An Image Search System for UAVs

Roth, Gerhard; Scott, William

Publisher's version / Version de l'éditeur:

UVS Canada's Advanced Innovation and Partnership 2005 Conference [Proceedings], 2005

National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de technologie de l'information

NRC · CNRC

*An Image Search System for UAVs **

Roth, G., and Scott, W.R.
November 2005

* published at UVS Canada's Advanced Innovation and Partnership 2005 Conference. Banff, Alberta, Canada. November 16-18, 2005. NRC 48289.

Canada

An Image Search System for UAVs

Dr. Gerhard Roth and Dr. William Scott
Institute for Information Technology, National Research Council Canada
Montreal Road, Building M50, Ottawa, Ontario, Canada K1A 0R6
Gerhard.Roth@nrc-cnrc.gc.ca, William.Scott@nrc-cnrc.gc.ca

***Abstract*— UAVs are becoming ubiquitous and will be widely deployed in many applications. The result will be a large amount of video data that needs to be organized and searched. A critical image processing application for UAVs will be a Google-like real-time search engine for large image and video databases. We have developed a novel indexing and search method that can be applied to large video databases. This technology enables a user to search for strongly similar images in a large image database created by intelligently subsampling the video sequences. The same image search technology can also be used for the automatic mosaicking of a set of unordered images to create a large panorama. A number of experiments with video taken from UAVs demonstrate the technology.**

I. Introduction

Applications for unmanned autonomous flying vehicles (UAVs) are rapidly increasing in number in both the military and civilian fields. A common thread among them is the fact that they all have some onboard image sensors which are essential to vehicle functionality. There are two modes of operation for these sensors: on-line for the real-time control of the vehicle, and off-line for the purpose of collecting surveillance and reconnaissance information. Clearly UAVs will collect a vast amount of video data that will be stored for further off-line processing. Therefore, a critical task will be the ability to organize and search this video data in a meaningful fashion. This problem will exist regardless of the particular details of the UAV technology that is used to collect the image data. The obvious solution of indexing UAV images via navigation data will not be enough. One reason is that the viewpoint of the onboard camera may be controllable so different views are recorded even from the same GPS location. Even in the best of cases a GPS location, when it is available, is only approximate. Therefore in this paper, we assume there is no GPS information associated with the archived video data.

UAVs will scan an environment in a repetitive fashion and these surveillance videos will accumulate over months and years. At a future date an operator will want to find certain images or parts of images that are approximate matches. This means there is a requirement for an image search application that can match different views of an object that are not identical, but instead, are what we call strongly similar. Two images are strongly similar if they are different views of the same object taken from approximately the same viewpoint, but under different lighting, scale and occlusion conditions. For example, in Figures 1 and 2 we see scenes under different scale and occlusion conditions. If the leftmost picture in each of these four images was selected, then the other three should be returned by a matching process as being the same picture. They are, according to our definition, strongly similar.

If strongly similar images taken from a UAV can be matched by an operator, then it will be possible to identify how the environment has changed or evolved over time.

Along with matching strongly similar images, such a search process must also be able to match subimages, and not just entire images. For example, an operator may manually select as a search region part of an image instead of selecting the entire image. Then the goal of the search algorithm is to find all image fragments in the entire image database that match this subimage in a strongly similar fashion. So the search process must be able to deal with strongly similar, but not identical images, and also be able to match image subparts, and not just entire images. An example of this search process in operation is shown in Figure 3

To achieve this goal requires the solution of two problems. First, the video sequence taken from the UAV must be converted into a sequence of significant key-frames, which are individual images that compactly summarize the video. The second is that these key frames, which could consist of tens of thousands of images, must be searchable in the strongly similar fashion described. To be able to search such a large number of image frames, it will be necessary to create an inverse index file in Google-like fashion. That is, a set of features must be extracted from each image and there must be an index structure mapping from these features to the images. This is the classic Google-like search process, except that it is image based. While such index based image based search methods do exist, they currently can not match strongly similar images nor can they handle subimage search. However, recently a new generation of search algorithms based on local image features, called interest points, have made good progress on these two fronts. What is unique about our approach is that we have an indexing and search method based on interest points that can be applied to large image databases. The key to success is that an index structure is created off-line which enables fast on-line search for strongly similar image subregions in an image database. Our approach is also scalable to very large image databases, the type that will be produced by UAV collection.

