

AFIT/GCS/ENG/99M-17

ANALYSIS OF THE APPLICABILITY
OF VIDEO SEGMENTATION TO UNMANNED
AERIAL VEHICLE SURVEILLANCE VIDEO

THESIS
Bradley L. Pyburn
1st Lieutenant, USAF

AFIT/GCS/ENG/99M-17

Approved for public release; distribution unlimited

19990409 098

| REPORT DOCUMENTATION PAGE | | | Form Approved OMB No. 0704-0188 | |
|---|--|---|---|--|
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503. | | | | |
| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE March 1999 | 3. REPORT TYPE AND DATES COVERED Master's Thesis | | |
| 4. TITLE AND SUBTITLE Analysis of the Applicability of Video Segmentation to Unmanned Aerial Vehicle Surveillance Video | | | 5. FUNDING NUMBERS | |
| 6. AUTHOR(S) Bradley L. Pyburn, 1st Lt, USAF | | | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology WPAFB, OH 45433-7765 | | | 8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GCS/ENG/99M-17 | |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Maj Steven M. Matechik AFRL/IFEC 26 Electronic Parkway Rome, NY 13441-4514 DSN 587-4426 COMM (315) 330-4426 | | | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER | |
| 11. SUPPLEMENTARY NOTES Maj Michael L. Talbert (advisor) michael.talbert@afit.af.mil DSN 785-6565 ext 4280 COMM (937) 255-6565 | | | | |
| 12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution unlimited | | | 12b. DISTRIBUTION CODE | |
| 13. ABSTRACT (Maximum 200 words) The focus of this research is to determine if applying edge detection segmentation (proposed by Ramin Zabih, Justin Miller, and Kevin Mai) to Unmanned Aerial Vehicle (UAV) video footage can provide meaningful segments for database storage and retrieval. The edge detection segmentation algorithm is applied to fifty-four UAV video sequences containing visual effects such as abrupt camera changes, camera zooms, motion (rapid and gradual), and cloud cover while varying the frame rate from 5 fps to 30 fps. An analysis of the results is performed to compare actual versus expected outcomes, similar sequences, and scenes with motion, along with explaining false positives/anomalies. Although the frame rate variation and analysis of the scenes with cloud cover are inconclusive, applying the edge detection segmentation algorithm to abrupt changes, rapid motion, and camera zooms produced favorable results, as these were all detected as scene changes. Several near-term and long-term benefits can be drawn from these results, and are provided at the conclusion of the paper, along with recommendations for future research. | | | | |
| 14. SUBJECT TERMS Video Database System, Video Information Storage and Retrieval, Video Segmentation, Video Scene Analysis, UAV Video Data | | | 15. NUMBER OF PAGES 109 | |
| | | | 16. PRICE CODE | |
| 17. SECURITY CLASSIFICATION OF REPORT Unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified | 20. LIMITATION OF ABSTRACT UL | |

The views expressed in this thesis are those of the author and do not necessarily reflect the official policy or position of the Department of Defense or the United States Government.

AFIT/GCS/ENG/99M-17

ANALYSIS OF THE APPLICABILITY OF VIDEO SEGMENTATION
TO UNMANNED AERIAL VEHICLE SURVEILLANCE VIDEO

THESIS

Presented to the Faculty of the Graduate School of Engineering
Of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Computer Systems

Bradley L. Pyburn, B.S.
1st Lieutenant, USAF

March 1999

Approved for public release, distribution unlimited

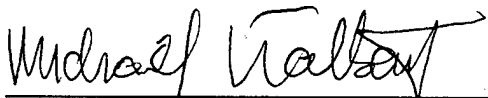
AFIT/GCS/ENG/99M-17

ANALYSIS OF THE APPLICABILITY OF VIDEO SEGMENTATION
TO UNMANNED AERIAL VEHICLE SURVEILLANCE VIDEO

THESIS

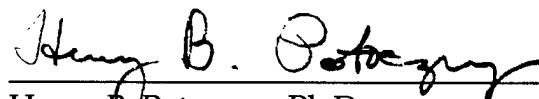
Bradley L. Pyburn, B.S.
1st Lieutenant, USAF

Approved:



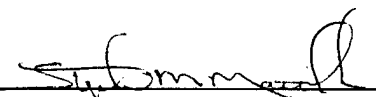
Michael L. Talbert, Major, USAF
Chairman

26 Feb 1999
Date



Henry B. Potoczny, Ph.D.
Member

25 Feb, 1999
Date



Stephen M. Matechik, Major, USAF
Member

25 Feb 99
Date

ACKNOWLEDGMENTS

In the course of this research endeavor, many people contributed their time and expertise to ensure its successful completion. I would like to take this opportunity to thank all those who have provided support in one way or another.

First, I would like to thank my thesis advisor, Maj Michael Talbert, for guiding my research. Your enthusiasm about multimedia information retrieval not only allowed me to select and properly scope an enjoyable topic, but also assisted me in successfully documenting the results.

For allowing flexibility in exploring the problem domain of video information storage and retrieval, I would like to thank my sponsor, Maj Stephen Matechik from Rome Laboratory. Although our meetings were few, all the information and suggestions you provided were invaluable in my research.

I would also like to take the opportunity to thank each of my instructors in the Computer Science Department at AFIT. Special thanks go to Dr Henry Potoczny, for making difficult subjects enjoyable and refreshing.

For their support, suggestions, and sometimes their humor, I would like to thank my classmates and peers in GCS-99M. More than once you made tough times easier to bear.

For their love and encouragement, I would especially like to thank my family. To Jackie, none of this would have been possible without you. Although you didn't assist me directly, simply listening to me after a tough day and being patient with my constant frustrations was priceless. To Taylor, Elissa, and Kirsten, thanks for your unconditional love and understanding why sometimes Daddy had to work and study instead of playing with you.

Finally, I would like to give thanks to God, for blessing me with the skills and abilities to be successful in the AFIT program and as an Air Force Officer.

Bradley L. Pyburn

TABLE OF CONTENTS

| | |
|--|-----|
| ACKNOWLEDGMENTS | II |
| TABLE OF CONTENTS..... | III |
| LIST OF FIGURES..... | VII |
| LIST OF TABLES | IX |
| ABSTRACT | X |
| 1 INTRODUCTION..... | 1 |
| 1.1 <i>Problem</i> | 3 |
| 1.2 <i>Scope</i> | 5 |
| 1.3 <i>Approach</i> | 5 |
| 1.4 <i>Assumptions</i> | 6 |
| 1.5 <i>Thesis Organization</i> | 7 |
| 2 BACKGROUND..... | 8 |
| 2.1 <i>Overview of Digital Storage and Retrieval</i> | 9 |
| 2.1.1 <i>Characteristics of Video Data</i> | 9 |
| 2.1.2 <i>Digital Representation of Video Data</i> | 11 |
| 2.1.3 <i>Video Database Management System</i> | 22 |
| 2.2 <i>Video Segmentation</i> | 31 |
| 2.2.1 <i>Pairwise Comparison</i> | 33 |

| | | |
|-------|--|----|
| 2.2.2 | Likelihood Comparison | 33 |
| 2.2.3 | Global Histogram..... | 34 |
| 2.2.4 | Local Histogram..... | 34 |
| 2.2.5 | Weighted Color Histogram | 35 |
| 2.2.6 | Edge Detection Segmentation Method | 35 |
| 2.3 | <i>Predator Unmanned Aerial Vehicle</i> | 36 |
| 2.3.1 | Predator UAV System Background..... | 37 |
| 2.3.2 | Predator UAV System Configuration | 38 |
| 2.3.3 | Predator UAV Product Dissemination | 39 |
| 2.3.4 | Predator UAV Video Data Retrieval Limitations..... | 40 |
| 2.4 | <i>Summary</i> | 40 |
| 3 | METHODOLOGY..... | 43 |
| 3.1 | <i>Algorithm Selection</i> | 44 |
| 3.1.1 | Algorithm Comparison..... | 44 |
| 3.1.2 | Data Characteristics | 46 |
| 3.2 | <i>UAV Test Data</i> | 46 |
| 3.2.1 | Test Data Acquisition | 47 |
| 3.2.2 | Video Sequence Selection | 47 |
| 3.3 | <i>Experimental Setup</i> | 49 |
| 3.3.1 | Hardware and Software Configuration..... | 50 |
| 3.3.2 | Experimental Process | 50 |
| 3.3.3 | Analysis of Results..... | 51 |

| | | |
|-------|---|----|
| 3.4 | <i>Summary</i> | 52 |
| 4 | RESULTS | 54 |
| 4.1 | <i>Frame Rate Variation</i> | 54 |
| 4.2 | <i>Abrupt Changes</i> | 55 |
| 4.3 | <i>Rapid Motion</i> | 61 |
| 4.4 | <i>Zooms</i> | 62 |
| 4.5 | <i>Cloud Cover</i> | 68 |
| 4.6 | <i>Motion</i> | 71 |
| 4.6.1 | <i>Slow/Gradual Motion at High Altitudes</i> | 72 |
| 4.6.2 | <i>Fast Motion at Low Altitudes</i> | 72 |
| 4.7 | <i>False Positives and Anomalies</i> | 74 |
| 4.7.1 | <i>Words/Telemetry Data</i> | 76 |
| 4.7.2 | <i>Corrupt Data</i> | 76 |
| 4.7.3 | <i>Motion</i> | 77 |
| 4.8 | <i>Summary</i> | 79 |
| 5 | CONCLUSIONS AND RECOMMENDATIONS..... | 80 |
| 5.1 | <i>Near-Term Benefits</i> | 80 |
| 5.1.1 | <i>Key Frames for Mission Content</i> | 81 |
| 5.1.2 | <i>Video Partitioning for Imagery Mosaics</i> | 82 |
| 5.1.3 | <i>Segmentation Based on Telemetry Data</i> | 82 |
| 5.2 | <i>Long-Term Benefits</i> | 83 |
| 5.2.1 | <i>Key Frames for Indexing and Storage</i> | 83 |

| | |
|--|-----------|
| 5.2.2 Mission Profiles | 84 |
| 5.3 <i>Future Research Directions</i> | 86 |
| 5.4 <i>Summary</i> | 88 |
| APPENDIX A - DATA AND SOFTWARE AVAILABILITY | 90 |
| BIBLIOGRAPHY | 91 |
| VITA | 95 |

