Efficient Indexing for Strongly Similar Image Retrieval

Roth, Gerhard; Scott, William

*Indexing for Strongly Similar Image Retrieval**

Roth, G., and Scott, W.R.
2007

* Proceedings of the Fourth Canadian Conference on Computer and Robot Vision (CRV 2007). 2007. NRC 49341.

Efficient indexing for strongly similar subimage retrieval

Gerhard Roth William Scott
Institute for Information Technology, National Research Council Canada
Montreal Road, Building M50, Ottawa, Ontario, Canada K1A 0R6

Abstract

Strongly similar subimages contain different views of the same object. In subimage search, the user selects an image region and the retrieval system attempts to find matching subimages in an image database that are strongly similar. Solutions have been proposed using salient features or "interest points" that have associated descriptor vectors. However, searching large image databases by exhaustive comparison of interest point descriptors is not feasible. To solve this problem, we propose a novel off-line indexing scheme based on the most significant bits (MSBs) of these descriptors. On-line search uses this index file to limit the search to interest points whose descriptors have the same MSB value, a process up to three orders of magnitude faster than exhaustive search. It is also incremental, since the index file for a union of a group of images can be created by merging the index files of the individual image groups. The effectiveness of the approach is demonstrated experimentally on a variety of image databases.

1 Introduction

In this paper we use salient regions (image locations with distinctive local intensity changes [6, 2]) to solve the subimage query problem: "Given a subimage of a particular image find any strongly similar subimage in an image database". Strongly similar means that the two matching subimages are different views of the same object, subject to changes in scale, illumination, viewpoint or partial occlusion. Though simpler than general image retrieval, strongly similar subimage retrieval has numerous applications in copyright protection, forgery detection, robotics and surveillance. Significant progress has been made by using scale and orientation invariant interest points [6, 1, 9]. Interest points and local descriptors are computed off-line for each image in a database. Then, in an on-line image retrieval process, the user manually selects a subimage in an image and initiates a search for similar subimages in the entire image database.

The on-line search process is as follows:

1. Collect the interest points and their descriptors in the manually selected search image sub-region (green rectangular region in Figure 1).
2. For each interest point descriptor in the selected subregion (the red points in Figure 1) find the closest matching descriptor in every image in the database.
3. Apply geometric constraints to remove any matches that are inconsistent.
4. Rank the searched images by the number of matching interest points.
5. Present the images to the user in ranked order by similarity.

Since each image typically has hundreds of interest points, finding the closest matching interest point descriptors is the bottleneck in the process. More efficient search requires a good solution to the closest point problem (step 2 above). Unfortunately, interest point descriptors have high dimensionality, in which case the performance of closest point search algorithms deteriorates [13].

Our interest point descriptor is a twenty dimensional signed integer vector [5]. We use the concatenation of the most significant sign bit of this vector as a hash function to efficiently find similar interest points. The MSB hash function h takes a single descriptor vector v and produces a signed integer k with i bits consisting of the most significant bit of the first i elements of v. Limiting the closest point verification process to searching only the descriptors that have the same MSB hash function value produces about 10% of the matches found by exhaustive search. Since there are many more matching interest points found by the exhaustive search than necessary for successful image retrieval, searching only the descriptors with the same MSB hash function value usually does not change the order of the retrieved images.

To implement this strategy, we create an inverse index file to quickly find all the descriptors that have the same MSB hash function value. This greatly reduces the search

Figure 1. Subimage Retrieval Example. Top left, selected image subregion with feature points; other quadrants - matching subregions in retrieved images.

time and makes it possible to search large image databases. It is possible to merge together the index files of a number of smaller image databases to create the index file for the larger image database defined by their union. The closest work to ours is [5] which describes a hash function for PCA SIFT feature detectors. However, it is not possible to merge index files in a simple fashion using Ke's hash function.

2 PCA SIFT Corners

Currently one of the most successful interest point detectors is the SIFT operator [6]. It uses a Difference of Gaussians (DOG) detector to isolate the location, scale and orientation of each interest point. A descriptor is computed from the gradients of the image patch around the interest point at the computed scale and orientation. A normalized histogram of these image gradients is found and a 128 element vector of these gradient orientations is used as the descriptor vector for this interest point. While the SIFT operator works well, the high dimensionality of the descriptor vector makes it difficult to create an efficient indexing scheme.

