MP 97W

MITRE PRODUCT

Image & Video Retrieval for National Security Applications:

An Approach Based on Multiple Content Codebooks

6/97

Thomas P. Bronez
Elizabeth S. Hughes

Sponsor: MITRE Corporation
Dept. No.: W96 **Project No.:** 51CCG89JC4

MITRE

Corporate Headquarters
McLean, Virginia

MITRE Department Approval: _____

Michael F. Brock
Department Head

MITRE Project Approval: _____

Thomas P. Bronez
Project Leader

Abstract

Content-based retrieval of images and video is examined in the context of national security applications (defense, intelligence, and law enforcement). The unique characteristics of CBR systems for these applications are developed and contrasted with those for commercial industry (entertainment, retail, publishing, etc.). The variety of visual media, the lack of internal image structure, the presence of significant raw metadata, and the presence of domain restrictions are discussed, along with other characteristics. A general CBR approach, and its implementation, that is well-suited to the unique characteristics of national security applications is described in detail. This approach is based on the concept of multiple vector quantization codebooks for content extraction, and indexed-based query that uses multiple content types simultaneously.

Table of Contents

Section 1

Introduction

In the context of imagery and video, the goal of content-based retrieval (CBR) systems is to retrieve a set of images or video clips from a large collection on the basis of the internal content of the desired items, in addition to associated alphanumeric keywords and attributes. Such systems are in their infancy but are of great interest to organizations that must deal with large collections of non-alphanumeric data. Aigrain, *et al* [1] observe that CBR is receiving attention in industries dealing with publishing, entertainment, retail, document, audiovisual tools, visual media research, telecommunications, robotics, data compression, and digital libraries. However, they notably overlook the national security community: defense departments, intelligence agencies, and law enforcement organizations.

The national security community has a growing interest and compelling need for CBR because of its ever-increasing collections of imagery and video. The increase in data is taking place at a time when budgets dictate a decrease in human resources for analysis. Dr. Paul G. Kaminski, United States Under Secretary of Defense for Acquisition and Technology, summarized the situation recently [5]:

> We really have two CONOPS related affordability problems. The first is that our ability to collect a flood of imagery, radar-based and otherwise, will place an increasing insurmountable workload burden on image analysts if we continue to use the current exploitation approach. Our second CONOPS problem is that we have not yet developed an efficient process for managing the collection of imagery from a distributed network of sensors and processors.

Section 2

CBR Characteristics

While the national security community is certainly eager to employ commercial technology wherever reasonable, in many cases their systems require different characteristics, or at least a different emphasis, than those of private industry. It is therefore worthwhile to consider some of the salient characteristics that should be addressed by a CBR system used for national security applications.

First, the national security community deals with a *great variety of visual media* in the form of visual (electro-optic) and non-visual (infared, synthetic aperture radar, and multispectral) images and video (ground surveillance, airborne reconnaissance, as well as commercial broadcast). These come from a variety of sophisticated, widely distributed sensors, some being large observation assets and others being relatively inexpensive platforms, such as unmanned aerial vehicles (UAV). Some of the sensors may make measurements uncommon in private industry; for example, the Video Exploitation Research and Development (VERAD) committee of the United States National Imagery and Mapping Agency (NIMA) has defined stereo video exploitation as a functional need within its community. It is therefore imperative that a CBR system for national security applications be able to process a wide variety of visual media. Since different media support different types of content (e.g., stereo, motion, etc.), it follows that the CBR system should be able to exploit multiple types of content, optimally handling any particular medium according to the types of content that medium supports.

Second, the national security community generally deals with visual media that contains *little internal structure*. For example, reconnaissance video is a free flow medium, unlike human-produced entertainment or information video such as news broadcasts. Also, surveillance images are quite different from professionally composed photographs found in photobanks, newspapers, or magazines. The lack of internal structure implies that a successful CBR system will probably have to handle a set of content types that is more diverse and perhaps more domain-specific than a commercial CBR system. Color, texture, shape, and related content descriptors that are useful in the commercial sector will probably be inadequate without careful tuning and extension for the national security domain.