LIST OF FIGURES

| | |
|---|----|
| Figure 2-1. Video Digitization Process | 12 |
| Figure 2-2. Digital Video Representation..... | 13 |
| Figure 2-3. MPEG Structure | 16 |
| Figure 2-4. Intra-Block Coding Process | 17 |
| Figure 2-5. Generic VDBMS | 23 |
| Figure 2-6. Hierarchical Abstraction of Video Data | 26 |
| Figure 2-7. Predator in Flight..... | 38 |
| Figure 4-1. Abrupt Scene Change at 30 FPS..... | 56 |
| Figure 4-2. Abrupt Scene Change at 5 FPS..... | 56 |
| Figure 4-3. Abrupt Scene Change at 30 FPS..... | 57 |
| Figure 4-4. Abrupt Scene Change at 5 FPS..... | 57 |
| Figure 4-5. Typical Abrupt Change..... | 59 |
| Figure 4-6. Abrupt Change with Low Threshold Value | 60 |
| Figure 4-7. Rapid Motion Scene Change/Choppy Edge Change Fraction | 63 |
| Figure 4-8. Rapid Motion/Uniformly High Edge Change Fraction | 64 |
| Figure 4-9. Zoom In Scene Change | 66 |
| Figure 4-10. Zoom Out Scene Change | 67 |
| Figure 4-11. Light Cloud Cover | 69 |
| Figure 4-12. Heavy Cloud Cover..... | 70 |
| Figure 4-13. Gradual Motion..... | 73 |

| | |
|---|----|
| Figure 4-14. Fast Motion | 75 |
| Figure 4-15. Telemetry Data Appears | 77 |
| Figure 4-16. Corrupt Data..... | 78 |
| Figure 5-1. Example Mission Profile Plot Based On Edge Change Fractions | 85 |

LIST OF TABLES

| | |
|---|----|
| Table 2-1. Comparison of Video with Textual and Image Data | 10 |
| Table 2-2. Summary of Segmentation Algorithms | 41 |
| Table 3-1. Test Cases | 49 |

ABSTRACT

The focus of this research is to determine if applying edge detection segmentation (proposed by Ramin Zabih, Justin Miller, and Kevin Mai) to Unmanned Aerial Vehicle (UAV) video footage can provide meaningful segments for database storage and retrieval. The edge detection segmentation algorithm is applied to fifty-four UAV video sequences containing visual effects such as abrupt camera changes, camera zooms, motion (rapid and gradual), and cloud cover while varying the frame rate from 5 fps to 30 fps. An analysis of the results is performed to compare actual versus expected outcomes, similar sequences, and scenes with motion, along with explaining false positives/anomalies. Although the frame rate variation and analysis of the scenes with cloud cover are inconclusive, applying the edge detection segmentation algorithm to abrupt changes, rapid motion, and camera zooms produced favorable results, as these were all detected as scene changes. Several near-term and long-term benefits can be drawn from these results, and are provided at the conclusion of the paper, along with recommendations for future research.

ANALYSIS OF THE APPLICABILITY OF VIDEO SEGMENTATION TO UNMANNED AERIAL VEHICLE SURVEILLANCE VIDEO

1 INTRODUCTION

In today's age, more information is being generated than at any other time in human history. Terms such as *information age* and *information revolution* have been coined to capture the excitement concerning the explosion of information available over the last decade. Information has become somewhat of a commodity and an asset, critical to the success of all types of organizations, from large corporate entities to the US armed forces. As the amount of information generated grows exponentially, it has become exceedingly difficult to retrieve the right piece of data at the right time.

To further complicate matters, information is not only plain text, but also media rich sources containing audio and visual stimuli. One example of a media rich source, video, is one of the most prevalent forms of information today. For example, video is used as a communications medium to broadcast up-to-the-minute news telecasts to consumers in their homes around the world. Video can also be used for other purposes, such as entertainment, corporate training, advertising, and university lectures over the Internet. An application of video

that is important to the military is reconnaissance and surveillance. Reconnaissance platforms such as the Unmanned Aerial Vehicle (UAV), in conjunction with high-bandwidth networks, such as the Global Broadcast System (GBS), provide real-time video feeds for war planners and war fighters.

In many of the aforementioned situations, it is suitable to extract the most important still images from a video, and use these to represent the sequence of events. However, in many applications, especially military uses, the detailed information that video provides (such as speed, direction, duration, etc.) is extremely important. This level of detail is exceedingly difficult to capture with any other data source [PATEL97]. Therefore, it is imperative the entire video is stored, instead of representative clips or sequences.

As a consequence of the large amount of video generated today, an efficient and effective means of managing and retrieving this data must be developed. However, traditional database management systems (DBMS) lack the capabilities to support an object type with spatial and temporal properties. Some *ad hoc* systems have been developed to handle video data, but they fall short of providing the robust utilities of a full-blown database system [ELMAG97]. Therefore, much research is being conducted towards developing a video database system.

As with any database system, before the data can be of any practical use, it must be indexed and annotated for insertion into the database and for

subsequent semantic-based retrieval and analysis [DAILI95]. However, indexing digital video is an extremely challenging and complex problem. This is due to the fact that in most cases, the item a user may be interested in is not the video in its entirety, but a particular segment contained within the video. As a result, indexes must be provided to the internal segments of a video, not just to the video itself. Indexes could be provided to the most atomic unit of video data, known as a *frame*. However, indexing by individual frames would be extremely inefficient. A more advantageous method would be to identify meaningful segments to serve as retrievable units [LIENH97].

Consequently, video is usually broken down into an atomic level of granularity known as a *shot*. A shot can be described as a collection of continuous video frames consisting of the same action in space and time [SETHI95]. The act of separating a video into its basic shots is known as *video segmentation*. Once a video has been segmented, its distinct shots can be indexed for future searches, or combined with similar shots to form a meaningful episode.

1.1 Problem

As stated earlier, the Unmanned Aerial Vehicle (UAV) provides real-time video feeds to tactical war planners. This UAV video data is also of considerable value to intelligence imagery analysts. Currently, imagery analysts must process hundreds of hours of video data, much of which contains limited useful

information. To further complicate matters, budget constraints and personnel cutbacks, in conjunction with a high operations tempo, place an enormous burden on an already overtasked workforce. Accordingly, a robust method is needed to allow the analysts to process the video data and extract the pertinent information, along with an effective facility for storage and semantic-based retrieval of that information.

Current operational intelligence data handling systems (IDHS) are not equipped to handle digital video. As a result, imagery analysts have no facility to retrieve UAV data on a particular target or area of interest. The current method consists of a manual, time-expensive search through an 8mm-video tape library containing several thousand tapes [WIEDE97]. Therefore, much work needs to be done towards applying state-of-the-art video database technology and applications to IDHS environments responsible for the exploitation of UAV data. As discussed earlier, video segmentation is a first step towards partitioning and indexing digital video for insertion in a database system. There are many video segmentation algorithms in the literature [BOUTH97] [LEEIP95] [MENGJ95] [SETHI95] [VASCO97] [WANGA94] [XIOLE95] [XIOIP95] [YEOLI95] [ZABIH97]. Most of these were developed to detect man-made scene cuts such as those that occur in edited motion pictures. However, little work has been done applying these algorithms to surveillance video such as UAV footage. Accordingly, the focus of this research is to determine if the application of state-

of-the-art video segmentation to UAV video footage can provide meaningful segments for database storage and retrieval.

1.2 Scope

Although much work is required to implement a video database system to store and retrieve UAV footage, this work will be limited to an analysis of video segmentation. Specifically, this research applies a pre-selected video segmentation algorithm to UAV footage, to determine if video segmentation is a viable means of partitioning UAV video data. The algorithm used in this research is the edge detection method proposed by [ZABIH97]. A description of this method is provided in section 2.2.6, and the rationale for its choice in this research is explained in section 3.1.

1.3 Approach

This research will be conducted in several steps. The first step consists of a literature review of video databases and video segmentation. Based on this research, an applicable video segmentation algorithm is selected from the literature for experimental purposes. The selection criterion for the segmentation algorithm takes into account not only the performance of the algorithm, but also the type of data it will be used with (continuous surveillance video footage). After the algorithm selection, UAV video footage is selected for experimental purposes. To include a wide variety of the typical visual effects a UAV may encounter, scenes with abrupt changes, camera zooms, slow motion, rapid

motion, and cloud cover are chosen. Finally, the selected segmentation algorithm is executed on each of the scenes and the results are collected for subsequent analysis. The analysis consists of comparing the results of expected versus actual outcomes, similar sequences, and scenes with motion, along with explaining anomalies and false positives.

1.4 Assumptions

A few assumptions are necessary in order to perform the research proposed in this document, and to validate the results and conclusions of the experiment. First, it is assumed the UAV data is available in digitally encoded MPEG (Motion Picture Expert Group, see section 2.1.2.2 for a full discussion) format. Second, the sampling of UAV video footage used for experimental purposes in this research is assumed to be a good representation of the typical visual effects the UAV visible light cameras encounter. Visual effects such as zooms, rapid camera movements, and abrupt camera feed changes form the cornerstone on which scene breaks are detected in UAV footage. Finally, it is assumed the time and processing requirements (CPU power, RAM, etc.) required to perform segmentation are available to UAV imagery exploitation analysts in their current or near-future IDHS environments. Video segmentation is a time and CPU-wise expensive process, and these resources must be available for video segmentation to be of any near-future benefit to UAV imagery analysts.

