

Automatic Video Segmentation in the Compressed Domain¹

Thomas Heath
Northrop Grumman, IT
AFRL/IFEC
32 Brooks Road
Rome, NY 13441-4114
(315) 330-2847
Thomas.Heath@rl.af.mil

Todd Howlett
Air Force Research Lab
IFEC
32 Brooks Road
Rome, NY 13441-4114
(315) 330-4592
Todd.Howlett@rl.af.mil

Maj. (Dr) John Keller
Air Force Research Lab
IFED
32 Brooks Road
Rome, NY 13441-4114
(315) 330-3944
John.Keller@rl.af.mil

Abstract—In the current military operational environment, an increasingly large quantity of motion imagery data is being collected by both manned and unmanned airborne surveillance platforms. The Department of Defense has made a commitment to move the collection of this data into the digital domain to support more effective and efficient exploitation, targeting MPEG-2 as the decompression algorithm of choice. A major impediment to the exploitation and management of this motion imagery is the sheer volume of that data.

With more information continually being provided to fewer people for exploitation, techniques need to be developed that will provide the analyst with manageable amounts of coherent video. Automatically segmenting the video into manageable duration clips is one way to achieve this. This paper describes an automatic, scene based, segmentation algorithm.

TABLE OF CONTENTS

1. INTRODUCTION
2. MPEG COMPRESSION OVERVIEW
3. SIGNIFICANT SCENE CHANGE DETECTION
4. SEGMENTATION OF VIDEO STREAM
5. CONCLUSION

1. INTRODUCTION

Surveillance missions can last for several hours. Ground based analysts are required to continually view the video being transmitted by the airborne sensor. Additionally, this video is being digitized and archived for post-mission analysis. Here, massive amounts of video need to be analyzed and catalogued for future reference.

During these missions, the scenes in the sensor field of view can vary dramatically. These scene changes can be indicators of a sensor platform change of station (i.e. breaking off one target and transiting to another). Since the

sensor is constantly transmitting imagery, the video taken during these transits frequently contains little information of interest and may be discarded, or otherwise ignored by the analyst. Detection of these transition phases can greatly benefit the intelligence analyst by providing identifiers associated with these changes.

Segmentation, or sub-dividing of the motion imagery into smaller more manageable video clips, can help to facilitate the video exploitation process. Several commercially available segmentation tools are available, but require manual intervention. It has been shown [1] that by using the information inherent in the MPEG-2 data structure, scene changes can be automatically detected. Once detected, these scene changes can be used as automatic segmentation markers. Segmentation based on scene changes can provide the analyst with homogeneous video segments allowing for more efficient management and exploitation of the surveillance video.

Since MPEG compression is obtained by a combination of spatial and temporal prediction, measuring the amount of, or changes in this prediction, can be used as an indicator that scene changes are occurring in the video. Once scene changes are detected, video segments can be generated by extracting portions of the original video bounded by these detected scene changes.

In addition to taking advantage of the inherent information embedded in the MPEG structure, segmenting video in the compressed domain is more efficient than segmenting uncompressed information. Additionally, there is no additional corruption of the image information resulting from a repeated encoding process.

Slightly modified from the technique presented by Kobla et al [1], the approach to automatically segment video presented in this paper relies on less information in the MPEG structure, yet yields comparable results. This technique will be applied to UAV surveillance video.

1. ¹ U.S. Government work not protected by U.S. copyright.

This paper presents an overview of MPEG compression, utilization of the pertinent MPEG data fields for scene change detection and a description of the segmentation process. Conclusion and results are then presented.

2. MPEG COMPRESSION OVERVIEW

To achieve compression, MPEG data structures, [2], [3], [4], [5], are generated by taking advantage of the spatial redundancy that occurs over time in a video stream. Each MPEG structure is composed of a sequence of Groups of Pictures (GOPs)(see figure 1). Typically, each GOP consists of a sequence of 10 to 15 frames. A frame can be one of 3 different types, Intra-Coded (I), Predictive (P), and Bi-Directionally Predicted (B). Each GOP typically consists of P and B frames and is bounded on each end by I frames. The sharing of spatial information occurs only within a GOP. This serves to limit the propagation of errors to within that GOP.