Third, the national security community generally deals with sensors that provide a *significant amount of raw metadata*, whereas commercial image and video sources often provide little raw metadata. This metadata typically includes sensor settings (for example, focal length, imaging band, etc.), high accuracy measurements of time and position, pointing direction, and even observation range information. As a result, detailed information can be

derived from the metadata, such as the ground coordinates and resolution of every pixel. This additional information should be a crucial element of a CBR system since it allows content features to be precisely calibrated and, therefore, more accurately interpreted. For example, for the context of UAV video, such metadata can be combined with terrain information to calibrate observed motion as "ground meters per second" rather than simply "image plane pixels per frame."

Fourth, many missions of the national security community fall within a *well-defined domain*. This can be exploited to focus the content space, because the characteristics of interest are often more specific than in the commercial case. For example, texture can be helpful for Navy littoral surveillance imagery even if only a restricted set of textures (water, surf zone, beach, scrub, etc.) are examined. Since there is no need to index the all of texture space (a difficult task since the space is difficult to quantify), texture becomes more viable as a useful content type.

Fifth, we note that there is interest not only in querying and retrieving items from large databases for archival analysis, but also in *screening high-throughput data streams* from sensors for real-time analysis. In this scenario, content analysis algorithms flag interesting images and video clips and divert them to human or higher-level machine analysis. This tiered concept places a premium on techniques that are computationally efficient and offer a very small probability of missing a scene of interest, perhaps at the expense of passing a greater number of uninteresting scenes to the analyst than would be acceptable in a commercial system.

Finally, the national security community often has *multiple data sources that should be considered simultaneously* by the CBR system. This requires correlation and cueing among different image types (for example, small-area high-resolution electro-optic images and large-area, low-resolution synthetic aperture radar images) and with non-visual information (for example, geolocation error ellipses generated by signals intelligence). The importance of geospatial correlation for image interpretation is underscored by the recent creation of NIMA from the Central Imagery Office and the Defense Mapping Agency. Though not discussed further in this paper, separate investigations are underway at The MITRE Corporation to combine conventional CBR approaches with temporal and spatial information and thereby add a spatio-temporal dimension to queries.

Section 3

General Approach

3.1 Collection Preprocessing

Given the wide variety of imaging sensors of interest to the national security community, it is essential to have a common framework for processing heterogeneous image sources. In general, there may be multiple source image formats, and a content codebook, described below, may operate on more than one of these formats. For example, a color content codebook requires a three-plane RGB source image. However, a texture content codebook can employ either a single-plane gray source image or a three-plane RGB source image; in the latter case, gray-scale conversion is done automatically by the codebook. Similarly, a motion codebook can use either a 15-frame stack of RGB or gray-scale images, or use a two-plane (horizontal and vertical velocity) motion image.

We employ a composite file format which allows multiple source images of different types to be stored together. The composite source image is generated by a sensor-aware preprocessor that selects raw data from the sensor, performs additional processing steps if needed, and formats the results. For example, in the context of video, the preprocessor would decide upon and extract a keyframe, and could also process frames around the keyframe in order to extract a motion field. The preprocessor would then store the RGB keyframe and the corresponding motion frame in the composite file. For other sensors, the preprocessing may be completely different. But the end result is the same, a composite file of related source images ready for content extraction. In this way, sensor-specific information (for example, what raw metadata is available and how to use it) is embedded entirely in the preprocessor and need not be comprehended in the subsequent content extraction or query steps. New sensors can be included by simply defining a new pre-processor.

3.2 Content Extraction

3.2.1 Overview

As shown in Figure 1, our content extraction approach begins with pre-defined vector quantization codebooks that transform one or more source images to a single *index image*, a two-dimensional array of codewords that represent the source content. Different codebooks are defined for different types of content, such as color or texture, or for quantizing a particular content type with different degrees of granularity. The remainder of the content extraction processing is the same regardless of content type, allowing different types of content to be quickly integrated into the retrieval system by simply creating a new codebook.