The processing begins with an off-line phase consisting of the following steps:

- Each video sequence is subsampled into significant still images that are a compact representation of that video.
- Each of these images is processed to find features called interest points along with their descriptor vector.

Fig. 1. Four views of a building at different scale.

Fig. 2. Four views of a poster with different levels of occlusion.

- An index file structure is created for the entire image database from these interest points and their descriptor vectors.

This off-line phase produces an index structure which can be used to dramatically speed up the search process by efficiently finding similar interest points across a large number of images. In essence, it groups interest points that have similar descriptor vectors together so that they can be searched efficiently. Our indexing process is approximate in the sense that the resulting search process only finds a subset of all the possible matches. However, because there are normally far more interest points obtained from these images than are necessary for reliable matching, the fact that the indexing process produces only approximate results is not a problem.

The on-line search proceeds in the following fashion:

- The user is presented all the images in the database and selects one as the search image.
- The user then selects a subimage of this search image (which many include the entire image) as the search region.
- The search process uses the index file structure to find all the images in the database which contain a subimage that is strongly similar to this search region.
- These images are ranked from most to least similar and are presented to the user in that order.

The rest of this paper will describe the technology used to achieve this goal. In the Experiments Section a number of different experiments on actual UAV datasets will be described.

II. Off-Line Processing Pipeline

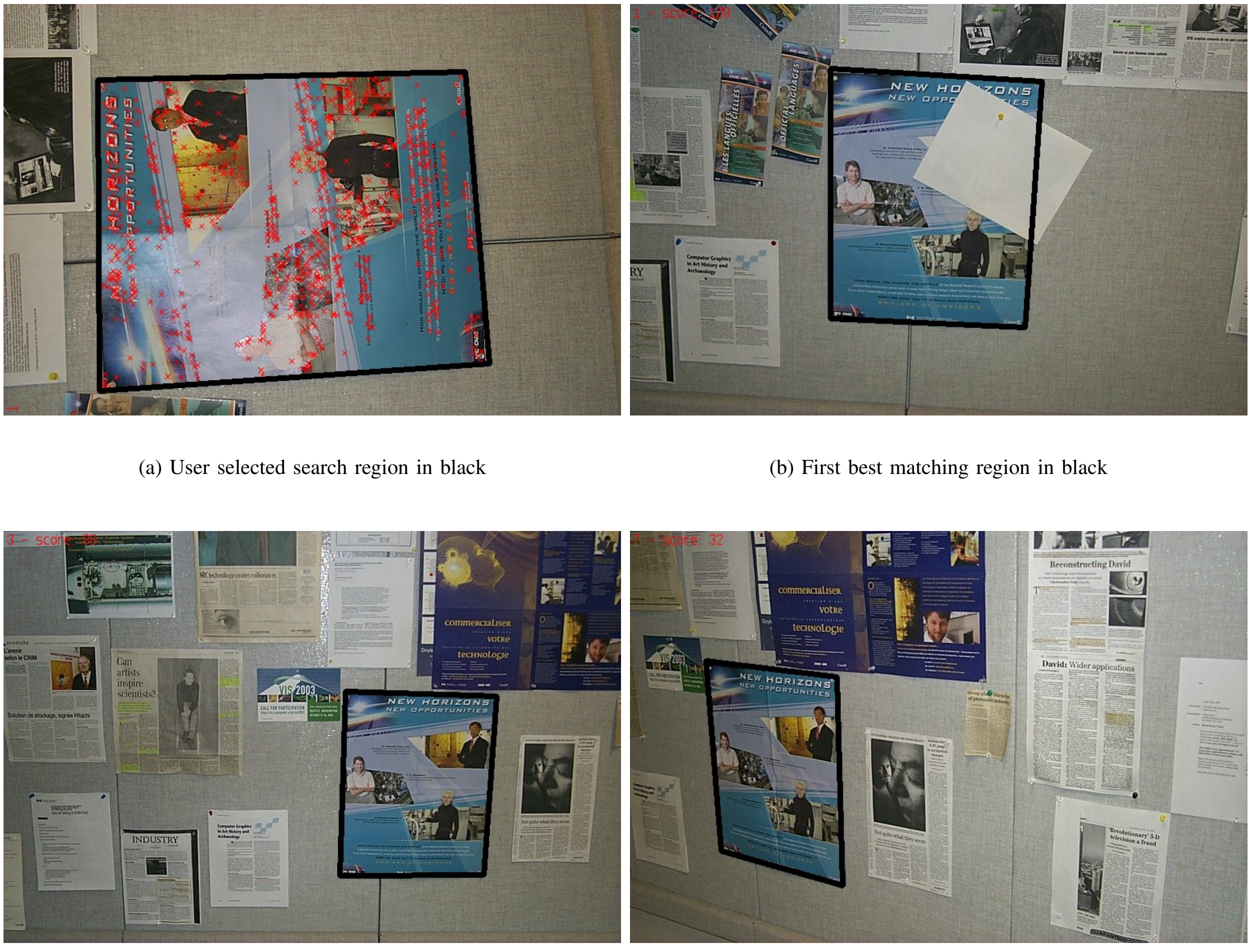
This section describes in more detail the steps in the off-line processing pipeline.