I frames are self-referential, using only data from within the current frame for compression and do not rely on other frames for information required to regenerate them into a picture. P-frames, are created using both self-referential information as well as information from a previously occurring reference frame. (Reference frames can be either I or P frames from which spatial information is shared with other frame(s)). B frames are similar to P frames, but they can use information from previous and subsequent frames for them to be regenerated into an image. Figure 1 shows a typical GOP and it's information sharing patterns.

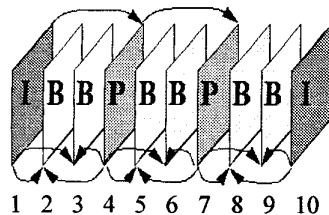


Figure 1: Relationship between reference and predicted frames

Frames are not compressed in the order in which they are received. As shown in figure 1, frame 4 is a reference frame providing information to frames 2 and 3. Therefore, this P frame must be created before frames 2 or 3. Figure 2 shows a typical compressed frames sequence.

All predicted frames, (P and B) are generated by first subdividing them into 8x8 macro-blocks. Each macro-block is then spatially correlated with the associated reference frame(s) by calculating a cumulative difference over the macro-block. If this difference is sufficiently small, then this spatial offset is used to provide a measure of the

apparent motion of the macro-block from one frame to the next. This apparent motion is saved as motion vectors. Additionally, the spatial correlation information is saved. If the spatial correlation is not sufficiently small, then prediction for that macro-block is suspended. (Remember, P and B frames can contain information that is self-referential.)

Upon completion of the spatial correlation, a Discrete Cosine Transform (DCT) is applied to the difference information in each macro-block within each frame, sorting the frequency components of each macro-block. Quantization and thresholding operations are then performed on the frequency information yielding a reduced information set. Frame data is then further compressed by performing a Run Length Encoding (RLE) compression scheme. Each compressed frame and it's associated motion vectors are then saved in the MPEG stream.

As mentioned earlier, I frames are not predicted. Creation of such frames is done by simply performing a series of DCTs across each I frame. As with P and B frames, quantization, thresholding and RLE operations are performed on macro-blocks in each of the frames to achieve maximum compression. Figures 2a-2c show a simplified general flow in the creation of I, P, and B frames.

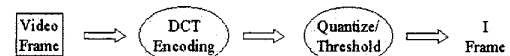


Figure 2a: I-Frame Generation

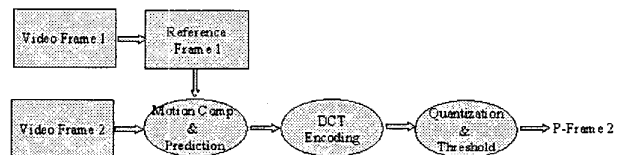


Figure 2b: P-Frame Generation

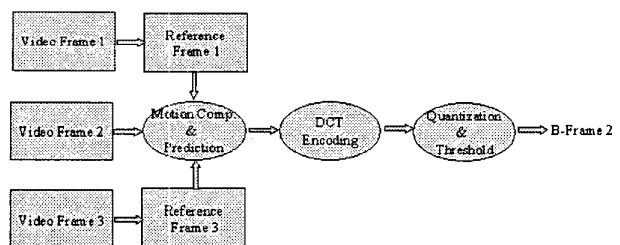


Figure2c: B-Frame Generation 1

3. DETECTING SIGNIFICANT SCENE CHANGES

As indicated above, predicted macro-blocks are the vehicle that the MPEG compression scheme uses to share temporal information while motion vectors are used to share spatial information. Recalling that macro-blocks can be intra-coded, forward or bi-directionally predicted, and are labeled as such within the MPEG stream, this information can be used to detect scene changes in the video stream.

The apparent motion present in a video stream can be created by panning a video camera across an area of interest. Movement within a video stream can also be created when the camera is static with objects moving within a fixed field of view. In either case, significant overlap occurs in frame to frame image content. This spatial overlap translates into a highly predictive MPEG stream. A highly predictive MPEG stream therefore implies a large number of Forward and Bi-directionally predicted macro-blocks and a corresponding low number of Intra-coded macro-blocks.

Sudden and abrupt content changes in the field of view serves to minimize the predictive nature of MPEG. Consequently, there is an increase in the number of intra-coded macro-blocks being generated for each frame, and a corresponding decrease in the number of predictive macro-blocks.