1.5 Thesis Organization

This document is divided into 5 chapters. Chapter 1 introduces the concept of video segmentation, and renders an explanation of its possible application to UAV video data. In Chapter 2, a background on digital video storage and retrieval is presented, including a discussion of digital representation of video and a survey of video database system issues and concepts. Also in Chapter 2, an overview of video segmentation is provided, along with a description of the Predator UAV system. Chapter 3 describes the methodology undertaken in this research, specifically the algorithm selection, UAV data acquisition, and subsequent algorithm experimentation and analysis. Finally, Chapters 4 and 5 present the analysis of the experimental results, and conclusions and recommendations for future research, respectively.

2 BACKGROUND

As stated in Chapter 1, video is common in many aspects of life, including military applications. Current operational intelligence reconnaissance systems, such as the Predator, Hunter, and Pioneer Unmanned Aerial Vehicle (UAV) systems, do not incorporate digital video storage and retrieval capabilities. As a result, no effective means exists for intelligence analysts to retrieve archived video information about a specific target, without performing a painstaking manual search. Consequently, the goal of this research is to determine if video segmentation can provide a foundation for building a digital video storage and retrieval environment for continuous surveillance video.

As a precursor to understanding the methodology employed in this research, a fundamental appreciation for digital video issues is required, along with a background of the employment and operation of the UAV. This Chapter provides an overview of these issues in three main sections. The first section, 2.1, provides an overview of digital video storage and retrieval. The second part of the chapter, section 2.2, is the most important section of Chapter 2. In this section, a discussion of video segmentation is provided, along with overviews of the various categories of segmentation algorithms in the literature. Finally, in section 2.3, the Predator UAV system is discussed, including its system configuration, product dissemination, and data retrieval limitations.

2.1 Overview of Digital Storage and Retrieval

This section provides an overview of digital storage and retrieval. It is broken into three main sections. In section 2.1.1, a discussion is provided on the unique characteristics of video data. Section 2.1.2 provides an overview of the digital representation of video, including the digitization process and compression standards. In section 2.1.3, the fundamental issues of video database management systems are outlined, including data modeling, data insertion, data indexing, and data query/retrieval.

2.1.1 Characteristics of Video Data

Digital video is a medium with extremely high resolution and very rich information content. Along with metadata such as date, title, etc., video provides other detailed information, including object motion, and time lapse event occurrence. These temporal and spatial aspects of digital video make it different from textual or alphanumeric data stored in traditional database management systems. Additionally, the unstructured format and large volume of digital video make it difficult to efficiently and effectively manipulate, store, and retrieve.

As mentioned earlier, digital video has many unique characteristics when compared to other data types. Table 2-1 [ELMAG97] provides an overview of the essential differences.

Table 2-1. Comparison of Video with Textual and Image Data

| Criteria | Textual Data | Image Data | Video Data |
|---------------------|-------------------------|-------------------------|-------------------------|
| Information | Poor | Rich | Extremely Rich |
| Dimension | Static and Non-spatial | Static and Spatial | Temporal and Spatial |
| Organization | Organized | Unstructured | Unstructured |
| Volume | Low | Median to High | Massive |
| Relationship | Simple and Well-Defined | Complex and Ill-Defined | Complex and Ill-Defined |

- **Information:** Video data inherently contains much more information than plain textual data. Consequently, the interpretation of video depends directly on the viewer and the application being employed. The interpretation can be ambiguous and different for each viewer.
- **Dimension:** Textual data is static and non-spatial, and can be considered one-dimensional. Image data is spatial, but does not contain temporal properties, making it two-dimensional. On the other hand, video data has both spatial and temporal properties, and can be considered three-dimensional.
- **Organization:** Traditional data types, such as textual data, have a simple underlying structure. Existing database management systems store alphanumeric data with a finite symbol set. However, video data does not have a clear structure, and, as a result, is difficult to model and represent.
- **Volume:** Textual data by nature has a small data volume. Image and video data, on the other hand, have a much larger data volume. Image data is of

the magnitude of several thousand bytes, and one minute of video may contain over 1,000 image frames.

- Relationships: Alphanumeric relationship operators (equal, not equal, etc.) in traditional database management systems are simple and well defined. However, there are no universally accepted relationship operators for image and video data. As a result, video data indexing, querying, and retrieval are more difficult than their textual counterparts.

2.1.2 Digital Representation of Video Data

Understanding the issues associated with video database systems requires knowledge of how video is represented digitally. Digital video representation can be divided into two main components: video digitization and digital video compression.

2.1.2.1 Video Digitization Process

Figure 2-1 [PATEL97] provides an overview for the digitization process for digital video. The process begins with an input analog video signal. The input signal is based on the National Television System Committee (NTSC) standard video format. The NTSC format specifies 60 interlaced fields per second of analog video. Each field is constructed by 240 horizontal scan lines, which, using interlaced mode, produces 480 scan lines. Specialized computer hardware (video capture) is used to capture the analog video and create its

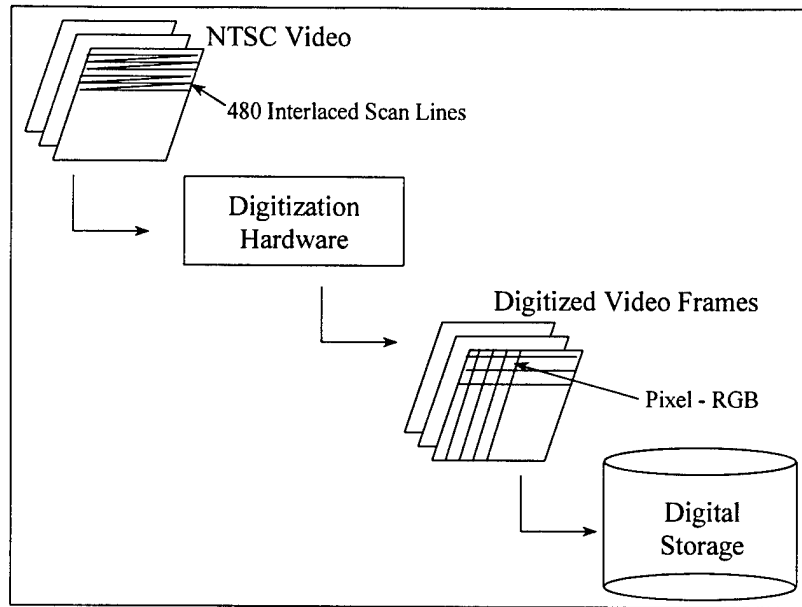


Figure 2-1. Video Digitization Process

digital form. The video capture hardware does this by sampling the scan line analog video signal and producing a digital value. The digitized samples are known as pixels, which represent the luminance and color in the image. Each scan line is sampled 640 times, producing a 640 X 480 rectangular grid digital video frame [PATEL97].

Once the analog signal has been captured and transformed by video capture hardware, it can be stored in digital form. Figure 2-2 [PATEL97] provides a description of the digital representation of video. The main component of digital video is frames, which are images with temporal positions as a function of time. Each frame is then composed of pixels. Pixels are defined as a function of spatial coordinates. Each pixel represents three color components for color video (or one brightness component for black and white

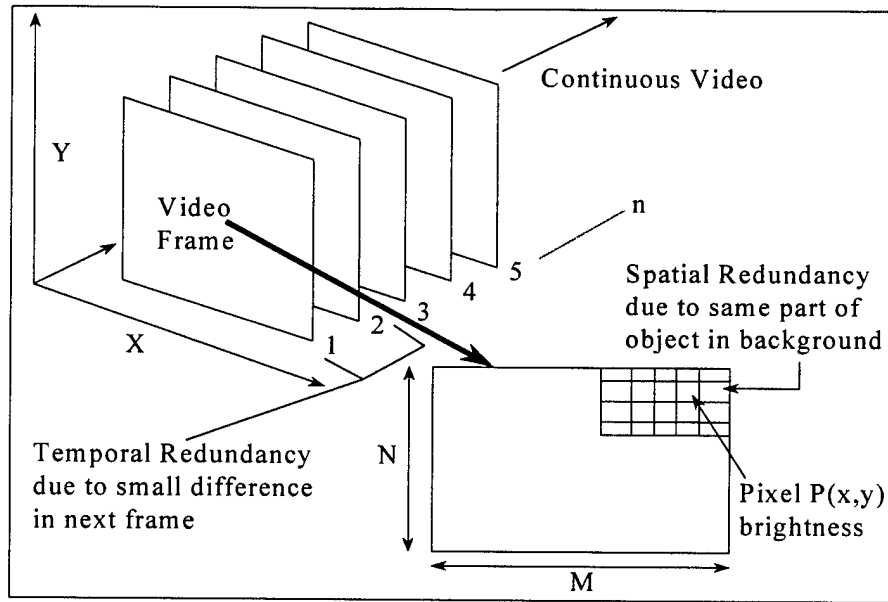


Figure 2-2. Digital Video Representation

video), and is enumerated in the range zero to 255. As a result, each color component of a pixel requires 8 bits (or one byte) of storage.

Based on the provided information about the digital representation of video, it is easy to see why digital video requires a large amount of storage and memory. For example, consider the memory requirement for digitizing video. At a frame rate of 30 frames/second, with full resolution (640 x 480 pixels), and true color (RGB/pixel), the digitization process requires the following amount of memory [PATEL97]:

$$30 \text{ frames/second} \times (640 \times 480 \text{ pixels}) \times 3 \text{ bytes/pixel} =$$

$$27,648,000 \text{ bytes/second}$$

At the current time, it is not feasible to have such large and expensive hardware to support video's memory and storage requirements. Therefore, much work has been accomplished towards developing video data compression techniques. The next section provides an overview of some of the most popular video compression standards.

2.1.2.2 Digital Video Compression Standards

As presented in the previous section, one of the large problems facing the design and use of video database systems is the large volume of data in raw video. For example, sixty seconds of uncompressed digital video can require upwards of one gigabyte of storage. Consequently, compression has shown to allow effective and efficient storage, transmission, and manipulation of digital video. The following sections provide an overview of some of the more popular video compression techniques and standards.