A region-based statistical analysis of the codebook output provides the information that we use for content indexing. Marginal content statistics are computed to determine the most frequently occurring content codewords that exist in the region being analyzed. Joint content statistics are also computed to take advantage of the relative spatial locations of the dominant codewords.

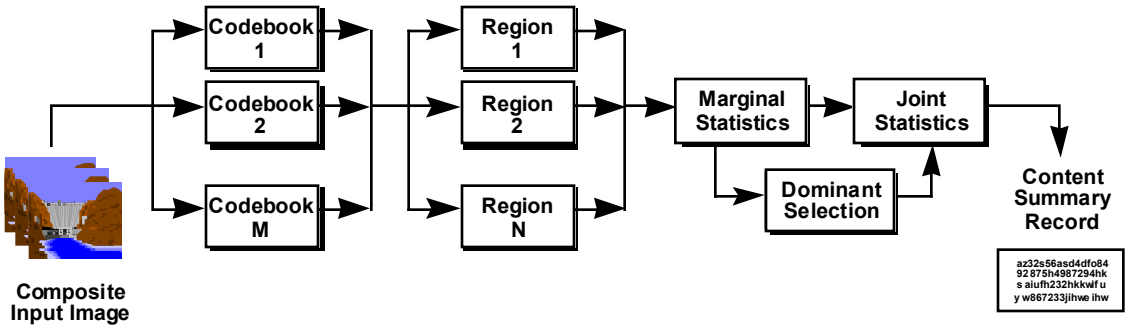

Figure 1. Content Extraction

3.2.1 Details

At the time of content extraction, several content codebooks are applied to the composite source image file. Each codebook examines the formats available within the file and selects the most appropriate format, if any, for extraction. If none of the formats are appropriate, then that codebook reports null content. We have implemented our content extraction software in ANSI C using the Vista computer vision library [7]. Vista not only contains many built-in image processing routines, but also supports a flexible image file format that is ideal for our composite source images.

Two functions comprise a content codebook, a forward mapping and an inverse mapping, between a source image and an index image. The forward mapping function takes an input image and assigns a codeword which represents a content value to each pixel in the image, or to a block of pixels. The index image generated by a codebook need not be the same size as the source image; for example, while a color codebook quantizes individual pixels, a texture codebook quantizes pixel blocks and therefore returns an index image of smaller dimensions than the source image. In fact, it is possible in principle to employ multi-resolution codebooks which yield index images having a pyramid structure. However, our implementation requires rectangular, two-dimensional index images.

The size, *S*, of codebook is the number of possible codewords that it can produce. Pixels in the index image are given values between 0 and *S-1*. A unique value is reserved to indicate "no content" at a particular pixel or pixel block. This is appropriate for certain codebooks. For example, a motion content codebook uses the "no content" value for source pixels where no motion estimate could be formed. Similarly, if an application-specific texture codebook is seeking a few particular textures, the "no content" value is employed for unknown textures. In our implementation, *S* can be no larger than 255, and the value 255 is itself reserved to indicate "no content".

The content codebooks are essentially vector quantizers. However, unlike usual vector quantization applications, we fix the vector basis a priori so that indexing may be used for efficient retrieval; there is no need to define, store, or retrieve a "content palette" for each image. The second codebook function, which maps index images into source images, corresponds to the decoder in a vector quantization system. Here it is used primarily for visualization of the content extraction results. Each index pixel is transformed into a source pixel or block of pixels that represents the basis vector of that index. This function should be consistent with the first, in the sense that transformation from an index image to a source image and back again should yield the same index image.

After codebook quantization, processing is performed over the entire image as well as for a set of fixed image subregions to characterize the image content. The subregions are formed by dividing the image into a $M \times N$ grid of equal sized blocks; where M and N are input parameters. The global image and its subregions provide a total of $MN + 1$ different regions for subsequent content processing. No automated or semi-automated image segmentation is performed at this time.

