Hierarchically Structured Multi-View Features for Mobile Visual Search

Xinrui Lyu, Haopeng Li and Markus Flierl
School of Electrical Engineering
KTH Royal Institute of Technology, Stockholm
{xinruil, haopeng, mflierl}@kth.se

Abstract

This paper presents an approach for using hierarchically structured multi-view features for mobile visual search. We utilize a graph model to describe the feature correspondences between multi-view images. To add features of images from new viewpoints, we design a level raising algorithm and the associated multi-view geometric verification, which are based on the properties of the hierarchical structure. With this approach, features from new viewpoints can be recursively added in an incremental fashion. Additionally, we design a query matching strategy which utilizes the advantage of the hierarchical structure. The experimental results show that our structure of the multi-view feature database can efficiently improve the performance of mobile visual search.

1 Introduction

Image-based information retrieval systems such as mobile visual search [1] [2] [3] have been developed rapidly in recent years. They allow interactive and semantic access to real-world objects by simply taking a picture of the desired object. However, mobile image retrieval is generally constrained by the limitations of bandwidth and computational capacity of mobile devices. Therefore, the so-called bag-of-features approach [4] is usually used where only the salient image features are extracted and sent as queries.

To match a query with the corresponding object at the server, a reliable database with efficient data structure plays a crucial role. The well-known vocabulary tree (VT) methods [5] [6] have been widely used in indexing the image features. It essentially utilizes k-means methods to partition the descriptor space into visual words. Then, the clustered visual words are used to construct the vocabulary tree. However, when the database grows by adding more features, it is important to flexibly accommodate and index new features [7]. In particular for multi-view imagery [8] [3], adding more images at different scales and perspectives is common.

For a multi-view feature database, the selection of image features is usually based on feature correspondences across multiple views. Features with well-established correspondences are more robust for matching with query features. By utilizing relevant multi-view feature correspondences, it is possible to achieve an advanced matching performance while using a smaller number of image features. However, three important issues need to be resolved when increasing the number of images for each object. First, new features should be added incrementally to the existing features. Second, the new database of features should be more efficient than the previous one.

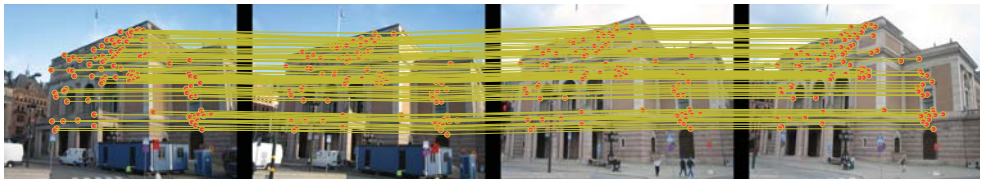

Figure 1: Hierarchical sets of features from four views.

Third, the increase in computational complexity of matching a query against the new database features should be limited.

In this paper, we propose a structure for multi-view features that improve the performance of mobile visual search. Hierarchically structured multi-view feature sets with multiple levels are constructed and used for efficient matching. With our structure, we are able to recursively manage the new features and update the database. To maintain the geometric consistency among the multi-view imagery, we propose a multi-view fundamental matrix. Taking advantage of our structured features, the recall-rate performance is improved without increasing the computational complexity of the query matching process.

2 Multi-View Image Features at the Server

Large-scale objects such as buildings are usually hard to match due to significant change of viewpoint and lighting conditions between query and server. Thus, an image database at the server with a considerable perspective diversity will improve the performance of mobile visual search.

For the server, we acquire multiple images from each building. For each image, we extract a set of the Scale Invariant Feature Transform (SIFT) features due to its robustness under rotation, scale change and affine transformation [9]. However, as there are multiple feature sets for the same object, the redundancy in feature space is high. Thus, an efficient feature selection algorithm is needed to discriminate features. On the other hand, as the images are taken from different perspectives with varying lighting conditions, using a robust representation of feature descriptors is important for reliable feature matching. Furthermore, the data structure should be augmentable to be able to to add features from new viewpoints.

