses stereo features for 3D object recognition. Our experiments will focus on the recognition of buildings. Unlike dominant color or structural model-based methods [6] [7], our method uses only the discrete 3D points without utilizing any prior knowledge. To characterize the 3D geometric structure of objects, we use a multi-view approach to extract stereo SIFT features with the corresponding 3D geometric information. To meet the bandwidth constraint on the client side, we propose a rate-constrained quad-tree for feature selection and encoding. With the tree-structured data, we are able to match the query features progressively. As the 3D geometric information of the features is available, we propose a method based on the Helmert transformation [8] to verify the 3D geometry.

2. MOBILE 3D VISUAL SEARCH

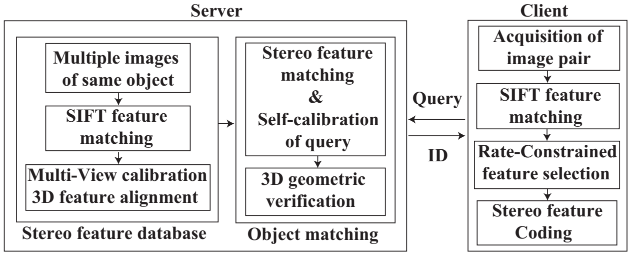
Server
Client
Multiple images of same object
SIFT feature matching
Multi-View calibration 3D feature alignment
Stereo feature database
Stereo feature matching & Self-calibration of query
3D geometric verification
Object matching
Query
ID
Acquisition of image pair
SIFT feature matching
Rate-Constrained feature selection
Stereo feature Coding

Fig. 1. Mobile 3D visual search

Our scheme for 3D visual search is shown in Fig. 1. The client extracts and encodes the stereo features of the query. The server holds the stereo-feature database derived from the multi-view imagery as well as the stereo-feature matching engine.

2.1. 3D Visual Queries on Mobile Devices

In the following, we discuss the algorithm at the client. We aim at an efficient selection and encoding of the most relevant stereo features while meeting the constraints of bandwidth and computational resources.

2.1.1. *Acquisition of Image Pairs and SIFT Feature Matching*

The images of the query building are acquired by a monocular digital camera which can be commonly found on mobile devices. Considering that the computational capacity of mobile devices is relatively low, we take only two images from different viewpoints for each building and define them as the image pair V_k.

For each image pair V_k, we extract the SIFT features and find the correspondences. The correct correspondences relate to the same 3D point in the scene and, hence, can be used to identify the geometric information of the related features. To refine the matching results, we use the eight-point epipolar-constrained [9] RANSAC algorithm [10]. Finally, we get a set of reliable feature correspondences, defined as the stereo feature set Q_c at the client side.

There are two advantages of sending stereo features. First, we can represent the geometric information associated with the features. Second, features with established correspondences on the client side are more robust for matching on the server side.

2.1.2. *Rate-Constrained Feature Selection*

For a query of the database at the server, we transmit a set of stereo features. In order to meet the bandwidth constraint, we sample stereo features from Q_c and send only the most reliable candidates.

Let B_b be the bandwidth constraint of the mobile device and $B_t = \rho M$ the actual bandwidth, where M is the number of transmitted stereo features and ρ the datarate per stereo feature, which will be addressed in Section 2.1.3.

Further, let $d(x, y)$ denote the feature distribution over the image support $I(x, y)$, where x and y are the image coordinates. In order to choose relevant features in a rate-constrained setting, we approximate the feature distribution by a quad-tree representation. We partition the image support into a set of M piecewise dyadic regions $T = \{R_m, m = 1, \ldots, M\}$, where the individual regions R_m satisfy $\cup_{m=1}^{M} R_m = I$ and $R_m \cap R_{m'} = \emptyset$ for $m \neq m'$. With that, the approximate feature distribution for a given partition T is

$$d_T(x, y) = \frac{1}{|Q_c|} \sum_{m=1}^{M} |S_m| g_m(x, y), \qquad (1)$$

where $|S_m|$ is the size of the stereo feature set S_m in a given region R_m and $g_m(x, y) = 1_{R_m}(x, y)$ the indicator function for the region R_m. Note that this approximation can be easily expanded to non-dyadic regions.