Each region is processed using histograms to determine the marginal content statistics. A discrete probability density function is formed by normalizing each histogram bin by the sum across all bins. Prior to this normalization, the counts are thresholded so that histogram bins with counts less than 0.01% of the number of pixels in the region are set to zero. Without thresholding, an image with a large number of "no content" pixels and a few pixels of some content could be misleadingly characterized as containing a large percentage of that content. Once a histogram is computed, it is sorted to determine the N dominant codewords, where N is usually related to the granularity of the codebook.

In addition to the marginal statistics, we also use joint probability measures to characterize an image's content based on the spatial relationships between the dominant codewords. The statistic that we use is similar to a co-occurrence matrix. Sometimes used for texture classification [3], co-occurrence matrices count how often pairs of gray levels of pixels, that are separated by a certain fixed distance and lie along a certain direction, occur in

an image. Our joint statistic incorporates multiple distances and orientations by counting within a local rectangular region. The joint probability *p(i,j)* is the probability that codeword *j* occurs within a spatial neighborhood centered around codeword *i*.

The index and probability of each dominant coefficient, along with the joint neighborhood statistics, are then used for region-based content indexing of an image. Our indexing technique is based upon the approaches described in [6] and [4]. By storing images based on their dominant content codes, a search for images that contain a particular content is immediately narrowed to a more manageable subset of the image database.

3.3 Content Query

Content query and retrieval is done as shown in Figure 2 using the database generated during the content extraction step. For each image in the collection and each codebook that was able to process the image, the database contains a list of regions, each region containing the marginal and joint statistics of the dominant codewords found in that region of the image with that codebook. By means of a visual interface, queries are ultimately framed in terms of the desired dominant codewords and their statistics. The visual interface varies with each codebook according to the particular type of content comprehended by that codebook.

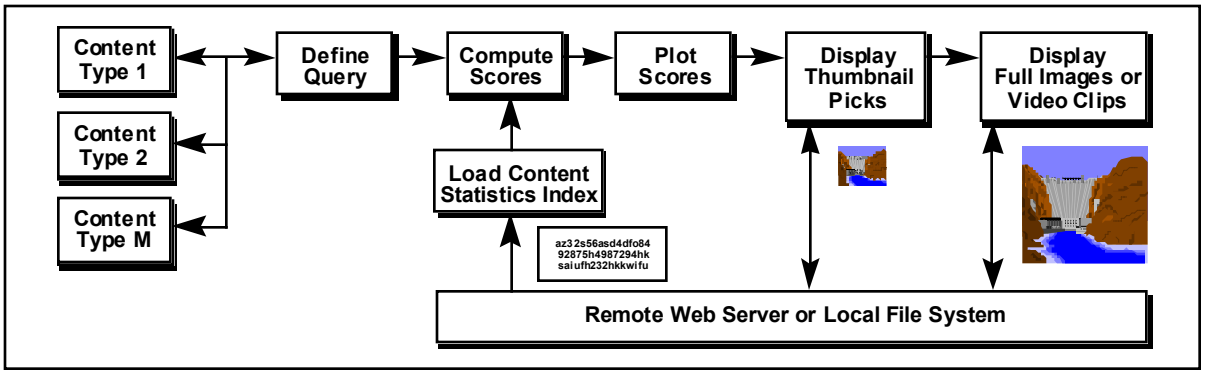

Figure 2. Content Query

A tiered procedure is used for efficiency. First, a list of candidate images is generated by finding all images that contain some of the desired dominant codewords, irrespective of their statistics. Because this step is purely an index match, it can be efficiently implemented by a lookup into a table constructed beforehand from the database. Next, the statistics record for each candidate image is retrieved from the database, and a metric score is computed to indicate the relevancy of that image to the query. A score is computed for each codebook, and the

scores are combined to generate a joint score for the image. We currently simply take the maximum of the scores as the joint score. However, formulation of the joint score is an important area for further investigation, especially for correlating different content types.

The joint scores for the candidate images are displayed in time order for video collections, and by descending score for unordered image collections. Upon user demand, image thumbnails are retrieved from the database and displayed for browsing. For videos, the thumbnails include motion visualization. If the thumbnail elicits sufficient interest, the full image or video clip can be retrieved from the database.