2.1 Hierarchical Structure of Sets of Features

Due to the high redundancy among the multi-view image features, comparing the features received from the client to all the features at the server is inefficient. Moreover, the reliability of features in the multi-view imagery is varying. For example, the features from the foreground are more reliable when compared with those from the background. Therefore, a best-feature-first policy should be applied such that more reliable features are used first.

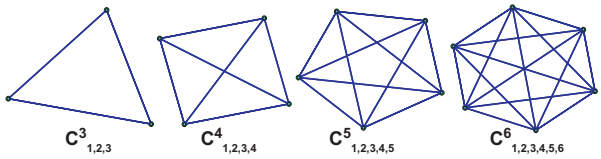

Figure 2: Feature correspondences modeled as complete graphs.

Multi-view feature correspondences provide a way to measure the reliability of the features. Let $f_i \leftrightarrow f_j$ be a feature correspondence, where f_i denotes the feature point in the i-th image and f_j the corresponding one in the j-th image. Further, let $f_i \leftrightarrow f_j \leftrightarrow \ldots \leftrightarrow f_k$ be a multi-view feature correspondence that indicates a correspondence among features in several images. With that, we define a set of feature correspondences using l images $C^l_{i,j,\ldots,k} = \{(f_i, f_j, \ldots, f_k) | f_i \leftrightarrow f_j \leftrightarrow \ldots \leftrightarrow f_k\}$ and represent it by a complete graph with l vertices. The i-th vertex in the graph represents a feature in the i-th image. Each edge represents the correspondence between two features. Our multi-view feature correspondences imply correspondences between all possible pairs of features. Hence, we represent them as undirected complete graphs. Examples of sets $C^l_{i,j,\ldots,k}$ are shown in Fig. 2. The advantage of modeling multi-view feature correspondences as complete graphs for geometric verification will be addressed in Section 2.2.2.

For our problem, a complete graph with more vertices is more reliable and representative than a complete graph with less vertices. Simply speaking, a large complete graph represents correspondences among many images. An example with four vertices is shown in Fig. 1. Here, a graph with four vertices is the largest since it establishes correspondences among all four images. Most of the correspondences relate to the foreground object. Note, there are no multi-view features located on the background objects such as containers, humans, and remote buildings.

As multi-view features with more vertices are more reliable for robust matching, we structure the sets of feature correspondences in a hierarchical manner. In particular, sets of feature correspondences with l vertices are placed on level l. Further, subsets with l vertices, where feature correspondences can be established, are placed on level l. An example for hierarchical sets of features with four levels from four views is shown in Fig. 3.

Each feature correspondence $f_i \leftrightarrow f_j \leftrightarrow \ldots \leftrightarrow f_k$ at level l has l SIFT feature descriptors with 128 coefficients each. Thus, if there is a common feature descriptor to represent the l corresponding features, we can reduce the number of descriptors significantly. Due to image noise, changes of perspective, and varying lighting conditions, we view the l feature descriptors as a set of measurements with outliers. To represent them by a reliable descriptor, we take the median of the l descriptors as a robust estimate [10]:

$$\widehat{d^l(u)} = \text{Median}\{d^l_h(u) : h = i, j, \ldots, k\},\ u = 1, \ldots, 128, \tag{1}$$

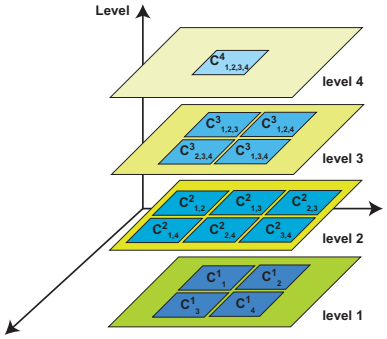

Figure 3: Hierarchical sets of features with four levels from four views.

where $\widehat{d^l}$ is the representative descriptor at level l, and d_h^l the feature descriptor in the h-th view.

The hierarchical structure offers four advantages: First, the quality of a set of features is supported by multiple correspondences as determined by the associated graph. Second, the representative descriptor which is estimated by the median of the feature descriptors is robust to changes of perspective and varying lighting conditions. Third, using the representative descriptors reduces the redundancy of the set of features significantly. Fourth, progressive matching and recursive adding of new features can be accomplished efficiently. We will address the fourth advantage in the following.