A recent attempt to reduce the dimensionality of the SIFT descriptor vector is the PCA-SIFT interest point [5]. This method uses the same DOG operator to find the interest point location, orientation and scale, but computes a different descriptor from the local image patch surrounding the interest point. Instead of a 128 dimensional histogram of gradient orientations for the SIFT operator, the PCA-SIFT method performs a Principal Component Analysis (PCA) on the image patch gradients. PCA analysis is a commonly used technique for dimensionality reduction that has often been applied to characterize image patches which strong redundancy, such as frontal views of faces. The local image patches surrounding each interest point are normalized so that their dominant orientation is in the same direction, which creates the redundancy that makes PCA effective. Then this normalized local gradient image patch is transformed into a 41 by 41 vector whose dot product is computed with the 20 pre-learned PCA basis vectors. The dot product produces a signed 20 element integer vector which is the descriptor vector for that interest point.

PCA-SIFT descriptors have three distinct advantages compared to the standard SIFT descriptor. First, the dimension of the descriptor is reduced from 128 to 20, resulting in faster nearest neighbour computations[1]. Second, the PCA representation is hierarchic, while the original SIFT representation of gradient orientations is not. Therefore, the PCA descriptor vector elements are naturally ordered in terms of

[1] A systematic comparison of interest point descriptors can be found at [9].

significance from the most to least significant element. To compare two PCA-SIFT descriptors, it is not necessary to check 20 elements; it may be sufficient to compare the first 10 to 15 most significant vector elements. This can further speed up the search process at the cost of some loss in search accuracy. Third, when comparing two PCA-SIFT descriptors, it is possible to use a fixed threshold to decide whether the Euclidean distance (L_2) between the two descriptor vectors are close enough to be considered a match. By contrast, the dynamic threshold used to compare SIFT descriptors [6] makes it more difficult to create an index structure for SIFT than for PCA-SIFT descriptors.

3 Most Significant Bit Matching Filter

The key insight for our indexing method is that, when two descriptor vectors are close enough in Euclidean distance to be a successful match, the most significant bits (MSB) of their descriptor vector elements also often match. This MSB filter can be used to speed up the exhaustive search process by simply not computing the true L_2 distance between descriptors that do not pass the filter test. Since the PCA SIFT descriptors are hierarchic, as the number of compared MSBs decreases, the chance of two descriptors having the same MSB increases. Using the MSB filter in this way does reduce the number of Euclidean distance computations, but this does not always produce a significant reduction in execution time.

3.1 Index File Creation and Modification

A better idea is to use the MSB filter to create an index file by taking the PCA-SIFT interest points and descriptors and grouping together all the descriptors which have the same MSBs. The granularity of this indexing process can be set by selecting how many MSBs are used as an index. In our implementation, four sets of index files are created, for the first $8, 12, 16$ and 20 MSBs. When indexing by a smaller number of MSBs, more descriptors will have the same MSB index value than when indexing by a larger number of MSBs. This makes it more likely that the correctly matching descriptor will be found but more descriptors will need to be compared.

To create the index file structure for a given number of MSBs, the following process is used. First, all the interest point descriptors with the same MSB hash key are grouped together. Then an MSB Descriptor File is created which has these groupings in sorted order, from the smallest to the largest MSB key. Finally the actual MSB Index File, as shown in Figure 2, is created with all the possible MSB hash keys for the image database along with a pointer into the MSB Descriptor File for each hash key. Since the MSB Index File contains at most 2^i entires, where i is $8, 12, 16$ or 20, it can be held in memory, while the larger MSB Descriptor File remains on disk.

Retrieval of interest point descriptors with the same MSB hash key is done by direct lookup using this in-memory MSB index file, followed by a disk read of contiguous interest point descriptors in the MSB Descriptor File. Since the MSB Descriptor File is sorted by hash key value, reading the descriptors for a number of hash keys can be done by sequentially sweeping through this Descriptor File. Euclidean distance comparison is done only for the set of returned MSB descriptors, which is a small subset of all possible descriptors.

It is trivial to modify this Index File structure to add or delete images to the database. Removing image number r from an image database means deleting all it's descriptors, which is done as follows:

1. Remove all entries in the MSB Descriptor File whose image number is r.
2. Using the modified MSB Descriptor File, recompute the MSB Index File.