We have implemented our query approach as a client-side tool written entirely in Java. Figure 3 shows a screen shot of the query tool. In this figure, a query panel for motion content is displayed at the bottom, image scores are shown in the middle, and a video thumbnail is shown at the top. The video thumbnail contains both an RGB image on the left and a gray-scale image with color-coded motion on the right.

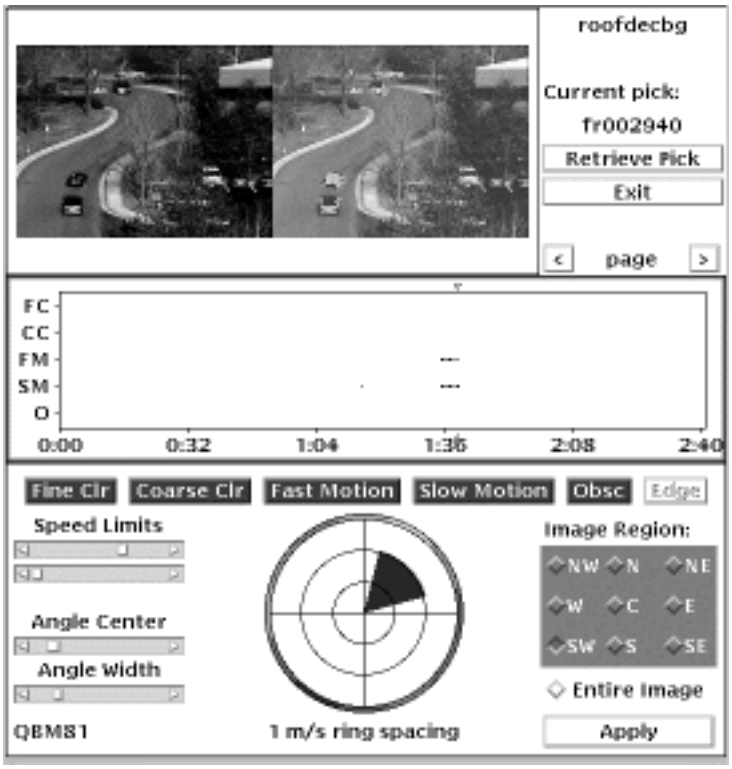

Figure 3. Content Query Tool

Section 4

Content Descriptors

4.1 Color Content

We currently have two query-by-color codebooks, representing coarse and fine color quantization schemes. Both codebooks require a RGB source image. The coarse codebook size is 27, and it simply quantizes each of the bands into 3 uniform bins. The number of dominant codewords to retain for content indexing is 3 for this codebook. For coarse color representation, we have found the RGB color space adequate. However, for the finer quantization codebook, we are using a non-uniformly sampled HSV color space of 73 colors to increase the perceptual uniformity between neighboring bins, with 7 codewords retained for content indexing. A non-uniform sampling of the HSV color space was used because of the visual redundancies between different hues at a low saturation. Our approach uses cyclic warping to quantize with an increasing number of hue bins as saturation increases.

We have used the query-by-color technique for parsing news broadcasts. For this application, we used features from the coarse 27-color codebook. Each of the four classes (start logo, anchor desk, black screen, and end logo) is represented by an averaged histogram. A minimum-distance classifier assigns a class to each frame using the same similarity measure that is used to compute scores during content query processing. A threshold on the distance is used to determine if the image should be considered "no content," rather than one of the four possible classes.

Figure 4 shows an example of the classification results from a half hour news broadcast. The occurrence of the start logo and all of the black frames were perfectly detected, and the anchor desk was detected 100% of the time with a false alarm rate of 2%. The technique failed to detect the occurrence of the end logo, most likely due to insufficient training.