2.2 Adding New Features to the Hierarchical Structure

For reliable search, more features from different viewpoints improve the probability of correct recall of a query. To improve the quality of the sets of features at the server, and hence, improve the recall results for visual search, we would like to add features captured from new viewpoints or at different lighting conditions to the existing sets at the server.

2.2.1 Level Raising Algorithm

The hierarchical structure of the multi-view feature sets offers the advantage that features from new images can be added to the existing sets in an incremental fashion. We propose a top-to-bottom algorithm to efficiently raise each level of the hierarchical structure. An example is shown in Fig. 4.

Let $S_k^{(t)}$ denote the set of new features from the k-th viewpoint at the updating step t. We assume that there are $k-1$ views in the existing sets, then the maximum number of levels is $m = k - 1$. As the top level contains the most robust features when compared to the other levels, we begin the raising algorithm from the top level.

In the beginning, we match the hierarchical sets $C_{i,..j}^{m,(0)}$ at level m with the feature set $S_k^{(0)}$ by using the fast nearest-neighbor criterion [9]. It will result in three sets

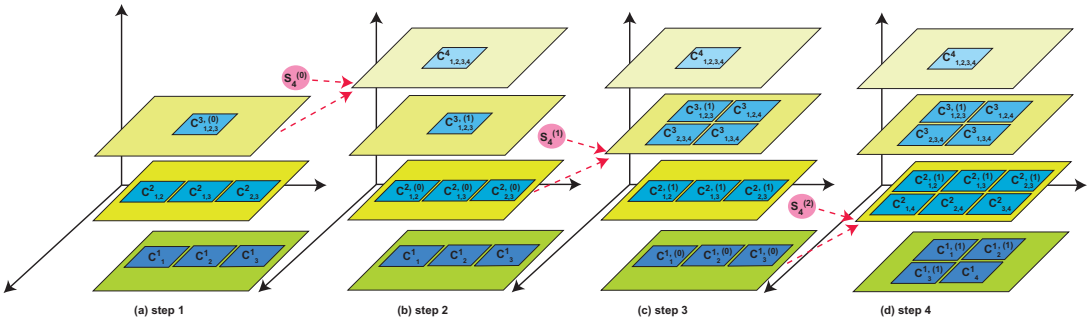

Figure 4: Adding new features to the current sets of features.

of features. The first set is the new hierarchical set $C^{m+1}_{i,\dots j,k}$ with added features from $S^{(0)}_k$ which has been raised to $m+1$. The second and third sets are the remaining hierarchical sets at level m and the updated set of new features at step 1, denoted as $C^{m,(1)}_{i,\dots j}$ and $S^{(1)}_k$, respectively. The remaining hierarchical sets $C^{m,(1)}_{i,\dots j}$ still sit on level m, while the updated feature set $S^{(1)}_k$ is matched with the sets at level $m-1$ in the next step. Note that after matching the new feature set S_k at all levels, a geometric verification is needed to eliminate possible outliers. We will address this verification in Section 2.2.2.

With our hierarchical structure, we are able to raise the sets at a lower level to an higher level if a matching feature can be found in the set of new features. The remaining features in $S^{(t)}_k$ which can not be matched at all will be placed at the bottom level $m = 1$. Usually, most of the features at the bottom level are from background objects which are less helpful. However, when continually adding new features, relevant features at the bottom level will be raised if new correspondences are found. Therefore, with this algorithm, we obtain a hierarchically structured set with scalable quality.

2.2.2 Geometric Verification using the Multi-View Fundamental Matrix

The level raising algorithm allows us to add new features to the existing set. However, the results of descriptor matching may contain outliers. Therefore, a suitable geometric verification is necessary for the level raising algorithm. In particular, it needs to accommodate the multi-view camera scenario.