A. *Finding Significant Image Frames in Video*

The first step in processing a set of UAV videos is video summarization. Video is normally obtained at 15 to 30 frames per second, making it impractical to create a database of images using each video frame. It is therefore essential to subsample the video into significant frames representative of the video sequence. UAV video will have different rates of change of the camera point of view at different parts of the video. Therefore, a good strategy is to sub-sample the video so that the change in viewpoint between significant frames is normalized to remove the effect of the different rates of UAV motion.

The approach we have used for video summarization is described in more detail in [1]. The program tracks local features from frame to frame [2], [3]. As the camera viewpoint changes between frames, some of the image features tracked in the initial frame are lost or have moved more than a specified distance in the image. When either condition occurs for a large fraction of the initial features, the initial frame is saved as a significant frame and the base frame for the tracker is reset to the current frame. The effect is to subsample the video in proportion to the speed of camera motion.

On the average, there are three hundred significant frames extracted from a ten minute video sequence. For example, the video named "pred1b" in http://www.cs.cmu.edu/~vsam/download.html is one of the Predator sequences processed in Section IV. It is one minute sequence of a Predator UAV following two vehicles on a road. With a sampling rate of 30 hz, this one minute video sequence has 1800 frames. Thirty significant frames are extracted from this video, a reduction by a factor of 60. Figure 4 shows the last five significant frames extracted from this one minute Predator video. Actual reduction rates vary



(a) User selected search region in black

(b) First best matching region in black

(c) Second best matching region in black

(d) Third best matching region in black

Fig. 3. Subimage retrieval example. Top left, selected image subregion. Other quadrants - matching subregions.

with the type of motion, but our experiments typically show a reduction factor of almost 60.

B. Searching Image Databases

Our goal is to be able to search a large collection of significant images frames produced by the video summarization process in an interactive fashion. Typically, the user is presented with a database of images and selects one of these images as the search image. A subimage or the entire image can be selected. The search is performed on the entire image database. Matching images are presented to the user in order of the best to worst match (e.g. as shown in Figure 3). While the basic image search paradigm is clear, the most important question is what features should be extracted from each image in order to do the search. Most image search systems in operation use global features for the search. By global, we mean that a feature vector is computed using all the image pixels. The most common global feature is a histogram of the frequency of occurrence of different pixel values [4]. Colour histogram search has had some success in image search applications that use global image features [5]. While histograms are invariant to changes in rotation, they are not invariant to changes in illumination. The most important limitation of systems based on global image features is that they are not able to search for matching subimages, only matching images.

We believe it is essential to be able to perform subimage search. This is clearly a requirement in UAV applications. Subimage search requires the use of local feature vectors computed from different parts of an image. Currently, very few image search systems are based on local image characteristics. The next section describe a search technique using local image features with which we have had some success.