This process can be easily expanded to delete the descriptors for a set of image files in the image database. Similarly, adding the descriptors for image number r to the image database can be done as follows:

1. Create the MSB descriptors and indices for image r from the PCA-SIFT descriptor.
2. Merge these MSB descriptors and indices for image r with those in the MSB Descriptor file and MSB Index file.

In both the removal and addition operations the running time is proportional to the number of descriptors in the MSB Descriptor Files and the number of entries in the MSB Index Files.

A common situation in image search is to search the union of a set of previously searched image databases. One way to do this is to recompute the MSB Descriptor and Index Files for this union. However, using our indexing structure it is possible to merge all the individual Descriptor and Index Files to create one large MSB Descriptor and Index File. Assume there are p distinct image databases with MSB Descriptor Files, $MSBD(i)$, for $i = 1 \dots p$, and similarly there are p MSB Index Files, for $MSBI(i)$ $i = 1 \dots p$. To create one MSB Descriptor File and one MSB Index File for the union of these image databases, the following steps are required:

1. Open each MSB Descriptor File, $MSBD(i)$, for $i = 1 \dots p$, and each MSB Index File, $MSBI(i)$, for $i = 1 \dots p$.

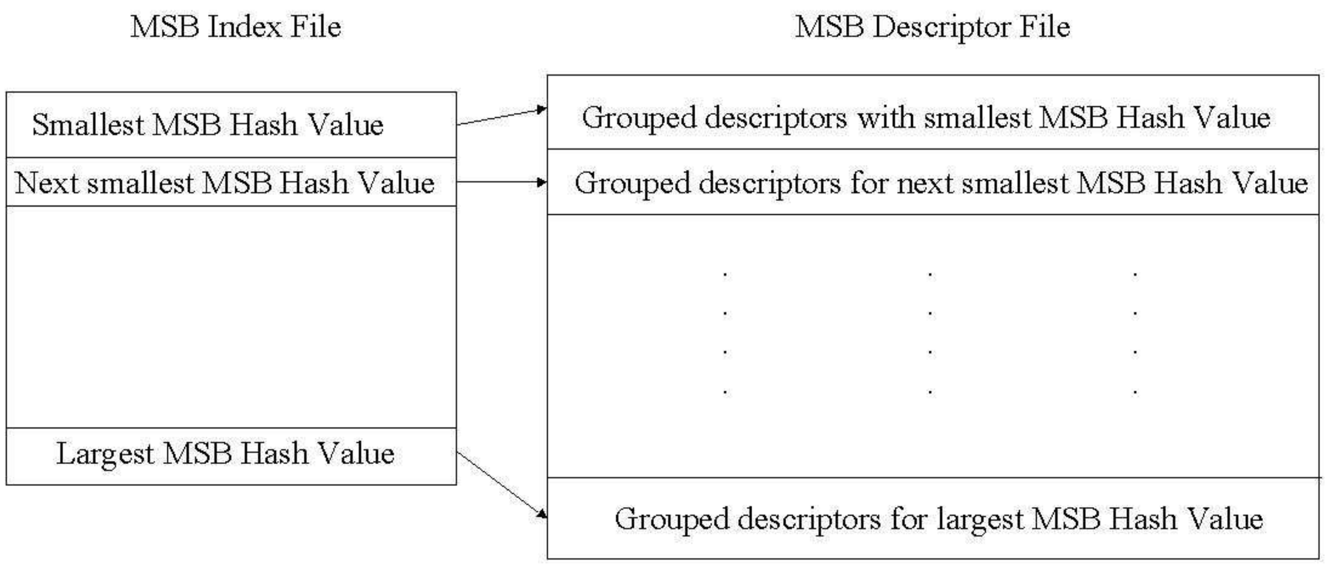

Figure 2. MSB Index File and the MSB Interest Point Descriptor File

2. Find the MSB Descriptor File with the smallest MSB hash key k.

3. Take all the descriptors in all the p files with that MSB hash key k and save them to their new MSB Descriptor File in image order.