MPEG

The Motion Picture Expert Group, or MPEG, meets under the International Standards Organization (ISO) to create and maintain standards for digital video and audio compression. MPEG video compression is a block-based encoding scheme. Specifically, the standard defines a compressed bit stream, which implicitly defines a decompressor [PATEL97]. The video stream consists of a header, a series of frames, and an end-of-sequence code. Because the stream is temporally compressed (most frames build upon previous frames), there are

periodic Intra-Pictures, or I frames. I frames provide full images to be used as periodic references, and allow random access to the video stream. Other frames are predicted, using either a preceding I frame (creating a P frame) or a combination of preceding and following I frames (creating a B frame). The order and interspersion of the I, B, and P frames are determined by the encoder. A typical sequence would be I-B-B-P. However, the order of pictures in the data stream is not the order of display. For example, the previous sequence would be sent as I-P-B-B [KAYLE95].

Figure 2-3 [PATEL97] provides an overview of the basic structure of an MPEG video stream. Each individual color frame is converted to the YUV color space (where the Y component provides the luminance, and the U and V components provide the chrominance information). The frames are then further decomposed into smaller units known as *macro blocks* (16 x 16 pixels) and *micro blocks* (8 x 8 pixels). Macro blocks are created by breaking frames up into slices of 16 pixels high, and then breaking each slice up into a vector of 16 x 16 pixel blocks. Each macro block contains luminance and chrominance components for each of four 8 x 8 pixel micro blocks. For each macro block, a spatial offset difference between a macro block in the P and I frame(s) is given if one exists, providing a motion vector. The motion vector is then combined with a luminance and/or chrominance difference value. Macro blocks with no differences can be skipped except in an I frame. Blocks with differences are

internally compressed, using a combination of a Discrete Cosine Transform (DCT) algorithm on pixel blocks and variable quantization of the resulting frequency coefficients [KAYLE95].

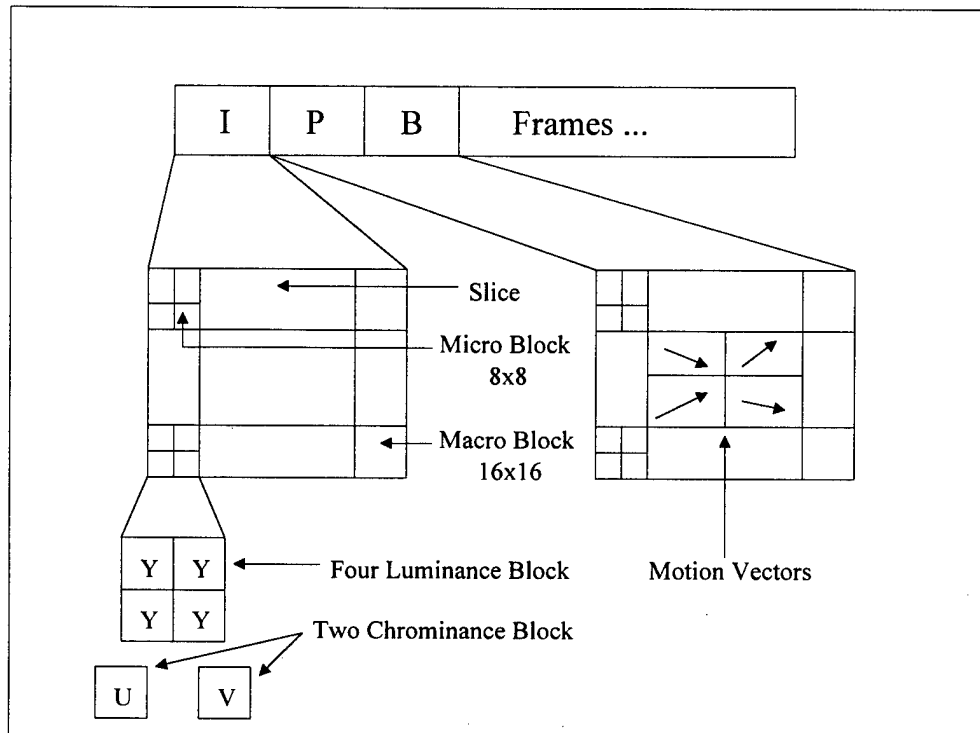


Figure 2-3. MPEG Structure

Figure 2-4 [SETHI95] provides an outline of the intra-block coding process. The DCT algorithm accepts signed 9-bit pixel values and produces signed 12-bit coefficients. The DCT is applied to one micro block at a time, converting each 8×8 block into an 8×8 matrix of frequency coefficients. The variable quantization process divides each coefficient by a corresponding factor in a matching 8×8 matrix and rounds to an integer. Quantization results in numerous zero coefficients, particularly for high-frequency terms at the high end

of the matrix. Accordingly, amplitudes are recorded in run-length form following a diagonal scan pattern from low frequency to high frequency. All control data, vectors, and DCT coefficients are further compressed using Huffman-like variable-length encoding [KAYLE95].

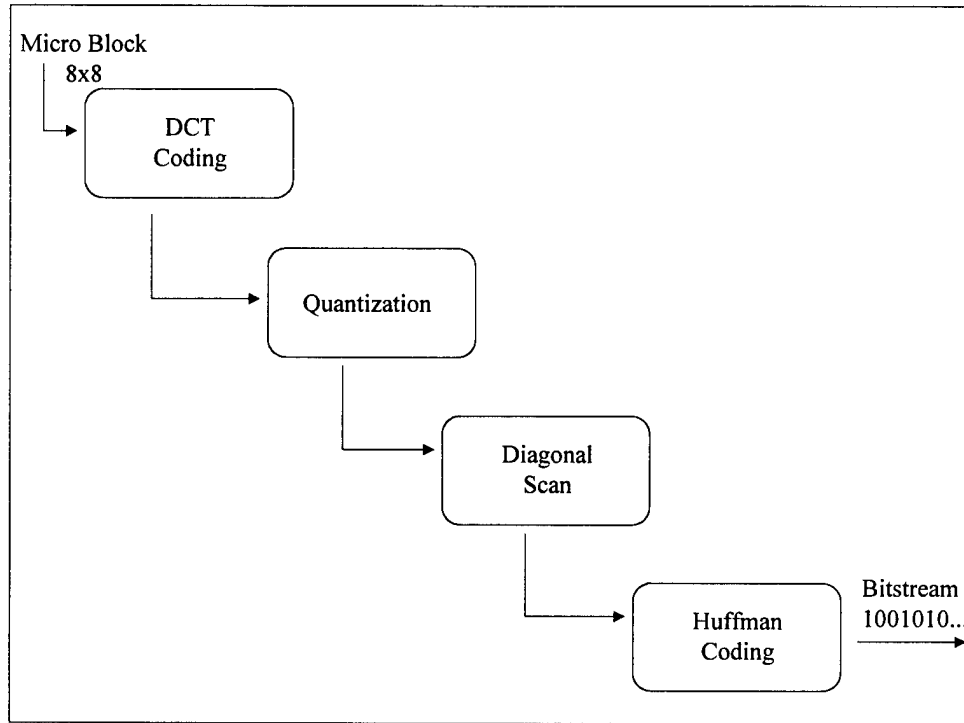


Figure 2-4. Intra-Block Coding Process

MPEG provides very good compression, but requires time-wise expensive computations to decompress the data for display purposes. As a direct result, achievable frame rates are greatly limited on systems that currently use MPEG compression. In spite of this disadvantage, it is widely believed that MPEG will soon become the *de facto* standard for all home and industry video applications [ELMAG97].

JPEG/MJPEG

JPEG is a standardized image compression scheme developed by the Joint Photographic Experts Group. It was designed for compressing either full color or gray scale images of real-world scenes. It works extremely well with photographs and naturalistic artwork. JPEG only handles still images, and is a lossy compression scheme. JPEG achieves its dramatic compression ratios (from 25:1 to 75:1) by exploiting known limitations of the human eye, most notably, that small changes in color are perceived less accurately than small changes in brightness [ELMAG97]. Therefore, JPEG compression is meant to be used on images viewed by humans, rather than by machines. The small errors introduced by JPEG compression may cause problems for any application that requires a machine interpretation of an image.

Although not strictly part of JPEG compression, most implementations start by converting the RGB image into a luminance/chrominance (YUV) color space. The raster data may be subsampled, combining adjacent pixels into a single value. A Discrete Cosine Transform (DCT) is then applied to convert the raster data into rate-of-change information. Quantization truncates the results of the DCT coding to a smaller range of values (this step makes JPEG lossy). The quantization coefficients determine how much data is lost, the extent of compression, and the quality of the reconstructed image. Finally, the results of quantization are compressed using either Huffman or arithmetic coding to

produce the final output. Decompression reverses the above steps, decompressing the quantized results and using a reverse DCT to reconstruct the image. The low-order bits lost to quantization are not reconstructable, so the decompressor inserts zeroes for them [KAYLE95]. This loss of bits results in a degradation of color richness.

MJPEG is simply motion JPEG. Although there is no uniform standard for MJPEG, various vendors have applied JPEG compression algorithms to individual frames of a video sequence and called the result MJPEG. Usually, the various versions of MJPEG implemented across vendor boundaries are not compatible. Despite the lack of compatibility, MJPEG has several advantages when compared to MPEG:

- Frame-based encoding (good for accurate video editing)
- Fairly uniform bit rate
- Simpler compression (no cross frame encoding; requires less computation)

The major disadvantage of MJPEG is that it does not support inter-frame compression (which MPEG does support). This makes the compression ratio for MJPEG about 3 times worse than MPEG [ELMAG97].

H.261

H.261 is a widely used international video compression scheme allowing the frame rates required for real-time video conferencing. H.261 provides a

standard scheme for video encoding and decoding of the moving picture component of an audiovisual service at rates approaching 2 Megabits per second. It was designed to be suitable for applications using circuit-switched networks as their transmission medium (such as telephone service). H.261 was originally targeted for Integrated Services Digital Network (ISDN), and has many hardware/software implementations (e.g. PC video cards). The H.261 encoding algorithm is very similar to that of MPEG, however, there are some differences, the largest being that the two schemes are not compatible. Additionally, H.261 requires less CPU power for real-time encoding than MPEG. Another difference is that H.261 includes a mechanism that optimizes available bandwidth by sacrificing picture quality for motion. This causes a quickly changing picture to have a poorer quality than a static picture [ELMAG97].