As we use the model distribution (1) to approximate the actual feature distribution, a larger M will lead to a better approximation with smaller approximation error variance $\mathrm{E}\left\{(d - d_T)^2\right\}$ [11]. However, it will also result in a higher bandwidth cost. Therefore, we need to balance the trade-off between approximation error and bandwidth. We obtain the optimal partition T_{opt} by solving the following rate-constrained problem

$$\begin{aligned} \min_{T} \quad & \mathrm{E}\left\{(d - d_T)^2\right\} \\ s.t. \quad & \rho M \leq B_b. \end{aligned} \qquad (2)$$

We solve this problem by recursively partitioning Q_c into M subsets from top-to-down. Each node in the tree is defined by a dyadic square R_m. A new partition T can be formed by decomposing one of the dyadic squares R_m into four dyadic sub-squares. We always decompose the dyadic square R_m which leads to a smaller error variance. With this recursive partition method, we need to visit each node of the tree only once. Two examples of the rate-constrained quad-tree approximation are shown in Fig. 2.

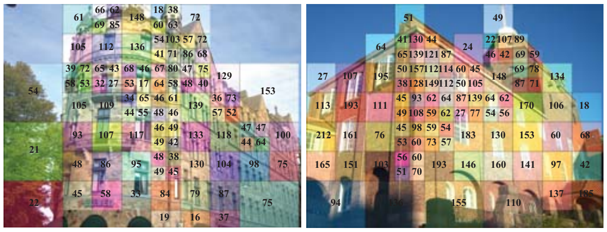

Fig. 2. Quad-tree partition of stereo feature space into at most 100 dyadic squares. Each highlighted block indicates one dyadic square R_m, the number in each square indicates the size of feature set $|S_m|$.

For a given optimal partition T_{opt}, we use the Euclidean distance ratio between the first and second nearest feature correspondence [3] to choose the most robust stereo feature in each subset S_m. In other words, we select the stereo feature with the smallest distance ratio in each S_m. Hence, M robust stereo features from Q_c will be transmitted to the server.

2.1.3. *Coding of Stereo Features*

After choosing M stereo features by rate-constrained approximation, the features are encoded with high accuracy such that the feature information can be reconstructed at the server.

We need the location information of features in both images to reconstruct the 3D geometric information from the stereo features. For efficient encoding, we encode the 2D location data in one image and the disparity information in the other. Additionally, the intrinsic information of the camera, including the focal length and the image size, is also transmitted. The use of intrinsic camera information is discussed in Section 2.2.2.

Therefore, we have three sets of parameters to encode: SIFT descriptors, location data and intrinsic camera information. As we require a high accuracy for the selected stereo features, we choose a double-precision representation for each parameter and use arithmetic coding to encode the parameter sets. With this approach, we achieve a datarate ρ of about 0.1 KBytes per stereo feature.

2.2. 3D Feature Database at the Server

2.2.1. *Representation of Feature Sets*

A database with efficient data structure plays a crucial role in mobile visual search. Comparing the features received from the client to all the features in the database is infeasible due to the large amount of features. Therefore, we reuse the quad-tree representation in Section 2.1.2 to efficiently index the features in the database for progressive feature matching. Note that we will address the matching strategy in Section 2.3. Unlike conventional databases of image-based features, we store only the indexed stereo features with geometric information. This approach reduces significantly the number of stored features and the geometric information can be used efficiently for 3D object recognition.

For the server database, we acquire multiple images from each 3D object, i.e., building. For N acquired images, we define K image pairs as $(1, 2), (2, 3), \ldots, (N - 1, N)$. With a similar procedure

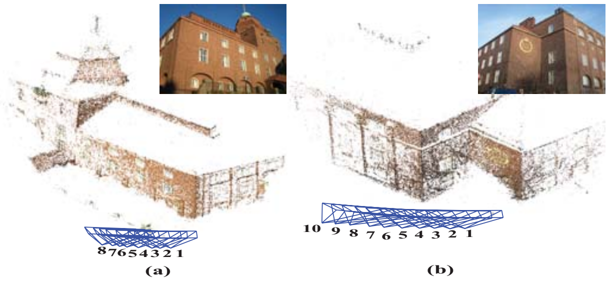

Fig. 3. 3D reconstruction of stereo features using multi-view images. A small snapshot is shown in the upper right corner. The subplot (a) uses 8 images with 58285 stereo features; the subplot (b) uses 10 images with 82213 stereo features.

as introduced in Section 2.1.1, we obtain the set of reliable stereo features Q_s^k from the k-th image pair at the server. With multi-view calibration, N views are aligned and the geometric information of the stereo features set Q_s^k is extracted.