In this work, we use the epipolar constraint as the geometric constraint. The n-view epipolar constraint can be expressed as

$$p\,\mathcal{F}_n\,q^T = 0, \tag{2}$$

where $p = [x_1, y_1, 1, x_2, y_2, 1, \dots, x_{n-1}, y_{n-1}, 1]$ is the image coordinate vector of the feature correspondences in $1, \dots, n-1$ views, $q = [x_2, y_2, 1, x_3, y_3, 1, \dots, x_n, y_n, 1]$ the vector in $2, \dots, n$ views. $\mathcal{F}_n$ is the n-view fundamental matrix in $\mathbb{R}^{3(n-1)\times 3(n-1)}$. Due

to the underlying symmetry, we write it as an upper triangular matrix

$$\mathcal{F}_n = \begin{bmatrix} F_{1,2} & F_{1,3} & F_{1,4} & \dots & F_{1,n} \\ 0 & F_{2,3} & F_{2,4} & \dots & F_{2,n} \\ 0 & 0 & \ddots & \ddots & \dots \\ 0 & 0 & 0 & \ddots & F_{n-2,n} \\ 0 & 0 & 0 & 0 & F_{n-1,n} \end{bmatrix}, \tag{3}$$

where each $F_{i,k}$ is a two-view fundamental matrix between the i-th and k-th view. Further, each non-zero block matrix $F_{i,k}$ in $\mathcal{F}_n$ represents an edge between the i-th and k-th vertex [11].

As the number of correspondences N is usually large, we have to work with an over-determined expression. We determine the fundamental matrix by the least square error solution $\mathcal{F}_n^*$ according to

$$\min_{\mathcal{F}_n} \quad \sum_{j=1}^{N} (p_j \, \mathcal{F}_n \, q_j^T)^2, \tag{4}$$

where j indicates the j-th feature correspondence.

However, the dimension of $\mathcal{F}_n$ is growing with the number of views. This leads to high costs for solving (4). Therefore, we propose a recursive algorithm for the estimation of the multi-view fundamental matrix by utilizing the properties of the level raising algorithm. For adding the n-th new viewpoint, we decompose $\mathcal{F}_n$ into two parts, denoted by the known part R_{n-1} and the new part B_n,

$$\mathcal{F}_n = \left[\begin{array}{ccccc|c} F_{1,2} & F_{1,3} & F_{1,4} & \dots & F_{1,n-1} & F_{1,n} \\ 0 & F_{2,3} & F_{2,4} & \dots & F_{2,n-1} & F_{2,n} \\ 0 & 0 & \ddots & \ddots & \dots & \dots \\ 0 & 0 & 0 & \ddots & F_{n-2,n-1} & F_{n-2,n} \\ 0 & 0 & 0 & 0 & 0 & F_{n-1,n} \end{array}\right] = \left[\begin{array}{c|c} \mathcal{F}_{n-1} & \begin{matrix} F_{1,n} \\ \vdots \\ F_{n-1,n} \end{matrix} \\ \mathbf{0} & \end{array}\right], \tag{5}$$

where the left part is R_{n-1} and the right part B_n, such that $\mathcal{F}_n = [R_{n-1}, B_n]$.

R_{n-1} contains the $(n-1)$-view fundamental matrix $\mathcal{F}_{n-1}$. It essentially represents the geometric constraints of the previous level which are unchanged. The new part $B_n = [F_{1,n}, \dots, F_{n-1,n}]^T$ represents the new geometric constraints between the views $1, \dots, n-1$ and the new view n which is introduced by the set of new features S_n. Therefore, we only need to accurately estimate B_n when adding a new view.

As B_n contains $n-1$ fundamental sub-matrices, we estimate each fundamental matrix $F_{k,n}$ individually. Note that we do not make any assumptions such as knowing the intrinsic parameters of the cameras. Hence, we do not utilize the dependencies among the fundamental matrices like [11]. After applying the level raising algorithm, we get a set of new feature correspondences $\{C^l_{i,\dots,j,n}\}$ where the indices $i, \dots, j \in [1, \dots, n-1]$ and $l \in [1, \dots, n]$. Now, we extract all correspondences between the views k and n with

$$C^*_{k,n} = \{(f_k, \, f_n) | f_k, \, f_n \in \bigcup_{l=2}^{n} \bigcup_{i,\dots,j} C^l_{i,\dots,j,\mathbf{k},\mathbf{n}}\}, \tag{6}$$

where $C^*_{k,n}$ denotes the accumulated set of correspondences between the views k and n. We apply the epipolar-constrained RANSAC algorithm [12] on $C^*_{k,n}$ and obtain a reliable estimate of the fundamental matrix $F_{k,n}$. The outliers in S_n are placed on the bottom level.