4. Create the new MSB Index File entry for the MSB hash key k.

5. Repeat steps 2 to 4 until all the p MSB Descriptor Files are empty.

This merging process takes space $O(p)$, and time proportional to the total number of descriptors and index file entries of the p image databases. This ability to simply and quickly merge previously created index and descriptor files to readily search the union of a set of image databases is unique to our approach.

4 Experimental Results

In this section, we present the performance of the proposed indexing method for strongly similar image search using a number of publicly-available data sets with differing image characteristics. We compare the retrieval performance of various levels of MSB-indexed search with results obtained by exhaustive search.

4.1 Evaluation Metrics

A variety of metrics[2] have been used to evaluate the performance of content-based image retrieval systems [3]. The most common metrics, precision and recall, have been used to evaluate retrieval systems returning a set of putative matches, where the number of matches is usually small relative to database size. In contrast, our system returns all images in the database, ranked in merit order.

We use the Normalized Modified Retrieval Rank (NMRR) metric for quantifying retrieval performance which is part of the MPEG7 standard [7]. This measure is independent of database size N or number of ground truth images $N_G(q)$ for a query q. The measure is intuitive to users whose experience is grounded in web-based search. NMRR is based on the concept that a user assigns value to the ranking of valid retrievals up to some limit which is a function of $N_G(q)$, beyond which retrievals are considered equally useless. All valid retrievals up to this limit are considered to have equal merit. Therefore the rank r_k of the k^{th} retrieved image is in the range of $[1, N]$. Using the NMRR metric a modified retrieval rank of $K(q) + 1$ is assigned to any ground truth retrieval not in the first $K(q)$ retrievals for query q. As the ground truth size $N_G(q)$ generally varies between queries, the constant $K(q)$ is set by:

$$K(q) = \min(4 * N_G(q), 2 * GTM) \qquad (1)$$

[2]Note, that Müller et al's definition for normalized average rank at equation (4) of [3] is weak in that it does not properly normalize average rank.

where GTM is the maximum $N_G(q)$ over all queries. Then, NMRR is defined as:

$$NMRR(q) = \frac{\mu(q) - 0.5 - 0.5 * N_G(q)}{K(q) + 0.5 - 0.5 * N_G(q)} \qquad (2)$$

where $\mu(q)$ is the unnormalized average retrieval rank defined as:

$$\mu(q) = \frac{1}{N_G(q)} \sum_{k=1}^{N_G(q)} r_k \qquad (3)$$

$NMRR(q)$ has range $[0, 1]$, with 0 indicating perfect retrieval (all ground truth images within the first $N_G(q)$ retrievals) and 1 indicating no valid returns in the first $K(q)$ retrievals. The smaller the value of $NMRR(q)$, the better the retrieval performance. Finally, Average Normalized Modified Retrieval Rank (ANMRR) is the average retrieval quality over a series of Q queries in a session:

$$ANMRR = \frac{1}{Q} \sum_{q=1}^{Q} NMRR(q). \qquad (4)$$

4.2 Experimental Procedure

The first two experiments test the ability to search image databases where the variations are due only to two-dimensional transformations, or to very limited changes in camera viewpoint. In the literature such matching images are called "near-duplicates" of each other. The image set cgaset12 (http://www.cs.cmu.edu/˜yke/retrieval) consists of 30 works of art, each subject to 20 two-dimensional corrupting transformations (noise addition,rotation, downsampling, etc. [8, 5]). This data set represents the problem of copyright violation detection, where the matching images are basically copies subject to corrupting two-dimensional transformations due to differences in digitization. This 6000 image database was queried using the all thirty baseline images as test cases. These tests resulted in perfect retrieval ($ANMRR = 0$) for all reference images for all indexing cases (exhaustive search through to 20 MSB). The second experiment in near-duplicate detection is in finding near-duplicates for linking multi-media content [4]. The images are from the well known TRECVID video data sets (http://ww-nlpir.nist.gov/projects/trecvid) which were created for the purpose of algorithm evaluation. In this case images are near-duplicates if they are pictures of the same news event taken from a slightly different viewpoint. This situation occurs often in multi-media reporting since many cameras present at events such as news conference are taking slightly different views of the same person or object. The images in the Columbia University near-duplicate image dataset consist of 150 duplicate pairs from the TRECVID data set to create an image database of 300 images. We extracted the duplicate images directly from the Columbia University web site (http://www.ee.columbia.edu/˜dqzhang). The experiments consisted of using each of the 150 images as a query image, and the only acceptable correct result would be the retrieval of the correct duplicate image of the pair. The tests again resulted in perfect retrieval for all 150 reference image queries ($ANMRR = 0$) for all indexing cases (exhaustive search through to 20 MSB).