2.2.2. *Multi-View Calibration*

Two-view self-calibration: Our multi-view calibration begins with a two-view self-calibration for the first image pair $(1, 2)$. We use the intrinsic information from the EXIF tags of the camera image file to estimate the intrinsic camera parameter, in particular the focal length $f = \frac{w \cdot f_m}{w_m}$, where f and w are the focal length and the image width in pixels, f_m and w_m are the actual focal length and the width of the CCD sensor in millimeters, which are readings from the EXIF tags.

To solve the ill-posed problem of self-calibration caused by outliers of feature correspondences, we utilize the reliable fundamental matrix F_1 of the epipolar-constrained RANSAC algorithm by imposing the singularity constraint rank$(F_1) = 2$ [12]. With the intrinsic matrices A_1 and A_2 of view 1 and 2, we calculate the essential matrix $E_1 = {A_2}^T F_1 A_1$. We set the view 1 as reference view at the origin of the 3D world coordinates. For a given essential matrix E_1, four solutions for translation matrix T_2 and rotation matrix R_2 of view 2 can be calculated by applying singular value decomposition (SVD) on E_1 [12]. Physically impossible solutions are discarded by using the positive depth constraint [13]. After knowing the self-calibration parameters, the set of 3D world coordinate $P_s^1 = \{(X_n, Y_n, Z_n)\}$ of the corresponding stereo feature set Q_s^1 can be calculated up to an unknown scale.

Adding views: To align all stereo features among all available views, we use the direct linear transform (DLT) [9] to add additional views to the calibration process. We compare the neighboring set Q_s^2 to Q_s^1 and choose the joint features $U_3 = Q_s^1 \cap Q_s^2$. The associated 3D world coordinates of U_3 can be extracted from P_s^1. The projection matrix of the new view is calculated by the DLT. The camera calibration parameters are obtained by decomposing the projection matrix with QR-factorization.

Bundle adjustment: Finally, we use the non-linear least squares method known as Levenberg-Marquardt algorithm [14] to minimize the reprojection error caused by accumulating relative orientations. We use the sparse bundle adjustment [15] for an efficient implementation. Two examples of 3D reconstruction of stereo features are shown in Fig. 3.

2.3. 3D Matching

As introduced in Section 2.2, the set of stereo features is approximated by a quad-tree representation. Now, this representation is used for progressive stereo feature matching. Additionally, the 3D geometric information of the stereo features permits efficient 3D matching.

2.3.1. *Strategy of Matching Pairs of Stereo Features*

Similar to Section 2.1.2, we compute the Euclidean distance ratio for each stereo feature in Q_s^k. The indexing of the stereo features in one S_m is achieved by sorting them according to the Euclidean distance ratio in ascending order. We define a level set for stereo features with the same index as

$$J_i = \cup_{m=1}^{M_s} S_m(i), \tag{3}$$

where M_s is the number of partitions, $\cup_{i=1}^{L} J_i = Q_s^k$, and $L = \max |S_m|$.

The level set J_i with smaller index contains more robust features. Therefore, we propose a progressive matching strategy based on the best-feature-first policy. We define ν as the minimum number of matched features after geometric verification. For each received set of query features, we match it against all stereo feature level sets J_i from the lowest level. This procedure will stop when the number of correctly matched features satisfies the threshold ν.

This strategy offers two advantages: First, the best-feature-first policy allows more reliable features to be used first. This accelerates the matching process. Second, the number of features in the server database can be significantly reduced since relatively few features are needed for matching. Note that this paper focuses on the tradeoff between recall and datarate. Fast methods like the vocabulary tree [1] can be used to speed up the process.

2.3.2. *Geometric Verification using the Helmert Transformation*

To obtain the correct matches, we need to introduce geometric verification into the matching process. Usually, an epipola-constrained RANSAC algorithm is used for single-view visual search. However, the epipolar constraint can not capture the 3D information of the object. With our multi-view approach, the 3D geometric information of stereo features can be efficiently used for geometric verification.

First, we need to extract the 3D geometric information from the query. Combining the location data of the features and the intrinsic camera information, we obtain the corresponding set of 3D world coordinates P_c by using the method of two-view self-calibration in Section 2.2.2.