III. Interest Points

When considering how to characterize images locally the key insight is to focus on the parts of the image that are "interesting". Consider a checkerboard as an example. In this case, the corners of the checkerboard are clearly the most interesting points. The regions that are purely black or

Fig. 4. Five significant frames from the predator video.

white are not unique, while the edges of the checkerboard are more unique but not as unique as the corners. Thus, if we could find corners in an image, their uniqueness would make them a good set of local features. Mathematically, what makes a corner interesting is the fact that the pixel location has two strong image gradients in orthogonal directions. This definition is imbedded in the classical Harris Corner operator [6]. Applying this operator to a typical image will generally produce hundreds to thousands of pixel locations that are corners.

However, along with its pixel location, it is also necessary to produce a description of each corner point that is useful for matching. This description is a high dimensional vector which is a function of the area surrounding the corner point. There are a number of ways to define this descriptor vector, the most obvious one being the local pixels surrounding the corner point. However, we desire a descriptor invariant to changes in lighting, scale, orientation and illumination and to partial occlusion. Finding an interest point operator and a descriptor which is invariant to all these changes is difficult. Dealing with changes in scale is especially difficult, since it requires that some estimate be made of the natural scale of that interest point. First generation interest point operators such as the Harris corner operator are not invariant to scale changes. A second generation of interest point operators and associated descriptors have recently been developed which are to some degree invariant to these kinds of changes. The most successful of these are the SIFT operator and a variant called PCA-SIFT interest points.

A. SIFT and PCA-SIFT Interest Points

Second generation interest point operators return a sub-pixel location of each point plus an orientation and scale [7], [8]. The local image patch at the given scale and orientation is then normalized and a descriptor is computed. Ideally, descriptor vectors should characterize the interest point region uniquely so that similar interest points have a small Euclidian distance between their descriptor vectors.

Currently, the best known and most successful second generation interest point detector is the SIFT operator [8]. It uses a Difference of Gaussians (DOG) operation to isolate the location, scale and orientation of each interest point. Using the dominant orientation of each interest point, a descriptor vector is computed from the gradients of the surrounding image patch at the computed scale and orientation. A normalized histogram of these image gradients is computed and a 128 element vector of gradient orientations is used as the descriptor vector for the interest point. While the SIFT operator works well, the high dimensionality of the descriptor vector makes it difficult to create an efficient indexing scheme for image matching.

A recent attempt to reduce the dimensionality of the SIFT descriptor vector is the PCA-SIFT interest point [9]. 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 algorithm performs a Principal Component Analysis (PCA) on the gradients of the image patch for a large number of interest points. The PCA analysis produces 20 basis vectors that characterize the typical normalized image gradient patch. For each interest point, a dot product of the normalized image gradient patch with the pre-learned 20 PCA basis vectors is computed and stored. The result is a signed 20 element integer vector which is the descriptor vector for that interest point.

There are three advantages to using PCA-SIFT as compared to the standard SIFT. [1] First, the dimension of the descriptor is reduced from 128 to 20 resulting in much faster matching. Second, the PCA representation is hierarchic, while the original SIFT representation of gradient orientations is not. This means that each of these descriptor vector elements are naturally ordered in terms of their importance. Therefore, to compare two PCA-SIFT descriptors, it is often sufficient to compare only the first 10 to 15 most significant vector elements. This dimension reduction can further speed up the search process at the cost of some loss in search accuracy. Thirdly, when comparing two PCA-SIFT descriptors, it is possible to use a fixed threshold to decide whether these two descriptor vectors are close enough to be considered a match. The usual method of comparing SIFT descriptor vectors is by the ratio of the first to second best match distance relative to a threshold. Creating an index structure for a fixed threshold match comparison is easier than for a dynamic threshold. For these reasons we use the PCA-SIFT interest point operator in our strong similarity image retrieval system.

[1] A systematic comparison of different types of descriptor vectors is found in [7].