DVI

Digital Video Interactive, or DVI, is a compression scheme developed by Intel® based on the region encoding technique. Each frame of a video sequence is divided and sub-divided into regions until they can be mapped onto basic shapes that fit the allotted bandwidth and required quality of the video application. The decoder can accurately reproduce the given shapes at the receiving end. The actual data sent over the network is a description of the region tree and the shapes of the leaves of the tree. DVI is an asymmetric coding scheme; it requires a large amount of processing for encoding and much less for

decoding. Although not a true standard, the DVI format is used extensively in commercial applications and on the Internet and World Wide Web (WWW) [ELMAG97].

QuickTime

QuickTime® is Apple's® cross-platform file format for the storage and interchange of sequenced data. Similar to DVI, it is not currently a standard, but is heavily used on the Internet and WWW. QuickTime movies are made up of time-based data streams, which may contain sound, video, or other sequenced information. The QuickTime data streams are built up from basic units known as *atoms*, which describe the format, size, and content of the movie storage element. Atoms can be nested recursively within *container atoms*. One type of container atom is the *movie atom*, which provides the time scale, duration, and display characteristics for the entire movie file. Movie atoms also contain one or more atoms that describe a single track of the movie, independent of other atoms and carrying their own spatial and temporal information. These atoms are known as *track atoms*, and contain data such as editing information, track priority in relation to other tracks, and display/masking characteristics. QuickTime also supports many other types of atoms, including text and media atoms. Although not currently a standard, QuickTime has the potential to become the computer industry standard for the interchange of video and audio sequences [ELMAG97].

2.1.3 Video Database Management System

As stated earlier, the unique aspects of digital video make it much different from textual or alphanumeric data stored in traditional database management systems. Digital video's spatial and temporal properties, volume, and complex relationships make it impractical to store and retrieve in an ordinary DBMS. In an effort to provide efficient and effective storage and management of digital video data, much research has been accomplished towards developing a Video Database Management System (VDBMS). A VDBMS is a software system that manages a collection of video data and provides content-based access to users [HAMPA95]. As in any other DBMS, the goal of a VDBMS is to provide an environment both convenient and efficient for retrieving and storing video information in a database [YEOYE97]. Figure 2-5 [ELMAG97] depicts the components of a generic VDBMS. Similar to traditional database management systems, a VDBMS must provide the essential data management functions of data modeling, data insertion, data indexing, and data query and retrieval. The video-unique aspects of these are briefly described below.

2.1.3.1 Video Data Modeling

Video data modeling deals with the representation or abstraction of video data. Specifically, video data modeling is the process of designing the representation for the video data based on its characteristics and information

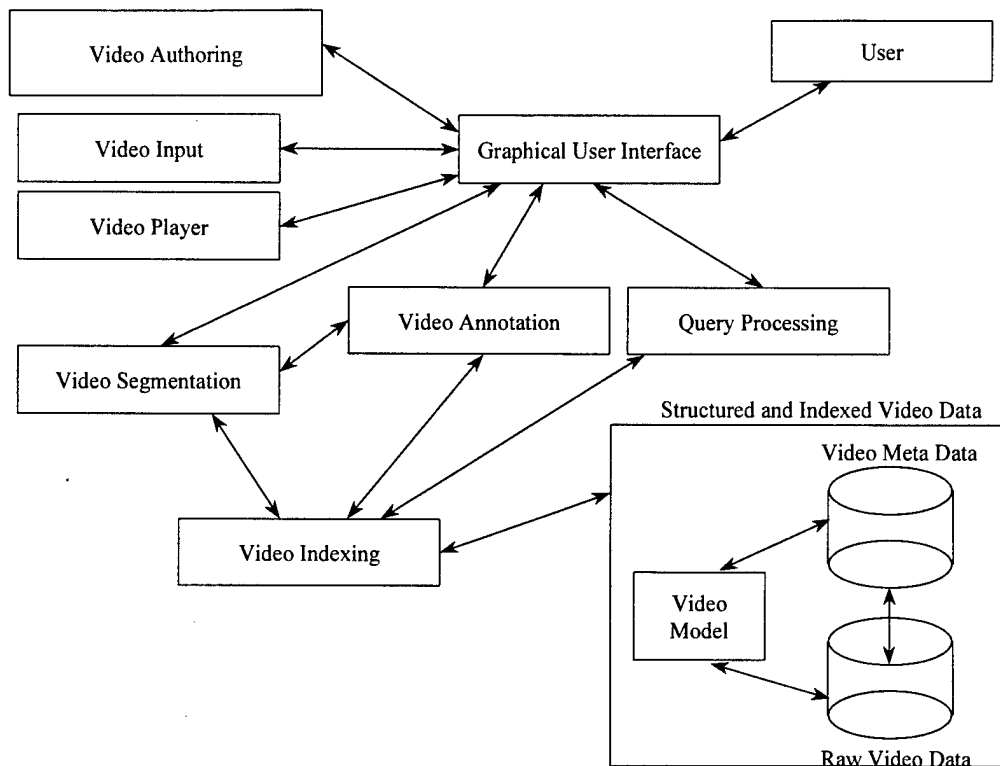


Figure 2-5. Generic VDBMS

content, and the application the data is intended to be used with. Modeling video data facilitates operations in the VDBMS such as data insertion, editing, indexing, browsing, and querying. As a result, constructing a video data model is usually the first task accomplished in the process of designing a VDBMS. Video data modeling is important in the design of a VDBMS because the abstraction chosen for the model directly determines which features will be used in the indexing process. Thus, the video data model can greatly impact the performance of the VDBMS.

Requirements for a Video Data Model

As a traditional database management system supports textual and numeric data, a VDBMS should support digital video as one of its native data types. To accomplish this, the underlying video data model should integrate the content attributes of the video data along with a description of its semantic structure. Physical objects contained within the video sequence, such as people, vehicles, buildings, etc., should be described, along with any associated audio communications relating to those objects. Spatial and temporal relationships among video segments should be expressed. The model should also support the automatic extraction of features such as color, texture, shapes, and motion [HAMPA95]. The following sections elaborate some of the essential requirements the video data model must support.

Multi-Level Abstraction Support

All video stream data contains two basic levels of abstraction: the entire video and individual frames. For most applications, including military reconnaissance, the entire video is too coarse a level of abstraction to be of any practical use. Conversely, a single frame is of little interest since it spans an extremely short interval of time (NTSC video specifies 30 frames per second). Consequently, other levels of abstractions are often desired, namely *shots* (also called scenes or clips, described below), and thus a hierarchical or multi-level

abstraction of video data can be formed. Multi-level abstraction support has several advantages [HJELS96]:

- Allows easy reference of video information and simplifies comprehension of its contents
- Provides good support for video browsing
- Simplifies video indexing and storage organization

The video shot is considered by some the basic structural element for characterizing video data [HAMPA95]. A shot is a contiguously recorded series of frames that represent continuous action in space and time [SETHI95]. Shots that are related in time and space can be combined to form an *episode*. Figure 2-6 provides a graphical depiction of the hierarchical abstraction of video data.

Spatial and Temporal Relationship Support

As stated earlier, the spatial and temporal characteristics of video data make it quite different from traditional data types. These characteristics underscore the importance that the video data model identifies physical objects contained within the video as well as their relationships in space and time. A typical user of a VDBMS may wish to issue queries that contain both spatial and temporal constraints. For example, temporal relationships (such as *before*, *meets*, *overlaps*, *during*, *starts*, *finishes*, *equals*) may be used in formulating queries to restrict the returned video sequences. Three-dimensional spatial relationships may also be used to determine which video sequences satisfy a

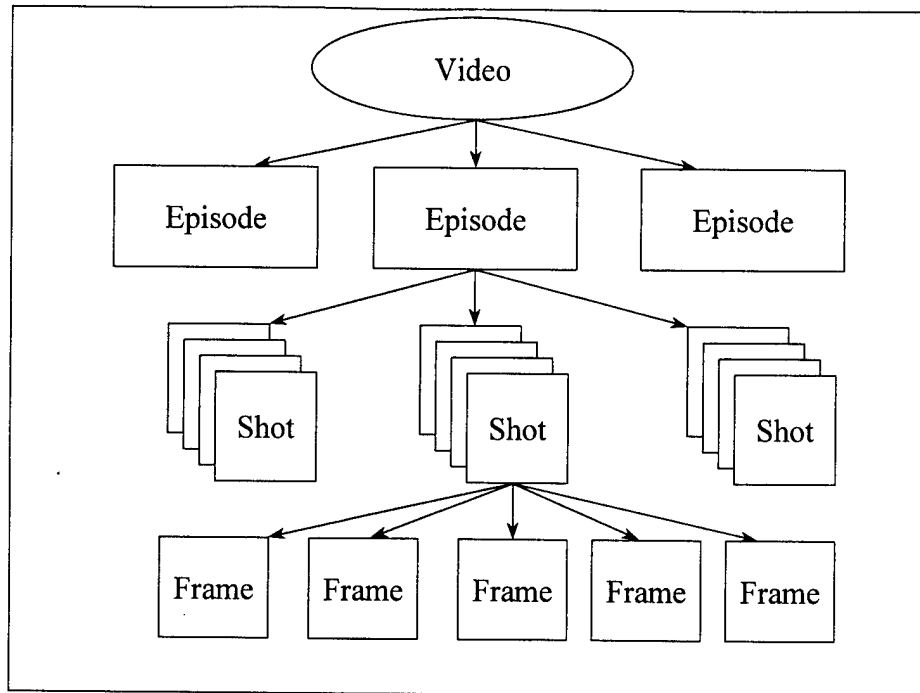


Figure 2-6. Hierarchical Abstraction of Video Data

given query. At the current time, very few research attempts have been made at the formal representation of spatio-temporal relationships of video objects and queries based on those relationships [HJELS96].