To test this premise, actual aerial surveillance video was used. The sensor payload was switchable between an Electro-Optical and an Infrared sensor. This video was taken on a partly sunny day, with intermittent cloud cover.

Figure 3 is a graph of the number of Intra-coded macro-blocks occurring over a sequence of 2000 predicted (B and P) frames from actual surveillance video. Since I frames are composed entirely of intra-coded macro blocks they are not used for scene change detection.

In figure 3, the horizontal axis represents frame numbers and the vertical axis shows the number of intra-coded macro-blocks. One can see that at about frame 600, identified by 'A', the number of intra-coded macro-blocks begins to increase. This is due to a switch of sensors from IR to EO. Note the significant jump in the number of intra-coded macro-blocks, and subsequently, the way in which their numbers begin to decrease. However, before reaching steady state, at about frame number 950, identified by 'B', cloud cover begins to interfere, again causing a decrease in the MPEG prediction. This cloud cover is sustained until about frame 1700 when the prediction begins to become more prevalent and the numbers of intra-coded macro-blocks decreases. This behavior continues, until another occurrence of cloud cover, indicated by 'C', at which point

the predictability of the video decreases and the numbers of intra-coded macro-blocks again increases.

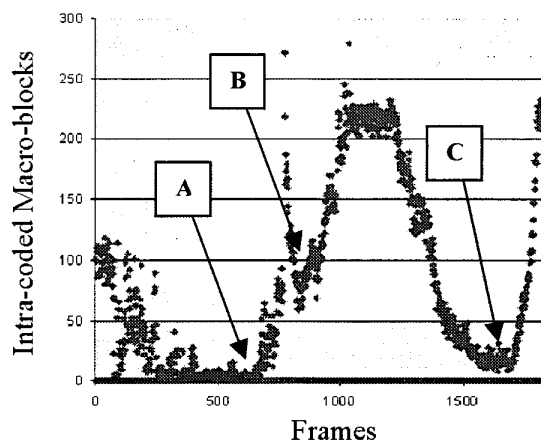


Figure 3: Intra-coded macro-block pattern for dynamic video

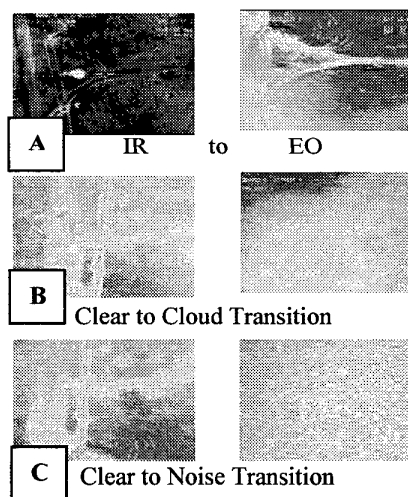


Figure 4: Visual Transitions causing Intra-coded Macro-block variances

Figure 4 shows the actual frames from the MPEG stream near the macro-block variances shown in Figure 3. Conversely, figure 5 shows an intra-coded macro-block pattern produced from a video stream with smooth frame to frame scene transitions. Clearly, the macro-block pattern is relatively stable, showing no significant changes in behavior.

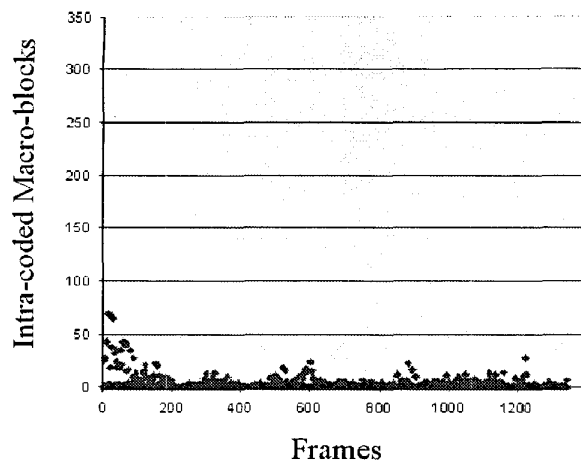


Figure 5: Intra-coded macro-block pattern for stable video

It is this macro-block relationship (an increase in intra-coded macro-blocks implies a reduction in frame to frame predictability) that is the basis of the segmentation process. A simple technique to detect a significant change in the number of intra-coded macro blocks can be used as an indicator of significant scene change. Additionally, a sustained and relatively large number of intra-coded macro-blocks can be used to indicate an extended period of significant frame to frame change or simply noise in the video stream. Regardless, the pattern that occurs in the intra-coded macro-block count is indicative of activity within the video stream.