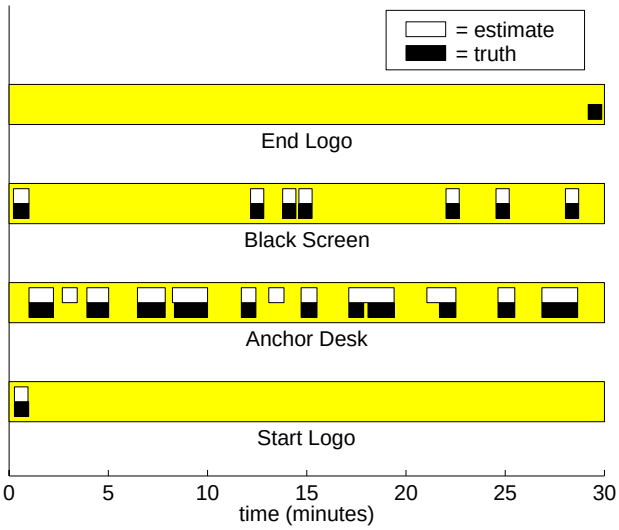

Figure 4. Color-based Analysis of News Broadcast

4.2 Motion Content

We have incorporated a motion codebook intended for general video exploitation. A video is preprocessed prior to content extraction processing to form a composite image file for every half-second of data (typically 15 frames). The composite file contains 3 images intended to support multiple content queries: a RGB image and a gray image of the center keyframe from each clip to support query-by-color and query-by-texture processing, and a 2-band velocity image containing the V_x,V_y motion estimate for every pixel to support query-by-motion.

Our motion estimation approach is based on an optical flow technique. We use a modified version of the optical flow function in the Vista library, a technique due to Lucas and Kanade [2] based on spatio-temporal first-order derivatives. The technique first smoothes an image sequence with a spatio-temporal Gaussian filter, and then estimates derivatives using four-point central differences. The Vista algorithm solves for velocity at each pixel by fitting a constant velocity model to the pixel's 5 x 5 neighborhood of derivatives, and the fit is found by weighted least squares (WLS) optimization. The reliability of the motion estimate is found by placing a lower bound on the eigenvalues of the least squares solution's characteristic equation.

We modified the Vista algorithm by replacing the WLS approach with a singular value decomposition (SVD) analysis of the system of 25 linear homogeneous equations from the

5 x 5 pixel neighborhood. This approach allowed us to use the 3 singular values directly to adaptively test the stability of the system, providing a more intuitive and robust reliability screening of the motion estimates. The reliability thresholds are based on the assumption that for a well-conditioned set of equations, the two larger singular values will be approximately equal in magnitude and the smallest singular value will be much smaller. A velocity of zero is assigned to pixels where the reliability of the motion estimate is not sufficient.

Since our applications typically involve a moving camera, it is necessary to remove background motion such as that resulting from a lateral camera pan. We have implemented a simple approach which is based on the assumption that a stationary background of non-homogeneous intensity will have approximately equal motion estimates dispersed spatially throughout the frame in a nearly uniform manner. A histogram of the motion estimates is used to find the dominant motion component, and the dispersion of this estimate is tested by a threshold on the variance of the X and Y locations where the dominant motion estimates occurred. If the dispersion criteria is met, the dominant motion estimate is subtracted from all non-zero velocities.

The motion codebook operates on the velocity image from within the composite file. We have defined two motion codebooks, one for slow moving objects where the V_x,V_y space is finely quantized (V_x and V_y ranging from -2 to 2 pixels/frame in steps of 0.5), and one for faster objects using a coarser quantization (V_x and V_y ranging from -6 to 6 pixels/frame in steps of 1). The codebook sizes are 81 and 169, respectively.

Figure 5 shows an example keyframe and the density function of the motion estimates obtained for the associated video clip. In this example, there are 4 cars and two pedestrians. The people in this frame are moving in the same direction as CAR 3, and their velocities are estimated as equal in magnitude due to the change in perspective between the near and far part of the frame. Therefore, their velocities contribute to the same peak of the density function, and their motions could only be separated spatially by subregion analysis.