Database	N	$[N_G]$	$\overline{N_G}$	Q
zubudfull	125	4-4	4	20
cars1	526	3-25	10.1	20
californiaCoast	1500	2-11	5.3	10
auton	345	8-89	38.9	8
allpred	218	5-27	14.8	10
nister-2000	2000	4-4	4	10
nister-4000	4000	4-4	4	10
nister-6000	6000	4-4	4	10
nister-8000	8000	4-4	4	10
nister-10000	10000	4-4	4	10

Table 1. Wide-Angle Test Image Databases

Having demonstrated good retrieval performance for the problem of near-duplicate image selection we next conducted retrieval experiments with ten image databases (Appendix A). The images could no longer be labeled as near-duplicates, but had more significant changes in viewpoint and scale. We call this set of image databases the wide-angle experimental datasets, since the images were taken with a wider angle between them than the near-duplicate datasets. Table 1 provides summary data on these experiments: image database size N, ground truth range $[N_G]$, average ground truth size $\overline{N_G}$ and number of queries Q. The first five data sets in Table 1 are small (less than 2000 images) and come from a number of different sources. Since some of these data sets involved substantial perspective changes between images, manually choosing the ground truth was at times difficult. For an image to be included in the ground truth set, our rule was that it had to include at least 50% of the targeted sub-region.

Recently a data base of $10,000$ images was created for testing object recognition algorithms which consists of four view of each of $2,500$ objects [10]. Since the same object is seen from four different viewpoints the ground truth data is known, which makes this data set ideal for testing similarity search algorithms. The data set was subdivided into five databases ranging from 2000 to 10000 images respectively (the last five datasets in Table 1).

Our experimental procedure was as follows. One image from the database was chosen as the query reference. In the nister data sets reference case retrieval was executed against the entire images, but in the other wide angle data sets retrieval was against a manually selected sub-image. Each query was run against the data sets for each of five indexing cases (exhaustive search, 8 MSB, 12 MSB, 16 MSB and 20 MSB). All correct retrievals were considered equally valid. Image retrieval rank and query execution time were recorded. Subsequently, the $ANMRR$ was computed, as described in Section 4.1. Retrieval experiments were conducted on a 2.6 GHz Pentium 4 PC with 512MB of RAM running Windows XP. Algorithms were implemented in C.

4.3 Image Retrieval Results

Experimental results are summarized in the following figures, where exhaustive search is labeled as $MSB = 0$ in the graphs. The left half of Figure 3 presents ANMRR performance for the first five databases described in Table 1. As expected, retrieval performance declines as the number of matching most-significant-bits used in the indexing process increases. The difficult nature of the auton video is apparent, as even exhaustive search provides relatively poor retrieval performance ($ANMRR = 0.3983$). The results for the last five data sets (nister sequences) are also shown in the right half of Figure 3. These data sets are also difficult because they contain many images, and the four views of each object often have significant perspective differences.

In the left half of Figure 4 we show the improvement in search time relative to exhaustive search for the first three image database sets (zubudfull, cars1, californiaCoast) [3]. The right half of Figure 4 shows the improvement relative to exhaustive search for the last five datasets, which are different sized subsets of the nister object collection. This last experiment shows that as the number of images increases the improvement over exhaustive search also increases. Overall a rapid improvement over exhaustive search (up to 250 times) is apparent as the number of matching MSBs increases. However, this must be traded off against a decrease in search performance.

5 Discussions and Conclusions

The ability to index and search large image databases is an important problem attracting the attention of a number of researchers [5, 10, 11, 12]. All the solutions in the literature build an index structure of-line, and then to use this index structure to speed up the on-line search process. For truly large scale search the index structure must be on disk, since approaches based on structures such as a kd-tree that hold all the keypoints in memory are not scalable [6]. The main distinction between different methods is in the type of index structure, and how it is created. Our MSB indexing approach has the advantage of having a very simple index structure, which can be created in the order of a minute for ten thousand images. With this approach index files from different image databases can easily be merged together to create the index file to search the union of all these images. As far as we know this ability is unique to the MSB indexing method. In theory the retrieval performance of systems that have a more sophisticated index structure should be superior to ours, but in the experiments so far our system performs comparably to others on the same databases [4, 5, 12, 11, 10]. Further work will be needed to systematically test the approach on very large image databases of more than a million images. To summarize we have described an indexing system enabling fast image and subimage retrieval. The advantages of this system are that it is simple, incremental, and index files created from different databases can easily be merged.