Video Annotation Support

The video data model should provide for simple and dynamic annotation of video data stored in a VDBMS. In contrast to textual and alphanumeric data types, digital video does not easily accommodate automatic feature extraction. Furthermore, the very structure of video data, while capturing some aspects, may not be suited for the representation of every characteristic of the material that it represents. For example, given videos of two city blocks, it may be

impossible to determine which city block is in Ohio and which is in Tennessee. For these reasons, it should be possible to link variously detailed descriptions of video content to arbitrary frame sequences. Also, many annotations may be modified since the interpretation of the human user and the application domain may change. As a result, the annotations must be dynamic. Presently, the video annotation procedure is mostly a manual, off-line process [GUPTA97].

2.1.3.2 *Video Data Insertion*

Video data insertion deals with introducing new data into the database management system. The insertion process usually consists of the following steps:

- Feature Extraction - Key information or features are extracted from the data for instantiating the video data model.
- Video Segmentation - The video stream is broken down into a set of basic units (shots) for indexing and retrieval purposes.
- Annotation - Based on the application domain, the necessary additional attributes are added to the video data (such as titles, times, areas, names, etc.).
- Indexing and Storage - Based on the features extracted from the raw video data and the subsequently added annotations, indexes are built on the data, and the data is stored in the VDBMS.

2.1.3.3 *Video Data Indexing*

Due to the large data volume in a VDBMS, retrieving the actual video can become extremely time consuming. As with text-based data, indexing organizes the video data in the VDBMS to make user access such as querying and browsing more efficient. Consequently, video data indexing is considered by some the most important step in the video data insertion process [JAGAD97].

When compared to simple indexing done in traditional database management systems, several factors make video indexing more difficult and complex [LIENH97]. First, in traditional database management systems, data is usually organized according to one or two key fields on which the data can be uniquely identified. However, the choice of unique attributes in a video is not as easy to determine. Unique identifiers for video data could be audio-visual features, user-supplied annotations, spatio-temporal information, or some combination of the three. Another complicating factor is the automatic generation of indexes. Traditional database management systems can automatically generate indexes based on the values in key fields. However, generating indexes based on the content of video data is not simply field-based, but value-based.

Despite the difficulties of video data indexing, much work is being accomplished towards developing indexes for video data. The ongoing work

can be classified into two main categories: annotation-based indexing and feature-based indexing.

Annotation-Based Indexing

Annotation-based indexing allows access to the data on semantic content rather than external information or attributes. Currently, automatic annotation of video data is unrealistic due to limitations in machine learning and computer vision. Consequently, annotation-based indexing is a manual process requiring frequent human intervention. The manual annotation process is usually performed by an experienced human user, such as an intelligence imagery analyst. Manual annotation of video data has several disadvantages:

- Manual annotation is time consuming, and thus is not appropriate for large quantities of video data
- Annotation is application dependent; therefore, certain domains may not be applicable to other applications
- Annotation is biased and limited by the human expert performing the work

Due to the disadvantages of manual annotation, many existing indexing techniques concentrate on the selection of keywords (usually from the annotations), data structures, and user interfaces to facilitate user access [ELMAG97].

Feature-Based Indexing

In contrast to the annotation-based indexing approach, feature-based indexing seeks to fully automate the indexing process. In feature-based indexing, image-processing algorithms segment the video data stream, identify representative frames, and extract key features from the data. Key features can be characteristics such as color, object motion, texture, detected edges, etc. Once key features are extracted, indexes can be built on those representative features. The main advantage of feature-based indexing is the ability to automatically generate indexes without human intervention, thus, saving time and reducing indexing errors. However, feature-based indexing lacks the ability to associate semantics with the extracted features. This disadvantage can cause serious problems with the typical semantic-based queries which video database management systems should support [ELMAG97].

2.1.3.4 Video Data Query and Retrieval

Video data query and retrieval deals with the extraction of video data from the VDBMS that satisfies user-specified search criteria. The search criteria may not require an exact match (similar to the types of queries performed in a traditional DBMS), but a *best match*. For example, a user may wish to see all video sequences with a frame *similar* to a given still image. Additionally, the search criteria may involve the content of the video, not just annotated information or metadata associated with the video sequence. For example, a user

may want to see all video sequences that involve a dog chasing a man. Consequently, the video data query and retrieval process is more complicated and computationally expensive than its text-based counterpart [HAMPA95].

The typical VDBMS query and retrieval process involves the following steps. First, the user expresses the query through the user interface. Once the query has been entered, it is processed and evaluated by the query processor. The attributes and feature values specified by the query are used to retrieve the corresponding video sequences from the database. Usually, the features specified in the query are directly related to the indexing structure(s); therefore, the query is processed by searching the indexing structure(s). Once the appropriate video sequences are retrieved by the database system, they are displayed through the user interface in an intelligible form.

2.2 Video Segmentation

As stated earlier, video database management systems must address the problems of data modeling, insertion, indexing, and query/retrieval. One fundamental aspect that has a great impact on these problems is the content-based temporal sampling of video data. Temporal sampling of video frames seeks to identify significant video frames for representation, indexing, storage, and retrieval of the data. Automatic content-based temporal sampling of video data is application dependent, and usually requires semantic interpretation of the video content. Since the artificial intelligence techniques (machine learning,

object recognition, image processing) required for this type of sampling are relatively immature, automatic content-based temporal sampling is not currently feasible [ELMAG97].

However, satisfactory results can usually be obtained by decomposing the video into segments by determining the boundary (scene break) between consecutive camera shots. The isolation of shots is of interest since shot-level organization is considered by some the most appropriate level of abstraction for video browsing and content-based retrieval [YEOYE97]. The process of decomposing video into shots is referred to as *video segmentation*. Selecting one representative frame from each shot, since a shot is a continuous sequence of video frames that have no significant inter-frame difference in terms of visual content, can then approximate content-based temporal sampling. The partitioning of video into shots is usually achieved by measuring inter-frame differences and detecting sharp peaks. There are many video segmentation algorithms in the literature [BOUTH97] [LEEIP95] [MENGJ95] [SETHI95] [VASCO97] [WANGA94] [XIOLE95] [XIOIP95] [YEOLI95] [ZABIH97], and they can be classified into several different categories based on the methods they use to determine scene breaks. The following sections provide an overview of the various categories.

2.2.1 Pairwise Comparison

Algorithms using the pairwise comparison method determine scene breaks by examining successive frames pixel by pixel. The algorithm takes as input two image frames, I and $I+1$. Each pixel of I is compared with the corresponding pixel in $I+1$, and the difference between the two pixels (if there is one) is added to the total difference between the two image frames. If the total difference between two successive image frames exceeds a specified threshold, a scene break is declared. Pairwise comparison may be used on color as well as grayscale image frames [XIOIP95].

2.2.2 Likelihood Comparison

The likelihood comparison algorithm takes as input two successive image frames, I and $I+1$. First, each frame is divided into uniform regions and the mean and variance of intensity values of each region is computed. The mean and variance of corresponding regions of I and $I+1$ are then compared to establish a likelihood comparison factor. The likelihood comparison factor provides an indicator as to whether two sets of values came from the same probability distribution. The likelihood comparison factor between corresponding regions of I and $I+1$ is compared against a specified threshold. If the comparison factor exceeds the threshold, a 1 is added to the total frame to frame difference value (indicating the two regions are not from the same distribution). Each region of I is compared to the corresponding region in $I+1$ in this manner, and the results

are added to the total frame to frame difference value. If the total frame to frame difference exceeds a specified threshold, a scene break is declared. Likelihood comparison may be used on color as well as grayscale image frames [XIOLE95].

2.2.3 Global Histogram

Algorithms using histogram techniques determine scene breaks by measuring probability distributions of pixel values in a given image. The global histogram is computed by dividing the color space (either full color or grayscale) into discrete *bins* and counting the number of pixels that fall into each bin. The difference between two image frames I and $I+1$ is determined by comparing their histograms. Each bin in the histogram of I is compared to the corresponding bin in the histogram of $I+1$. The difference between bins is added to the total frame to frame difference value, and if this value exceeds the specified threshold, a scene break is declared [XIOIP95].

2.2.4 Local Histogram

In this method, a frame is divided into uniform, non-overlapping regions. Histograms of each region are computed (in the same manner they are computed in the global histogram method) and compared to the corresponding histograms from the successive frame. As in the global histogram method, corresponding bins from each histogram are compared, and the difference is added to the regional difference. The total frame to frame difference is calculated by summing

all the regional differences between the two successive frames. If the total difference exceeds the specified threshold, a scene break is declared [XIOLE95].

2.2.5 *Weighted Color Histogram*

In certain video sequences, a series of frames may contain a dominant color. Based on the application domain, the dominant color may be given a greater weight in determining if a scene break occurs between two frames. In the weighted color histogram method, a histogram is created for each of the successive image frames being compared. The histogram is then weighted by the luminance values for the color space being used (for example, red, green, and blue would be used for RGB). The histograms between the two successive image frames are compared, and if the difference exceeds a specified threshold, a scene break is declared [DAILI95].

2.2.6 *Edge Detection Segmentation Method*

The edge detection segmentation method proposed in [ZABIH97] is based on the observation that during a shot transition new intensity edges appear far from the locations of old edges, and old edges disappear far from the location of new edges. An edge pixel that appears far from an existing edge pixel is defined as an *entering* edge pixel, and an edge pixel that disappears far from an existing edge pixel is an *exiting* edge pixel. By counting the number of entering and exiting edge pixels, this algorithm can detect and classify cuts, fades, and dissolves.