From Section 2.3.1, we get stereo feature correspondences between query and database, which are based on descriptor matching. With that, the associated 3D world coordinates of the query and the server database are available. The two-view self-calibration can reconstruct the 3D world coordinates up to an unknown scale with relative translation and rotation. For a correct 3D matching, we use the so-called seven-parameter Helmert transformation to describe the relationship between the 3D world coordinates of query and server object points

$$\vec{p_c} = s\,\Phi\,\vec{p_s} + \vec{t}, \tag{4}$$

with $\vec{p_c} \in P_c$ and $\vec{p_s} \in P_s^k$, where s is the scale parameter in $\mathbb{R}^+$, Φ the rotation matrix in $\mathbb{R}^3$ and $\vec{t}$ the translation parameter in $\mathbb{R}^3$.

The seven-parameter Helmert transformation can be determined by finding the least squares solution using the Hanson-Norris method [16] with at least three correspondences between world coordinates. However, some misalignment of world coordinates

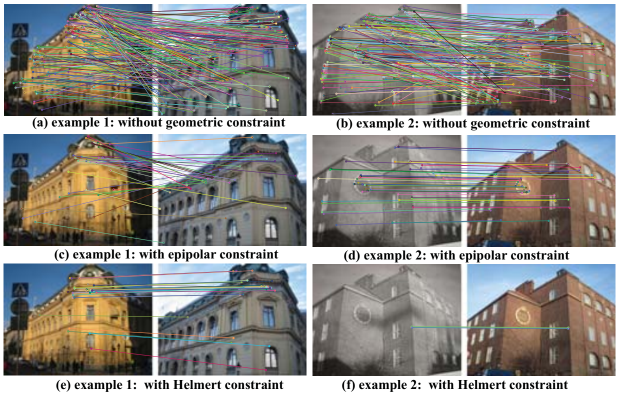
(a) example 1: without geometric constraint
(b) example 2: without geometric constraint
(c) example 1: with epipolar constraint
(d) example 2: with epipolar constraint
(e) example 1: with Helmert constraint
(f) example 2: with Helmert constraint

Fig. 4. Comparison of using different geometric constraints. The subplots (a) (c) (e) show example 1; the subplots (b) (d) (f) show example 2. In each subplot, the left image is the query and the right image is the corresponding one from the database.

caused by erroneous calibration parameters should also be considered. Therefore, we integrate the Helmert constraint into the RANSAC procedure. In this case, we use the Helmert transformation as a model to fit the sets of 3D world coordinates and find the sample consensus. Only three correspondences are needed per RANSAC iteration.

3. EXPERIMENTAL RESULTS

We evaluate our mobile 3D visual search for the multi-view image dataset *Stockholm Buildings*[1] which comprises 50 buildings of the city. The server holds 254 images of the 50 buildings. At least 2 views have been recorded for each building. The client may use up to 100 additional test images of the 50 buildings. We took server and test images at the different viewpoints and at different times. The images have been taken by a Cannon IXUS50 digital camera at resolution 2592 × 1944 pixels.

3.1. Geometric Verification using the Helmert Transformation

Now, we verify the method of using the seven-parameter Helmert transformation for geometric verification. At this point, the datarate is not constrained.

The first example is shown in Fig. 4(a)(c)(e). Due to the change of viewpoints and lighting conditions, a large amount of features are wrongly matched without using any geometric verification. As shown in Fig. 4(c), the epipolar constraint does not work here since it can not find a consistent solution in the highly noisy environment. With 3D geometric information, our Helmert-constrained approach is able to find the correct feature correspondences.

The second example is shown in Fig. 4(b)(d)(f). Here, the query pair is taken from another image, a so-called ”picture pair of picture”. In other words, the query content is actually a picture on a billboard, instead of a real building. As shown in Fig. 4(d), the 3D recognition fails as the epipolar constraint permits many correspondences. As the actual 3D positions of the query features sit on a flat surface, our Helmert-constrained based method accepts only very few correspondences. However, they are quickly discarded by the RANSAC procedure.