With that, we only need to update the sub-matrix B_n when adding the n-th viewpoint. This recursive algorithm allows us to implement the multi-view geometric verification at low computational complexity.

2.3 Progressive Query Matching

With the hierarchical structure of the feature sets at the server, we are able to progressively match query features. As the hierarchical structure leads to a scalable quality of features, we implement a best-feature-first matching strategy by using the features at the highest level first. Therefore, this top-to-bottom matching process has a similar structure than the feature adding process in Section 2.2.

Let Q_c denote the set of query features which contains $N = |Q_c|$ features. As the sets of features at the server is hierarchically structured, we always choose the features in a top-to-bottom order. For each object on the server, we pick up to N feature correspondences. Therefore, the computational load among all objects is also balanced. Due to our strategy, only a small number of server features is used for matching. We use the nearest-neighbor criterion for matching, where the representative descriptors are generated by (1), followed by geometric verification with the epipolar-constrained RANSAC.

Therefore, we count the number of matched features between the query and each object on the server. We define ν as the minimum number of matched features after geometric verification. The corresponding object will be chosen when the number of correctly matched features satisfies the threshold ν. Otherwise, we sort the matching results and choose the best candidate.

In general, at least eight correspondences are needed for computing the fundamental matrix [13]. Let $\widetilde{Q_c} \leftrightarrow \widetilde{Q_s}(l)$ denote the feature correspondences between query and server at level l, where $\widetilde{Q_c} \subset Q_c$ are the matching correspondences of the query and $\widetilde{Q_s}(l) \subset \{C^l_{i,j,\dots,k}\}$ that of the server at level l. However, as $\{C^l_{i,j,\dots,k}\}$ is a set of feature correspondences with different image indices, we cannot apply the epipolar-constrained RANSAC directly. Considering the i-th view, we align all corresponding features at level l associated with the i-th view for the RANSAC

$$Q_s(l,i) = \{f_i | f_i \in \bigcup_{j,\dots,k} C^l_{\mathbf{i},j,\dots,k}\}. \tag{7}$$

Then, geometric verification will determine the matching correspondences for the i-th view $\widetilde{Q_s}(l,i) \subset Q_s(l,i)$. Finally, the set of matching correspondences at the server $\widetilde{Q_s}(l) = \bigcup_i \widetilde{Q_s}(l,i)$ is simply the collection across all views at level l.

3 Experimental Results

We evaluate our hierarchically structured multi-view feature set for the multi-view image dataset *Stockholm Buildings*[1] which comprises 50 buildings of that 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 acquired server and test images at different viewpoints and at different times. The images have been recorded by a Cannon IXUS50 digital camera at a resolution of 2592×1944 pixels.

The query features are selected and encoded with the rate-constrained feature selection method from our earlier work [3]. It utilizes stereo features to obtain more reliable query features. Note, this rate-constrained feature selection differs from single-view feature-based methods as discussed in prior frameworks [1]. The advantage of stereo features is explained in [3].

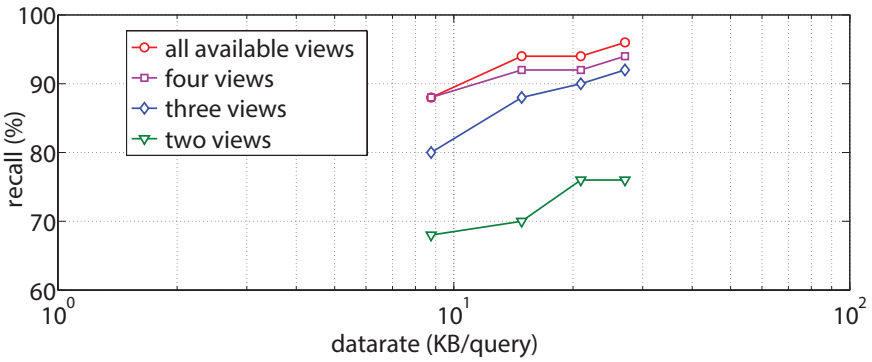

Figure 5: Comparison of the recall-datarate trade-off using hierarchically structured sets of features. The total number of images at the server for two views is 100 images, for three views 150 images, for four views 200 images, and for all available views 254 images.