The algorithm takes as input two consecutive image frames, I and $I+1$. An edge detection step is performed, which results in two binary image frames E and $E+1$. The term ρ_{in} denotes the percentage of edge pixels in $E+1$ which are more than a fixed distance r from the closest edge pixel in E . Thus, ρ_{in} represents the proportion of entering edge pixels. It should assume a large value during a fade in, cut, or a dissolve. Similarly, the term ρ_{out} denotes the percentage of edge pixels in E which are farther away than a fixed distance r from the closest edge pixel in $E+1$. Thus, ρ_{out} measures the proportion of exiting edge pixels, and should assume a high value during a fade out, cut, or a dissolve. Using the previously describe terms, the dissimilarity measure is the following:

$$\rho = \max(\rho_{in}, \rho_{out})$$

This measure of dissimilarity represents the fraction of changed edges. Scene breaks can be determined by searching for peaks in ρ , which is known as the *edge change fraction*.

2.3 Predator Unmanned Aerial Vehicle

As stated earlier, the focus of this research was to determine if video segmentation could provide a foundation for building a digital video storage and retrieval environment for continuous surveillance UAV video. The first two sections of this chapter provided an overview of digital video storage/retrieval and video segmentation. To have a complete understanding of the approach taken in this research, it is critical to comprehend some of the elementary issues

associated with the Predator UAV. This section provides a basic profile of the UAV system.

2.3.1 Predator UAV System Background

In the Desert Storm conflict, numerous military commanders were frustrated with their inability to obtain timely imagery intelligence. This inability illustrated the need for a long dwell, theater controlled, imagery reconnaissance capability with sufficient range and endurance to cover a typical target area. The result of these requirements was an Advanced Concept Technology Demonstration (ACDT) system known as the Medium Altitude Endurance Unmanned Aerial Vehicle (MAE-UAV), or Predator. Predator is a tactical reconnaissance system with an airborne platform that provides high quality still and motion, color and gray-scale imagery of tactical targets. Predator can loiter over a target 500 nautical miles from its launch point, providing live imagery and narration (provided by ground personnel), exploited still imagery, and textual reconnaissance reports to theater commanders through standard military networks [WIEDE97]. The following sections provide a brief overview of Predator's system configuration, product dissemination, and data retrieval limitations.

2.3.2 Predator UAV System Configuration

The standard Predator system configuration consists of three to four Predator air vehicles and their ground support equipment. The air vehicle is composed of carbon fiber composite materials and designed with high aspect ratio wings. This design sacrifices speed to produce high efficiency and long operating endurance (see Figure 2-7 for the Predator in flight).

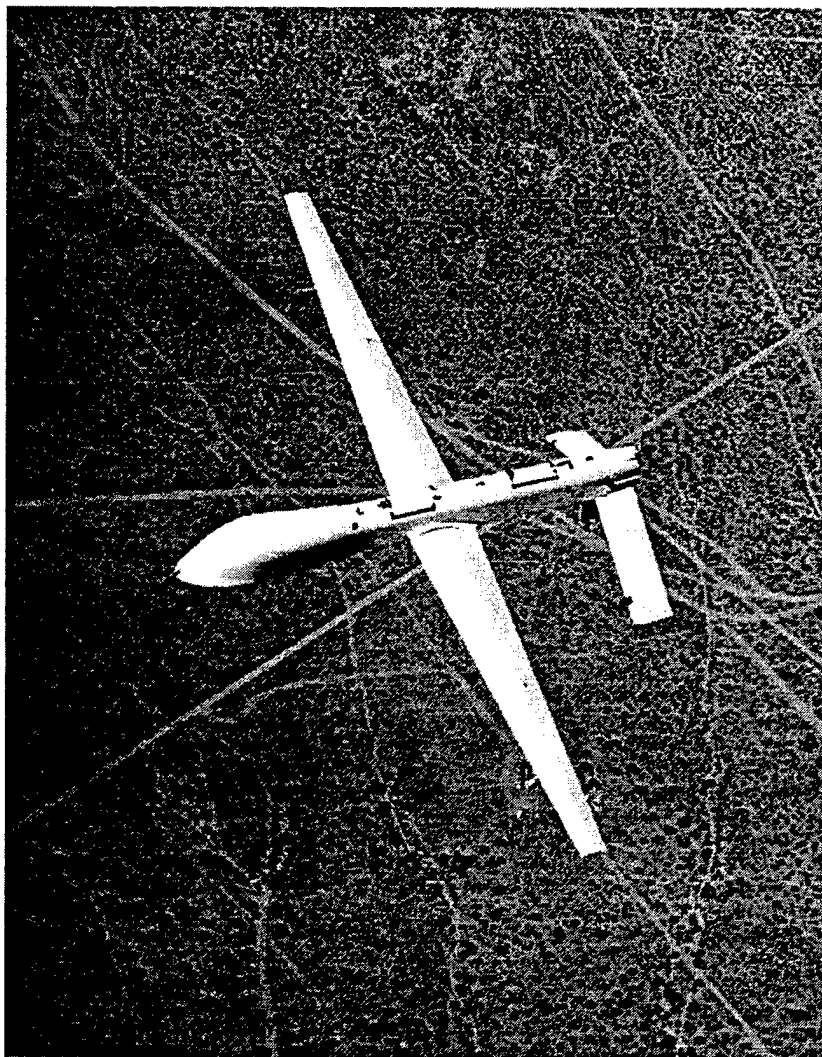


Figure 2-7. Predator in Flight

The air vehicle contains several reconnaissance sensors, including motion imagery cameras (visible light), infrared cameras, and Synthetic Aperture Radar (SAR). The typical system also includes a Ground Control Station (GCS) which houses the imagery exploitation personnel. Ground Data Terminals (GDT) provide communication links between the air vehicle and the GCS, and between the GCS and SATCOM dissemination systems [WIEDE97].

2.3.3 Predator UAV Product Dissemination

Predator motion video is typically downlinked to the GCS, where it is overlaid with telemetry and support data (including Latitude, Longitude, Elevation, etc.). The composite video data is transmitted to a SATCOM shelter, where it is converted to MPEG, and transmitted over satellite to critical theater operational centers (US Sector HQ at Tuzla, Joint Endeavor HQ in Sarajevo, Combined Air Operations Center at Vicenza, and Joint Analysis Center (JAC) at Molesworth, UK). At these command centers, the compressed video is decompressed and converted to analog, where it is then presented to command and analytical users. Additionally, significant video frames are captured and converted to National Imagery Transmission Standard Format (NITSF) for transmission to JAC Molesworth via SIPRNET. Once at JAC Molesworth, the images are added to the 5D (Demand Driven Direct Digital Dissemination) database system, which is accessible by most of the operational intelligence community [WIEDE97].

2.3.4 Predator UAV Video Data Retrieval Limitations

Predator has been deployed on two separate military operations in Bosnia, Operation Nomad Vigil and Operation Nomad Endeavor. In both operations, every hour of sensor operation was recorded on 8mm-video tape, creating over 2,000 hours of video on over 1,000 tapes. In this scenario, it is difficult to find any particular target or scene of interest, especially if the date of collection is unknown. The current process requires an analyst to perform a painstaking manual search of the tapes. Given that on average at least half of the tapes would need to be viewed to find a particular target, an analyst would have to view over 1,000 hours of video. Expanding collection systems, future deployments, and personnel cutbacks further exacerbates this number. Under this scenario, it is exceedingly difficult for an imagery analyst to find a scene or target of interest in the Predator video archive [WIEDE97].

2.4 Summary

Digital video is an extremely rich medium with many unique characteristics, including massive volume, spatio-temporal characteristics, and complex relationships. Consequently, to provide effective and efficient storage and management of video data, much research has been accomplished towards developing a video database management system. Critical to the design of any database system, a VDBMS must address the issues of data modeling (including multi-level abstractions, spatio-temporal relationships, and annotation), data

insertion, data indexing (including annotation-based indexing and feature-based indexing), and data query and retrieval.

A fundamental aspect that greatly affects each of these issues is the temporal sampling of video frames, which facilitates the representation, indexing, storage, and retrieval of data. To achieve the temporal sampling of video data, a procedure known as video segmentation is performed. Video segmentation is the process of decomposing a video into shots by determining the boundary between camera shots. There are many segmentation algorithms in the literature, each of which seeks to exploit some known characteristic of video data. Table 2-2 summarizes the various methods described in section 2.2.

Table 2-2. Summary of Segmentation Algorithms

| Method | Description |
|--------------------------|---|
| Pairwise Comparison | Examine successive frames pixel by pixel |
| Likelihood Comparison | Divide frames into uniform regions; compute mean and variance for each region; compute likelihood factor; compare factor region by region for successive frames |
| Global Histogram | Divides color space into bins; count colors that fall into each bin; compare corresponding bins for successive frames |
| Local Histogram | Divide frames into uniform regions; create histograms for each region; compare histograms region by region for successive frames |
| Weighted Color Histogram | Create histogram for each frame; weight histogram based on color space; compare histograms for successive frames |
| Edge Detection | Detect edges for each frame; count entering and exiting edge pixels between successive frames; compute edge change fraction between frames based on max of entering/exiting edge pixels |

Predator UAV is a tactical reconnaissance system developed as an ACTD in support of operational requirements. The system consists of an air vehicle, a ground control station (GCS), and ground data terminals (GDT) that provide communication links. The air vehicle contains several data sensors, which provide mission critical intelligence products to commanders and analysts. Although Predator UAV furnishes numerous hours of video on completed missions, there is no simple way to search and retrieve video data on a specific target.

3 METHODOLOGY

Currently, imagery intelligence analysts have no simple, automated way to search and retrieve UAV video data on a target or area of interest. The present method is a painstaking, by-hand examination of thousands of tapes. Even after the correct tape is found, a manual exploration of the tape must be performed to find the scene of interest. Future UAV missions coupled with personnel cutbacks will create a difficult environment for search and retrieval of data. Consequently, an automated data management system must be developed for UAV video data.

The most logical approach to provide this capability is to store the UAV video data in a video database system. Critical to the storage of any type of video data in a database management system is the temporal sampling of data. As described in Chapter 2, temporal sampling provides significant frames for representation, indexing, storage, and retrieval. Although automatic content-based temporal sampling of video is not presently feasible, video segmentation does provide a viable alternative.