B. Exhaustive Search Using Interest Points

A subimage retrieval system can be built using PCA-SIFT interest points and their associated descriptor vectors. The first step is to compute the interest points and descriptors for a database of images off-line and to store them in a file associated with each image. The user manually selects a subimage to search for and the on-line system initiates a search for similar subimages in the image database.

The basic on-line search algorithm is as follows.

1) Retrieve the interest points and their descriptors in the manually selected search image sub-region.
2) For each interest point descriptor in the selected image subregion, find the closest matching descriptor in each database image.
3) Apply geometric constraints to remove inconsistent matches (Optional).
4) Rank the searched images by the number of matching interest points in the image.
5) Present the images to the user in ranked order, from most to least similar.

However, there are thousands of interest points and descriptor vectors associated with each image. Consequently, exhaustive search is impractical for databases of more than a few hundred images. The search process requires greater efficiency. Clearly, two descriptor vectors that are strongly dissimilar need not be compared. If similar descriptor vectors could be grouped together, it would be possible to limit the search to only those descriptor vectors that have a chance of being a good match. To achieve this goal requires the creation of a suitable index system for these descriptors. We describe such an approach in the next section.

C. The MSB Index System

The key to our indexing system is a hashing method for finding similar local descriptors in large image databases. The descriptor associated with each PCA-SIFT interest point is a 20 dimensional signed integer vector [9]. Our hash function is simply the concatenation of the most significant sign bits (MSBs) of this vector. We use this hash function to group descriptor vectors having the same MSBs together. Limiting the search process in this way reduces the number of matching feature points. However, there are many more matching interest points found by exhaustive search than are necessary, so the order of the retrieved images is usually not affected when using the MSB hash function. The redundancy in matching interest points makes it possible to use this simple MSB hash function to reduce the search time and still achieve excellent retrieval results.

Our test system uses four different index files; grouping descriptors for 8, 12, 16 and 20 significant bits. The 20 bit descriptor vector produced by the PCA-SIFT process is hierarchical allowing the indexing process to work at different levels of granularity. The smaller the number of MSBs that must match, the better the retrieval performance but the smaller the speedup. This index structure has three important advantages:

1) Creating the index file is easy for large image databases.
2) Index files for very large image datasets can be created by merging the index files of a number of smaller image databases.
3) Search times can be reduced up to three orders of magnitude when searching very large image databases.

The merging process is efficient, being linear in both space and time. The most similar research to ours is [10], which describes a hash function for the same PCA-SIFT feature detectors. However, unlike our MSB hash function, Ke's hash function is not incremental so it is not possible to merge index files created from Ke's hash function in a simple fashion.

D. Index File Creation

An index file is created by the following process. Compute all the PCA-SIFT interest points and their associated descriptors for a given image database. Next, group descriptors with the same MSB together in the file sorted by MSB order. Details of this process are described in [11].

1) Index File Modification: One advantage of this indexing scheme is that it is trivial to modify the index file structure when changes are made. If new images are to be added or deleted to the image database then this requires only a single pass through the MSB index file. Another common situation in image search applications is when the goal is to search the union of a set of previously searched image databases. In this case the individual image databases already have associated MSB index files. Creating one large MSB index file for their union requires only a simple merge process of these MSB index files. The ability to easily modify the MSB index files when images (and their descriptor vectors) are added or removed from the database is an important advantage of our approach.

IV. EXPERIMENTS IN STRONGLY IMAGE SIMILAR RETRIEVAL

The experiments are performed on four UAV datasets. In each case only one search example is chosen, more systematic experiments are described in [11]. Figures 5, 6, 7,and 8 show the search region and the best three results obtained in four different search modes: exhaustive search, and indexed search with 8, 12, 16 and 20 MSBs. The rows in each Figure are the type of search (exhaustive, 8, 12, 16 or 20 MSBs indexed). The first column in each figure consists of the manually selected search region, which is the same in all four cases. The remaining columns show the best three matches returned by this type of search. Table I also shows the execution times for the different databases and search modes, along with the improvement ratio relative to exhaustive search. Consider the case where number of MSBs in the indexed search increases as we move from row two to five in each figure. The table shows that in this case, the execution time decreases while the figures show that search performance decreases. The loss in performance is clear; the first row has the three closest matching images returned by exhaustive search and by comparison in the remaining rows the indexed search process does