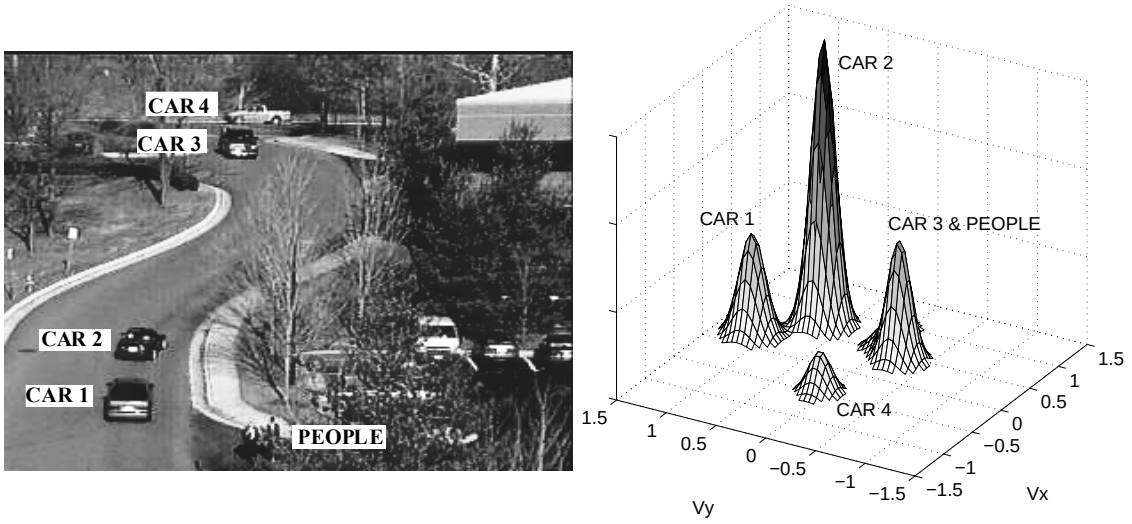

Figure 5. Motion Content Analysis

4.3 Cloud Content

We have developed a more specialized codebook for the purpose of screening unmanned aerial vehicle (UAV) surveillance videos to distinguish frames that are obscured by clouds from frames with a clear view of the ground. This codebook differs from those previously described in that it processes beyond a discrete quantization of a multidimensional feature (color or velocity) space to perform a classification of the source image. The size of this codebook is 2, since the only possible return values are 0, indicating a clear view, and 1, indicating a view that is obscured by clouds. This is a case where the codebook does not assign content at every pixel of the source image, but rather one content codeword is returned that classifies the entire image.

The cloud codebook is based on the query-by-color technique described earlier. It first performs a very coarse 5 bin quantization of a source image's gray intensities (if the source image is RGB, the codebook converts it to grayscale). Next, the marginal and joint content statistics are computed in the same way described in section 3.2.2. This is a case where processing that is usually done after the codebook quantization is also done inside of the codebook to generate the content features used by a classifier. Marginal and joint content statistics are still computed on the codebook output.

For this application, the joint statistics are very useful discriminants since large values along the main diagonal of the joint probability matrix (the self-occurrence probabilities) indicate a high degree of uniform intensity, while a low self-occurrence value indicates an area

of more varied intensity (more detail). A neighborhood size of 45 x 45 pixels is used for computing joint statistics, for source images with 480 rows and 640 columns.

For classification, the four most dominant gray coefficients are retained and a three element feature vector is formed from the self-occurrence probabilities of the second, third, and fourth most dominant coefficients. The statistic for the most dominant gray was not used because it tends to represent the overall image background, and is similar for both cloud and ground scenes. A linear classifier was designed using a Fisher linear discriminant and training on approximately 100 frames of video.

Results using a simple linear classifier appear to be very promising for detecting breaks in cloud cover using query-by-color techniques. Figure 6 shows the classification results from a short segment of UAV video. In this example, the probability of correctly classifying clear ground shots was 82%, with a false classification rate of 8%. We are in the process of investigating improvements to our classification approach, with the goal of always identifying frames that show a clear view of the ground. This is essential for a video screening application, where false alarms are more tolerable than missed detections since our goal is to minimize and not eliminate human interaction in the screening process.