[3] Search times are not shown for the allpred and auton datasets. The low resolution of these image datasets resulted in detection of sparse feature sets. Consequently, retrieval was so fast that it was overshadowed by random background operating system tasks, rendering execution times statistically unreliable.

References

[1] L. Amsaleg and P. Gros. A robust technique to recognize objects in images, and the db problems it raises. Technical Report No. 1371, IRISA - CNRC, November 2000.

[2] J. Hare and P. Lewis. On image retrieval using salient regions with vector-spaces and latent semantics. In *Fourth International Conference on Image and Video Retrieval*, volume LNCS 3568, pages 540–549, 2005.

[3] H.Muller, W. Muller, D. McGSquire, S. Marchand-Maillet, and T. Pun. Performance evaluation in content-based image retrieval. *Pattern Recognition Letters*, (22):593–601, 2001.

[4] A. Jaimes, S.-F. Chang, and A. Loui. Detecting image near-duplicate by stochastic attribute relational graph matching with learning. In *ACM Multimedia*. ACM, 2004.

[5] Y. Ke, R.Sukthankar, and L. Houston. Efficient near-duplicate detection and sub-image retrieval. In *ACM Multimedia*, pages 1150–1157. ACM, Oct. 2004.

[6] D. Lowe. Distinctive image features from scale invariant viewpoints. *International Journal of Computer Vision*, 60(2), 2004.

[7] B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada. Color and texture descriptors. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 11, pages 703–715, June 2001.

[8] Y. Meng, E. Chang, and B. Li. Enhancing dpf for near-image replication. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[9] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(10):530–534, 2005.

[10] D. Nister and H. Steweius. Scalable recognition with a vocabulary tree. In *IEEE Computer Society Conference on*

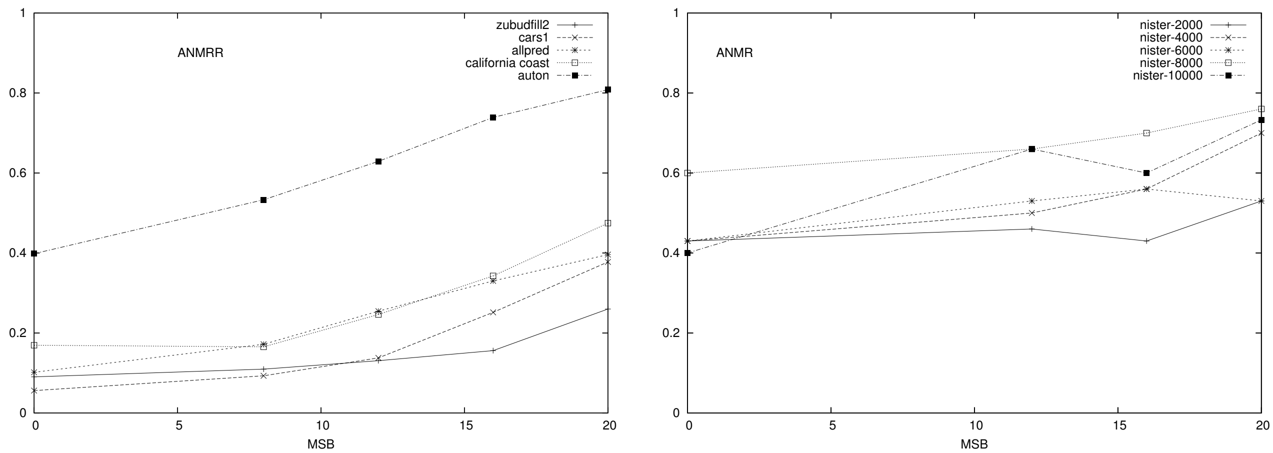

Figure 3. Average Normalized Modified Retrieval Rank (ANMRR) - Wide-Angle Datasets