Database	Image Size	Number Images	Exhaustive	8 MSB's	Ratio	12 MSB's	Ratio	16 MSB's	Ratio	20 MSB's	Ratio
Predator UAV	320x240	217	0.33	0.08	4.1	0.04	8.2	0.04	8.2	0.04	8.2
Insitu UAV	320x240	344	0.22	0.04	5.5	0.03	7.3	0.03	7.3	0.03	7.3
Georgia Tech	352x240	1591	2.55	0.36	7.1	0.141	18.0	0.061	41.8	0.06	42.5
California Coast	720x469	1500	17.1	2.25	7.6	0.44	38.9	0.22	77.7	0.16	106.9

TABLE I

SEARCH TIMES IN SECONDS AND THE TIME IMPROVEMENT RATIO RELATIVE TO EXHAUSTIVE SEARCH.

not always return the same three matching images. Increasing the number of MSBs in the index file is a trade-off between speed and performance; the search is faster but the retrieval performance deteriorates.

A. Predator UAV Videos

The web site http://www.cs.cmu.edu/˜vsam/download.html has a set of videos taken from a Predator UAV consisting of five minutes of six different scenes. This has been summarized by 217 significant images. Results from a representative search are shown in Figure 5.

B. Insitu UAV Videos

The Insitu Group has a UAV described at http://www.insitugroup.com from which we have processed a video sequence. The video is approximately eight minutes and there is considerable camera motion. This has been summarized into 344 images and the results are shown in Figure 6. Notice that the search region in this example is very small, with only a few interest points, but the results are still good.

C. Georgia Tech UAV Videos

Georiga Tech has a UAV from which we have processed the two video files (obgtar2003attempt2_5x.avi, and obgtar2003attempt3_5x.avi from web site (http://controls.ae.gatech.edu/uavrf/movies) seven minutes long. The video summarization process produces 1591 significant images, which is large because these videos are already subsampled. The results are shown in Figure 7.

D. California Coast Aerial Images

The last dataset is different in that it consists not of a UAV video but a set of indexed aerial images of the California coast (http://www.californiacoastline.org). These images are referenced via GPS and are sequential in the order they were taken along the coast. We have downloaded 1500 such images and the results are shown in Figure 8.

V. CONCLUSION

This paper has described an indexing structure for a general image search system for UAV video. The research platform is operational and is currently undergoing systematic testing. The creation of panoramas from a set of overlapping images is also an important UAV application, and is called mosaicking [12]. The same image search technology is being used for the automatic mosaicking of a set of unordered images to create a large panorama.

ACKNOWLEDGMENT

The authors would like to thank the Insitu group for the use of their UAV video sequence.

REFERENCES

[1] A. Whitehead and G. Roth, "Frame extraction for structure from motion," in *Proceedings of Visualization, Imaging and Image Processing*, Marbella, Spain, September 2001, pp. 658–663.

[2] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[3] S. Birchfield, "Derivation of kanade-lucas-tomasi tracking equation," May 1996, unpublished.

[4] Y. Rui, T. Huang, and S. Chang, "Image retrieval: current techniques, promising directions, and open issues," *Journal of Visual Communications and Image Represenation*, no. 10, pp. 39–62, 1999.

[5] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 1, no. 7, pp. 11–32, 1991.

[6] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147–151.

[7] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003. [Online]. Available: http://lear.inrialpes.fr/pubs/2004/MS04a

[8] D. Lowe, "Distinctive image features from scale invariant viewpoints," *International Journal of Computer Vision*, vol. 60, no. 2, 2004.

[9] Y. Ke and R.Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2004.

[10] Y. Ke, R.Sukthankar, and L. Houston, "Efficient near-duplicate detection and sub-image retrieval," in *ACM Multimedia*. ACM, Oct. 2004, pp. 1150–1157.

[11] G. Roth, "Efficient indexing for strongly similar subimage retrieval," 2006, submitted to CIVR2006-International Conference on Image and Video Retrieval.