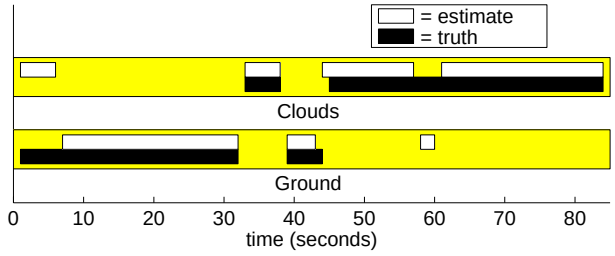

Figure 6. Cloud Content Analysis

4.4 Other Content Types

There are three other types of content that we believe are potentially useful for national security applications. These include texture, edges, and multispectral content. We are currently investigating these content types for possible incorporation into our content-based retrieval system.

Our approach for a texture codebook is to classify a specific set of classes that are relevant for a Navy reconnaissance application. The approach is similar to that of the cloud codebook, making use of the spatial joint statistics to classify textures representing ocean, surf, sand, and scrub.

We are also investigating edges as valuable content for video screening to aid in distinguishing between man-made and natural objects. This is done by indexing edges based on line segment lengths and their relative orientations.

Since our content extraction approach makes it very easy to add existing classification algorithms to our retrieval system in the form of a codebook, we plan to incorporate a multispectral classification algorithm that identifies different types of geographical features by their spectral signatures. A multispectral codebook will require a source image containing multiple planes (one for each spectral measurement band).

Section 5

Conclusions

We have examined CBR of images and video in the context of national security applications (defense, intelligence, and law enforcement). There is a growing need for CBR systems in this area, as imaging sensors become cheaper and more sophisticated, and as budgets for human analysts decline. CBR for national security should address characteristics and emphases that differ from CBR systems for commercial industry. We have described a general CBR approach, and its implementation, that we believe is well-suited to the unique characteristics of national security applications. Our approach is based on preprocessing of multiple imaging sensors, vector quantization content codebooks for extraction of multiple content types, and tiered indexed-based query for efficient, simultaneous use of the extracted content. We have described some of our codebooks in detail, and recommended other content types appropriate to national security applications for future codebook development. In the future, we plan to integrate spatio-temporal metadata into the query tool.

List of References

1. P. Aigran, H. Zhang, and D. Petkovic, "Content-Based Representation and Retrieval of Visual Media: A State-of-the-Art Review," *Multimedia Tools and Applications*, Vol. 3, pp. 179-202, 1996.
2. J.L. Barron, D.J. Fleet, S.S. Beauchemin, "Performance of Optical Flow Techniques," *International Journal of Computer Vision*, Vol. 12, No. 1, pp. 43-77, 1994.
3. C.C. Gotlieb, H.E. Kreyszig, "Texture Descriptors Based on Co-occurrence Matrices," *Computer Vision, Graphics, and Image Processing*, Vol. 51, pp. 70-86, 1990.
4. C.E. Jacobs, A. Finkelstein, D.H. Salesin, *Fast Multiresolution Image Querying*, University of Washington Technical Report 95-01-06, 1995.
5. P. G. Kaminski, keynote address to 1996 IEEE National Radar Conference, 14 May 1996.
6. A. Vellaikal, C.C. Kuo, "Content-Based Image Retrieval Using Multiresolution Histogram Representation," *SPIE Digital Storage and Archiving Systems*, Vol. 2606, pp. 312-323, Oct. 1995.
7. A.R. Pope, D.G. Lowe, "Vista: A Software Environment for Computer Vision Research," *Proceedings CVPR 1994*.

Distribution List

Internal

M. D. Adams	W621	
E. R. Anthony	W622	
M. Beebe	W521	
M. F. Brock	W621	
T. P. Bronez (100)	W621	
B. P. Flanagan	W622	
S. W. Hansen	K302	
L. Hirschman	K329	
S. D. Huffman	W556	
E. S. Hughes (5)	W622	
G. M. Jacyna	W525	
E. L. Lafferty	K312	
D. H. Lehman	A255	
W. W. Martin	W521	
M. T. Maybury	K331	
A. E. Merlino	K302	
S. L. Olson	Z140	
E. A. Palo	A390	
S. Polk		W525
J. P. Root	W432	
P. E. Silvey	K223	
M. J. Zoracki	W440	