3.1 Recall-Datarate Performance

We investigate the trade-off between recall and datarate for hierarchically structured multi-view feature sets. 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].

To evaluate the recall-datarate performance, we adjust the number of available views at the server. As shown in Fig. 5, the recall for queries of the same datarate is increasing when adding features from new views to the hierarchical sets at the server. 12% is added to the recall rate when using three views instead of two, and up to 8% when using four views instead of three.

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

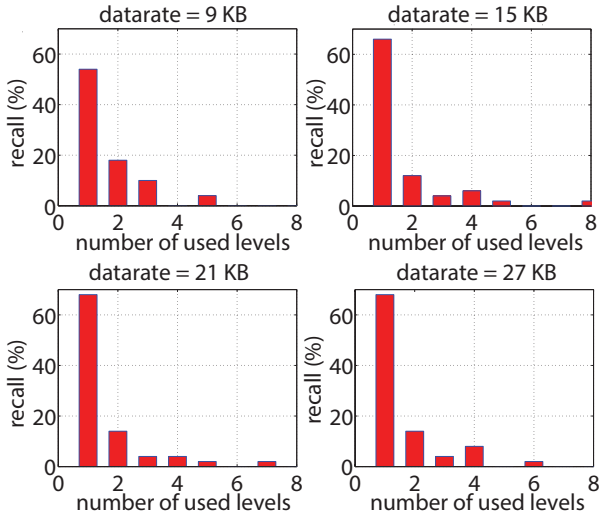

Figure 6: Increment of recall for matching with a given number of used levels.

3.2 Efficiency of Query Matching

With our hierarchical structure of feature sets, we are able to apply a best-feature-first matching strategy for query matching. Therefore, we evaluate the efficiency of our hierarchical structure for a given number of used levels. For this experiment, we use the whole dataset with all available views (254 images). We determine the increment of recall for matching with a given number of used levels.

As shown in Fig. 6, the increment of recall for matching with the top level only is significantly higher than that with the other levels. More than 50% of the queries are matched correctly on the top level. This confirms that the higher levels contain more reliable and representative features when compared to those of the lower levels. Thus, the best-feature-first matching strategy efficiently utilizes our hierarchical structure.

4 Conclusions

We discussed hierarchically structured multi-view feature sets for mobile visual search. The feature correspondences in the multi-view imagery are modeled as complete graphs. Further, the multi-view feature sets are hierarchically structured for augmentation and recall. With a level raising algorithm and multi-view geometric verification, we can efficiently add features from new viewpoints. This improves the reliability of the feature set at the server. Moreover, the hierarchical structure offers feature sets of different qualities. With that, we can progressively match search queries by utilizing a best-feature-first strategy. The experimental results show that our hierarchically structured feature sets improve the recall-datarate performance of mobile visual search. Future research may incorporate 3D geometric information and more compact feature descriptors, such as CHoG [14].

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.

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*, Jun. 2011.

[3] H. Li and M. Flierl, "Mobile 3D visual search using the helmert transformation of stereo features," in *Proc. of the IEEE International Conference on Image Processing*, Sep. 2013.

[4] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. of the International Conference on Computer Vision*, Oct. 2003.

[5] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2006.

[6] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, R. Vedantham, R. Grzeszczuk, and B. Girod, "Inverted index compression for scalable image matching," in *Proc. of the IEEE Data Compression Conference*, Mar. 2010.

[7] B. Pires and J. Moura, "Feature matching in growing databases," in *Proc. of the IEEE International Conference on Image Processing*, Sep. 2012.

[8] P. Turcot and D. Lowe, "Better matching with fewer features: The selection of useful features in large database recognition problems," in *Proc. of ICCV Workshop on Emergent Issues in Large Amounts of Visual Data*, Oct. 2009.

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

[10] P. Huber and E. Ronchetti, *Robust Statistics*, 2nd ed. New York: Wiley, 2009.

[11] N. Levi and M. Werman, "The viewing graph," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2003.

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

[13] R. Hartley, "In defense of the eight-point algorithm," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580 –593, 1997.

[14] 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*, Jun. 2009.