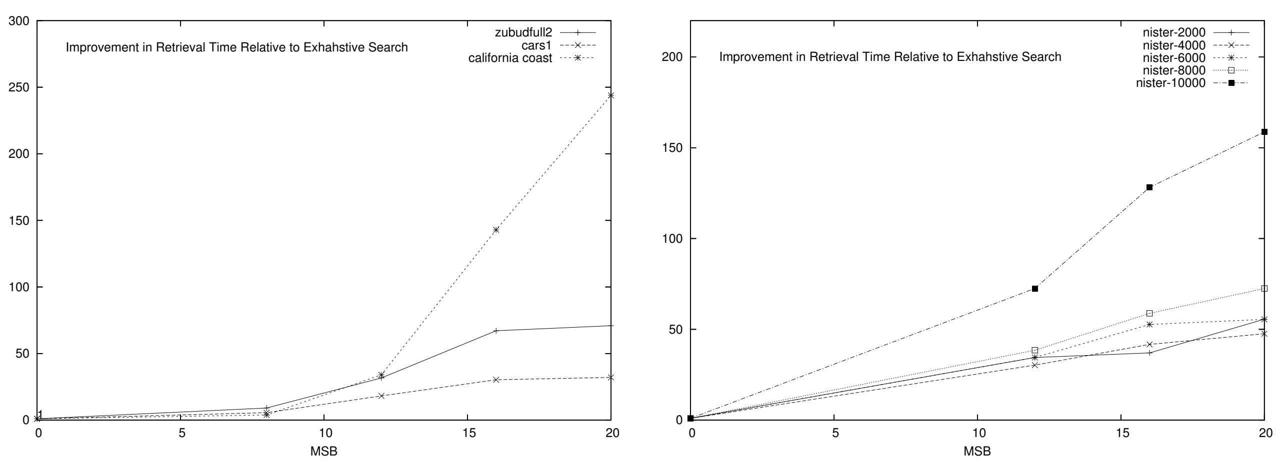

Figure 4. Performance Improvement Relative to Exhaustive Search - Wide-Angle datasets

Computer Vision and Pattern Recognition, New York, New York, June 2006.
[11] S. Obdrzalek and J. Matas. Sub-linear indexing for large scale object recogntion. In *British Machine Vision Conference*, 2005.
[12] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of ICCV 03*, 2003.
[13] M. Smid. Closest point problems in computational gometry. In J. Sack and J. Urruita, editors, *Handbook on Computational Geometry*, pages 877–935. 1997.

A Wide-Angle Experimental Image Datasets

Ten databases were used in the wide-angle retrieval experiments.

cars1 : http://www.vision.caltech.edu/html-files/archive.html contains the data set which consists of rear views of automobiles taken in traffic. Images in a given ground truth set vary mainly with distance/scale change. Perspective changes are limited.

californiaCoast : http://www.californiacoastline.org) contains the data set which consists of views of the California coast taken from a helicopter involving a mix of natural and human structure. Images in a given ground truth set involve modest scale change but typically large viewpoint/perspective changes.

allpred : http://www.cs.cmu.edu/˜vsam/download.html contains the data set which consists of four segments of sub-sampled Predator UAV video. The video is generally of low resolution, limited dynamic range with some artifacts and motion blur. Both scale and perspective changes are involved.

auton : http://www.insitugroup.com contains the data set which involves sub-sampled video from three passes of an Unmanned Aerial Vehicle (UAV) over a target complex in an otherwise empty desert region. This real world video presents a difficult challenge for image-based retrieval. Images involve large perspective changes plus some range/scale variation. Given the large perspective changes, establishing the ground truth was difficult and our determination was perhaps overly conservative.

zubudfull : http://www.vision.ee.etch.ch/showrecon/zubud contains the data set which consists of sets of five images each of 25 buildings in Zurich. Images in a given ground truth set include some camera rotations ($+, -90°$), some scale change plus perspective change that is often quite large (as much as $90°$ in the horizontal).

nister : http:/vis.uky.edu/˜stewe/ukbench contains the data set which consists of four views of 2500 distinct objects taken from a lab and home environment. We divide this data set into five different subsets ranging from 2000 images (500 objects), to 10000 images (2500 objects) in steps of 2000 images (500 objects).