[12] H.-Y. Shum and R. Szelski, "Panoramic image mosaics," Microsoft Research, Redmond, WA, Tech. Rep. MSR-TR-97-23, 1997.

(a) Exhaustive search (b) First retrieval (c) Second retrieval (d) Third retrieval

(e) 8 MSBs search (f) First with 8 MSBs (g) Second with 8 MSBs (h) Third with 8 MSBs

(i) 12 MSBs search (j) First with 12 MSBs (k) Second with 12 MSBs (l) Third with 12 MSBs

(m) 16 MSBs search (n) First with 16 MSBs (o) Second with 16 MSBs (p) Third with 16 MSBs

(q) 20 MSBs search (r) First with 20 MSBs (s) Second with 20 MSBs (t) Third with 20 MSBs

Fig. 5. Predator search results: exhaustive (a-d), indexed search with 8 (e-h), 12 (i-l), 16 (m-p) and 20 (q-t) signficant bits.

(a) Exhaustive search (b) First retrieval (c) Second retrieval (d) Third retrieval

(e) 8 MSBs search (f) First with 8 MSBs (g) Second with 8 MSBs (h) Third with 8 MSBs

(i) 12 MSBs search (j) First with 12 MSBs (k) Second with 12 MSBs (l) Third with 12 MSBs

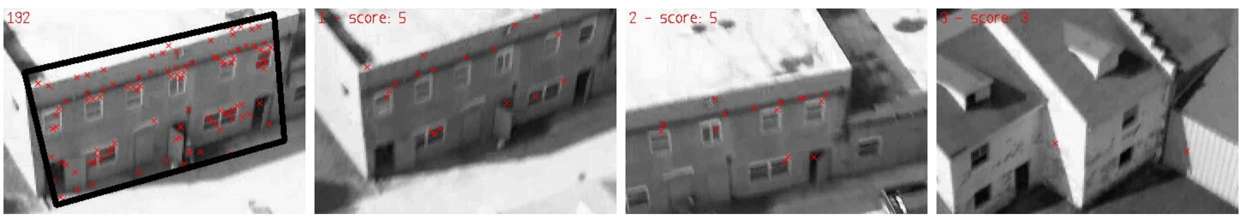
(m) 16 MSBs search (n) First with 16 MSBs (o) Second with 16 MSBs (p) Third with 16 MSBs

(q) 20 MSBs search (r) First with 20 MSBs (s) Second with 20 MSBs (t) Third with 20 MSBs

Fig. 6. Insitu search results: exhaustive (a-d), indexed search with 8 (e-h), 12 (i-l), 16 (m-p) and 20 (q-t) signficant bits.

(a) Exhaustive search (b) First retrieval (c) Second retrieval (d) Third retrieval

(e) 8 MSBs search (f) First with 8 MSBs (g) Second with 8 MSBs (h) Third with 8 MSBs