4. SEGMENTATION OF THE VIDEO STREAM

The physical segmentation of an MPEG video stream is achieved by locating occurrences of changes in the numbers of intra-coded macro-blocks. To do this, a running 10 point average of the number of intra-coded macro-blocks in each frame is maintained. This allows for the detection of significant changes in the numbers of the occurrences of these macro-blocks.

As the MPEG stream is being parsed, the header information is read and forwarded to an output file. This header information is retained for use in subsequent video segments. As each new block of information is read in and written, it is also parsed. During this parsing operation, macro-blocks are identified and counted. The latest information regarding the intra-coded macro-blocks is added to the sliding window average. See Figure 6 below.

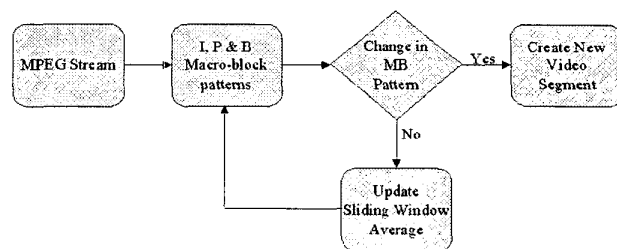


Figure 6: High Level Flow Diagram of Automatic Segmentation

If it is determined that a significant change in the macro-block count has occurred, the output video can be closed and all filters and counters are reset. However, prior to closing the file, it must be ensured that the last group of pictures read is completely written to the output file. This will ensure that all frames in that group can be reconstructed.

5 CONCLUSIONS

This paper describes how information inherent in an MPEG-2 video stream can be utilized to automatically detect significant scene changes in a video stream of interest. Detection of these scene changes and segmentation of the video stream using these detections can be used to facilitate exploitation and pictorial interpretation.

It is anticipated that automatically segmented video will be homogeneous in nature, containing similar image content over time. The homogeneous nature of the segmented video can then be automatically mosaiced. This combination of automatic segmentation and mosaicing will provide an analyst with a series of still images, rather than a single lengthy video clip allowing for more efficient exploitation of the video information.

REFERENCES

- [1] Vikrant Kobla, David Doermann and Azriel Rosenfeld, *Compressed Domain Video Segmentation*, CAR-TR-389, CS-TR-3688, September, 1996
- [2] Murray & VanRyper, *Encyclopedia of Graphics File Formats*, O'Reilly & Associates, Sebastopol, CA. 1994
- [3] ISO/IEC 13818-2. Information Technology – Generic coding of moving pictures and associated audio

information: Video

[4] Barry G. Haskell, Atul Puri, Arun N. Netravali, Kluwer, Digital Video: An Introduction to MPEG-2, Boston, Academic Publishers, 1996

[5] Shanawaz Basith, Digital Video: An Introduction: MPEG: Standards, Technology and Applications, 1996

Mr. Thomas Heath is a Principal Computer Analyst with Northrop-Grumman Information Technologies, working on site at the Air Force Research Laboratory, Rome, NY. He began his career working for General Electric, Ocean and Radar Systems developing real-time signal processing applications for use in US Navy surface ship sonars. During the past four years, he has been performing R&D tasks related to Image Exploitation for the US Air Force at the Rome Research Site. His research interests include real time acoustic and imagery applications

Mr. Todd Howlett is a computer engineer working for the Air Force Research Laboratory's Multi-Sensor Exploitation Branch. He has worked to develop imagery and information systems for the Air Force. He is currently pursuing research related to motion imagery. He has a BS in Electrical Engineering from the State University of New York at Buffalo and a MS in Electrical and Computer Engineering from Syracuse University.

Major John Keller is the Senior Global Information Architect at the Air Force Research Laboratory, Rome Research Site, Rome, New York. He received his BSEE from the University of Florida, and his MSEE and PhD, both specializing in pattern recognition and signal processing, from the Air Force Institute of Technology in Dayton, Ohio. His research interests are currently focused on extraction/exploitation of spatio-temporal information within motion imagery.