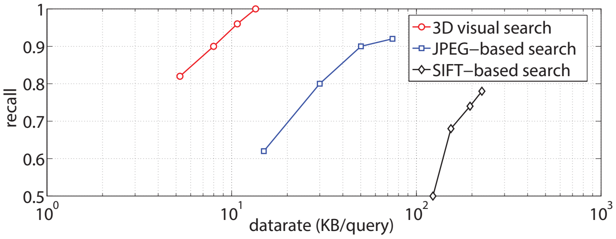

Fig. 5. Comparison of the recall-datarate trade-off between reference schemes and 3D visual search.

3.2. Trade-Off between Recall and Datarate

In the following, we investigate the trade-off between the recall and the datarate for 3D visual search. The recall is defined by the percentage with which the query object is retrieved correctly from the server database. The datarate is simply the size of the query packet which is sent to the server. We choose $\nu = 12$ for the minimum number of matched features after geometric verification [1].

We compare our mobile 3D visual search with two other schemes which use single views only to recognize query building. The first reference transmits the JPEG-compressed image. We resize the image from 460 × 340 to 128 × 100 to vary the transmission rate. The second reference transmits the compressed SIFT features from a single view only. We encode the SIFT descriptors and location data by arithmetic coding. For a given datarate budget, we choose an appropriate number of SIFT features. For a fair comparison, we use the same dataset and progressive matching for all schemes. The reference schemes use the epipolar-constrained RANSAC. The results are shown in Fig. 5.

For the JPEG-based scheme, the recall quickly degrades when decreasing the resolution. For the single-view SIFT-based scheme, the recall is limited due to challenging cases (i.e., large baseline, different lighting conditions) even at high datarate. Our method achieves 100% recall at 13.5 KB per query by sending only 130 stereo features on average.

4. CONCLUSIONS

We discussed a scheme for mobile 3D visual search and tested it on the dataset *’Stockholm Buildings’*. We use a multi-view approach to characterize the 3D geometric information of the query building. Due to the mobile setting, we define stereo features and a rate-constrained quad-tree representation. This allows progressive matching at the server, and hence, accelerates the search. To fully utilize the geometric information, we propose a Helmert-constrained RANSAC for geometric verification. The experimental results show that our 3D visual search outperforms JPEG and single-view SIFT-based methods. Future research may incorporate more compact feature descriptors, such as CHoG.

5. ACKNOWLEDGMENTS

This work has been supported in part by the Swedish Research Council in the context of the project ”Mobile Sensing” of the Strategic Research Area ICT-TNG.

[1] `http://www.ee.kth.se/~haopeng/sthlmbuildings`

6. REFERENCES

[1] B. Girod, V. Chandrasekhar, R. Grzeszczuk, and Y. Reznik, "Mobile visual search: Architectures, technologies, and the emerging MPEG standard," *IEEE Trans. on Multimedia*, vol. 18, no. 3, pp. 86 –94, Mar. 2011.

[2] D. Chen, G. Baatz, K. Koser, S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, C. Xin, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk, "City-scale landmark identification on mobile devices," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2011.

[3] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60(2), pp. 91–110, 2004.

[4] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615 –1630, 2005.

[5] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "CHoG: Compressed histogram of gradients A low bit-rate feature descriptor," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2009.

[6] W. Zhang and J. Kosecka, "Localization based on building recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2005.

[7] Y. Chung, T. Han, and Z. He, "Building recognition using sketch-based representations and spectral graph matching," in *Proc. of the IEEE International Conference on Computer Vision*, Oct. 2009.

[8] G. Watson, "Computing Helmert transformations," *Journal of Computational and Applied Mathematics*, vol. 197, no. 2, pp. 387 –394, 2006.

[9] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd edition, 2004.

[10] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[11] M. Newman and G. Barkema, *Monte Carlo Methods in Statistical Physics*, Clarendon Press, 1999.

[12] S. Carlsson, "Geometric computing in image analysis and visualization," Lecture Notes, KTH Royal Institute of Technology, Stockholm, Mar. 2007.

[13] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*, pp. 124 –125, Springer, 2003.

[14] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 2nd edition, 2006.

[15] M. Lourakis and A. Argyros, "SBA: A software package for generic sparse bundle adjustment," *ACM Trans. Math. Software*, vol. 36, no. 1, pp. 1 –30, 2009.

[16] R. Hanson and M. Norris, "Analysis of measurements based on the singular value decomposition," *SIAM Journal on Scientific and Statistical Computing*, vol. 2, pp. 363 –374, 1981.