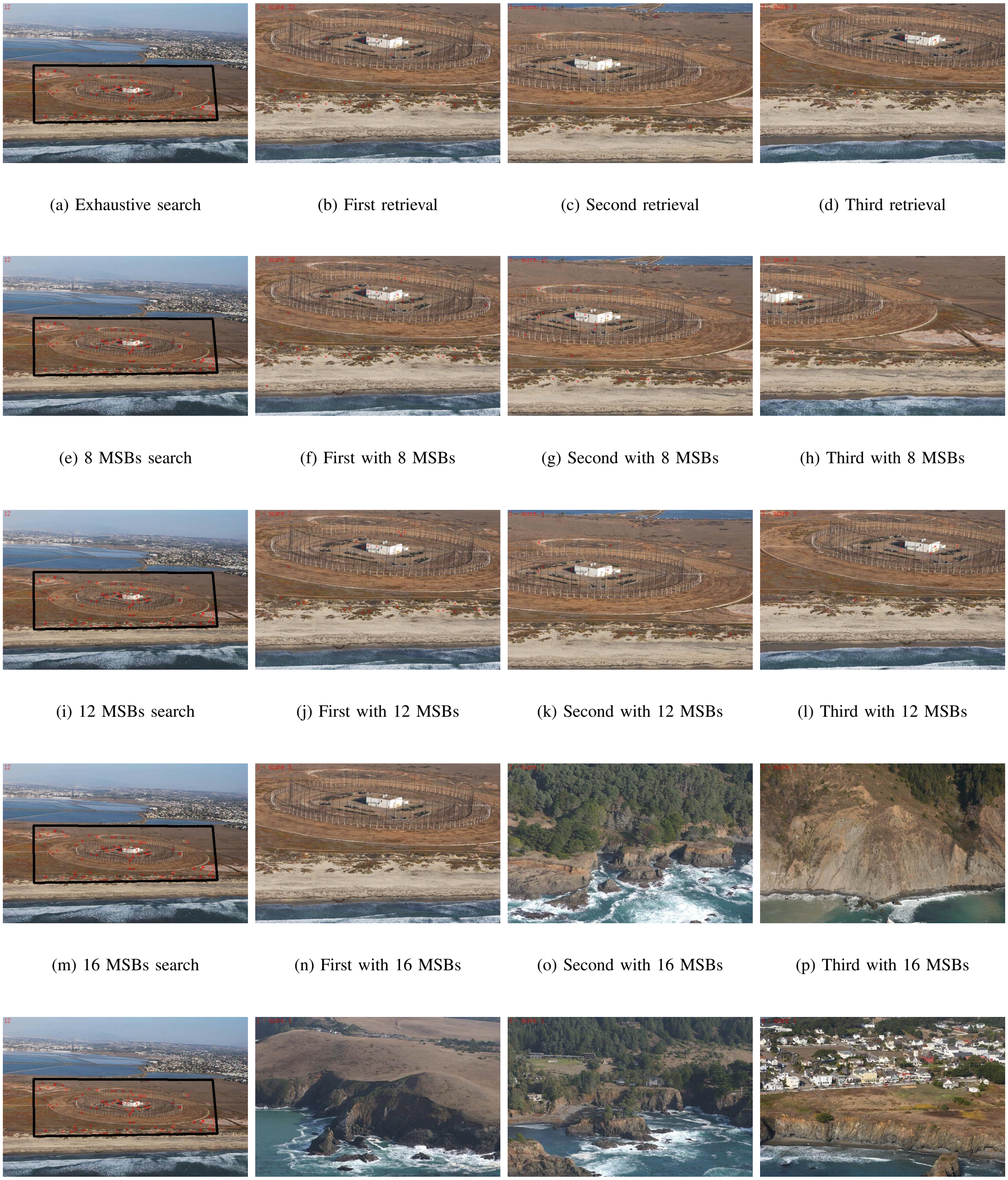


(i) 12 MSBs search (j) First with 12 MSBs (k) Second with 12 MSBs (l) Third with 12 MSBs

(m) 16 MSBs search (n) First with 16 MSBs (o) Second with 16 MSBs (p) Third with 16 MSBs

(q) 20 MSBs search (r) First with 20 MSBs (s) Second with 20 MSBs (t) Third with 20 MSBs

Fig. 7. Georgia tech search results: exhaustive (a-d), indexed search with 8 (e-h), 12 (i-l), 16 (m-p) and 20 (q-t) signficant bits.



(a) Exhaustive search

(b) First retrieval

(c) Second retrieval

(d) Third retrieval

(e) 8 MSBs search

(f) First with 8 MSBs

(g) Second with 8 MSBs

(h) Third with 8 MSBs

(i) 12 MSBs search

(j) First with 12 MSBs

(k) Second with 12 MSBs

(l) Third with 12 MSBs

(m) 16 MSBs search

(n) First with 16 MSBs

(o) Second with 16 MSBs

(p) Third with 16 MSBs

(q) 20 MSBs search

(r) First with 20 MSBs

(s) Second with 20 MSBs

(t) Third with 20 MSBs

Fig. 8. California coast search results: exhaustive (a-d), indexed search with 8 (e-h), 12 (i-l), 16 (m-p) and 20 (q-t) significant bits.