Video segmentation decomposes videos into segments by determining scene breaks between consecutive camera shots. This process works relatively well with man-made, edited video footage. However, it was unknown at the time of this research if applying video segmentation to UAV video footage would provide meaningful scene breaks for storage and retrieval purposes.

Accordingly, the focus of this research is to analyze the applicability of applying state-of-the-art video segmentation to UAV video data.

The methodology of this research consists of three major steps. First, an applicable video segmentation algorithm is selected from the literature. Once an algorithm has been selected, UAV test data are gathered for experimental purposes. Finally, an experiment is performed to analyze the applicability of the segmentation algorithm. The following sections provide a description of each of these major steps.

3.1 Algorithm Selection

Many different methods of automatic video segmentation have been proposed (see section 2.2). Each of these seeks to exploit some known characteristic of video data, and using that characteristic, define a measure of dissimilarity between successive frames of a video. Although several of the algorithms are effective in their respective domains, only the edge detection method [ZABIH97] was chosen for this research. The following sections describe the rationale behind the selection of the edge detection segmentation method.

3.1.1 Algorithm Comparison

Research performed in [DAILI95] examined several different video segmentation techniques by systematically comparing their performance across different types of videos. In particular, an ABC news video was used because it contained a large variety of shot-transition effects, such as cuts, fades, dissolves,

and wipes. It also contained several scenes with short duration and fast object or camera motion. In this experiment, the various segmentation methods tested from section 2.2 included pairwise comparison, histogram, weighted histogram, and edge detection. Along with those methods, the chi squared, pure moment invariant, and range of pixel-value changes methods were also tested.

The majority of the methods correctly identified the transitions 95 percent of the time or more. However, many of the methods had a corresponding high percentage of false transition identifications. In a situation where a human may intervene or serve as a filter, a high percentage of correct identifications is much more important than a low number of false identifications. This is because the human observer can filter out the false positive transitions. However, in situations where there is no human intervention, such as automatic segmentation for a large digital library, a low percentage of incorrect transition identifications becomes increasingly important.

Of the various methods tested, many had two times as many false transition identifications as they had correct transition identifications. However, the edge detection method described in section 2.2.6 had a relatively low number of false transition identifications when compared to the other methods. The only method with a lower number of false positives was the pairwise comparison method. Nevertheless, its percentage of correct transition identifications was only 73 percent, while the edge detection method was 92 percent.

3.1.2 Data Characteristics

An important consideration in the choice of segmentation algorithms was the UAV data characteristics. The UAV typically hovers for several minutes over an area of interest (for example, performing battle damage assessment or surveillance of a possible target). During data collection, the Ground Control Station (GCS) may switch the single camera feed between one of the several visible light imagery cameras on the UAV. The abrupt change between two different cameras can be considered a scene change. Additionally, the visible light cameras may perform an abrupt zoom or pan across an area or between two different areas of interest. Stationary camera shots interspersed with abrupt camera motion can also be considered scene changes. Since UAV footage usually contains objects with sharp edges (such as tanks, surface-to-air missile sites, buildings, roads, etc.), scene changes will cause a large change in edges between successive video frames. As a result, the edge detection method proposed by [ZABIH97] was chosen for this research.

3.2 UAV Test Data

Once an algorithm had been selected, appropriate UAV video footage was needed for experimental purposes. This section describes the process for acquiring the test data, along with a description of the types of video sequences chosen for testing the algorithm developed by [ZABIH97].

3.2.1 Test Data Acquisition

The UAV video footage used for experimental purposes in this research was provided by Air Force Research Laboratory's (AFRL) Signal Data Handling Branch (AFRL/IFEC). Over seven hours of UAV footage were provided on analog VHS and 8mm tapes. Using *SNAZZI*TM [DAZZL97] video capture software and hardware at the 88 Communications Group Multimedia Center, the footage was converted to MPEG format. Despite being in standard digital format, the data requires further manipulations since the algorithm provided by [ZABIH97] expects a sequence of raw video frames in Portable Graymap (PGM) format. To accomplish this transition, a software package called *mdcdecoder* [MPEGDC98] is used to convert the MPEG sequences to raw Portable Pixmap (PPM) format. The final conversion from PPM to PGM format is carried out by the *Convert* utility of the *ImageMagick*TM [CRIST96] software suite.

3.2.2 Video Sequence Selection

Once a methodology is developed to convert the analog UAV data to PGM format, sequences can then be selected as test cases for the edge detection algorithm. During the selection process, it is important to include scenes that encompass a wide variety of the typical visual effects a UAV may encounter, such as the following:

- Abrupt changes - During data collection, the GCS may switch the single camera feed between one of the several visible light imagery

cameras on the UAV. Additionally, the camera feed may be turned on/off as areas of interest are entered or exited. These abrupt changes are common on a typical UAV mission.

- Camera Zooms - When a specific target or area of interest is detected, the UAV camera typically performs a *zoom in* to enhance the detail of the imagery sent to the GCS. Likewise, to provide a broader view of an area, a *zoom out* is performed.
- Motion Across Scenery - On a typical mission, the UAV films large segments of countryside and/or residential areas while approaching the area of interest. In some instances, the UAV travels at a relatively slow speed, and the segments are usually filmed from a high altitude, causing the change of scenery to be gradual. In other instances, the UAV may be traveling at a relatively low altitude, causing the change of scenery to be more abrupt in nature.
- Stationary Shots Separated by Rapid Motion - Occasionally, a UAV may be responsible for performing surveillance on several targets in the same locale. In this situation, the imagery feeds on the targets are interspersed with rapid motion as the visible light camera quickly rotates from one target to another.
- Cloud Cover - In some instances, a UAV may encounter cloud cover as it attempts to perform surveillance on a target. In this situation, a

stationary shot may be interrupted by moderate to heavy cloud cover, obscuring the target.

Fifty-four different scenes were selected as test cases from the UAV tapes provided by AFRL. Each of the scenes captured at least one of the visual aspects described above (some captured several of the visual aspects). Table 3-1 summarizes the actual numbers of each of the different types of visual effects used as test cases in this research.

Table 3-1. Test Cases

| Visual Effect | Number of Test Cases |
|--|--|
| Abrupt Changes | 18 |
| Camera Zooms | 15 |
| Motion Across Scenery Changes | 54 - All the sequences contained motion of some type |
| Stationary Shots Separated by Rapid Motion | 14 |
| Cloud Cover | 4 |

3.3 Experimental Setup

After completion of the data acquisition process, segmentation is performed on the selected UAV video sequences using the edge detection algorithm, and the results are collected for analysis. This section provides a description of the hardware and software configuration of the experiment. Additionally, the procedure for testing the algorithm is described, along with a discussion of the method for analysis of the experimental results.

3.3.1 *Hardware and Software Configuration*

The source code for the edge detection algorithm was obtained via File Transfer Protocol (FTP) from [ZABIH97] and stored on the *Hawkeye* network in the Signal Information Processing Laboratory at AFIT. The code was subsequently compiled under the *Solaris* 2.5.1 operating system using the *gcc* compiler, and executed on a *Sun UltraSparc* workstation.

3.3.2 *Experimental Process*

The executable program, *dissolvem*, takes as input a series of PGM frames, and provides as output a matrix of information regarding inter-frame differences. The first column of data from the matrix output of *dissolvem* contains the edge change fraction (the measure of dissimilarity between successive frames). Peaks in the edge change fraction are used to determine scene breaks. Consequently, the edge change fraction is collected to analyze the results of applying segmentation to UAV video data.

The experimental process consists of executing the *dissolvem* program on each of the selected scenes (a series of PGM frames) and collecting the edge change fraction. During the experiment, the frame rate of each scene is varied across the following frame rates: 30 frames/second, 10 frames/second, and 5 frames/second.

3.3.3 Analysis of Results

Once the experiment concluded, the data collected from *dissolvem* can be analyzed to evaluate the performance of the edge detection algorithm on UAV surveillance data. The analysis consists of the following major steps:

- Expected versus Actual Outcome - Certain video sequences are expected to produce specific results when segmentation is applied. For example, abrupt camera changes should produce a noticeable spike in the edge change fraction when compared to surrounding data points. Therefore, the results of the segmentation algorithm are analyzed to determine if actual results are similar (or dissimilar) to expected results.
- Comparison of Similar Sequences - Different UAV video sequences with similar visual effects (abrupt camera changes, zooms, etc.) should produce comparable results when applying the edge detection algorithm. The edge change fractions collected from UAV sequences are analyzed to ascertain if similar sequences produce like or dislike results.
- Analysis of scenes with motion - During a typical UAV flight, several hours of data will contain motion over gradual scenery changes (such as footage of city blocks, fields, etc.). The edge change fractions of

scenes with motion are collected and analyzed to provide insight into how the edge detection segmentation algorithm responds to motion.

- Explain Anomalies - The edge detection segmentation algorithm was originally designed to be used with man-made edited video footage. As a result, applying this segmentation method to UAV data may produce anomalous results. Consequently, anomalies will be discussed and explained (as possible).
- False Positives - As with any segmentation method, false positives will be common. False scene change identifications will be analyzed and discussed as appropriate.

3.4 Summary

The focus of this research is to determine if applying video segmentation to UAV video footage provides meaningful scene breaks for storage and retrieval purposes. The methodology of the research consists of three major steps. First, an applicable segmentation algorithm is selected from the literature. Based on experimentation performed in [DAILI95], and analysis of the characteristics of UAV data, the edge detection segmentation method proposed by [ZABIH97] was chosen. After the algorithm selection, UAV video footage is selected for experimental purposes. To include a wide variety of the typical visual effects a UAV may encounter, scenes with abrupt changes, camera zooms, slow motion, rapid motion, and cloud cover are chosen. Finally, the edge detection program

dissolven is executed on each of the scenes (varying the frame rates) and the edge change fraction is collected for subsequent analysis. The analysis consists of comparing the results of expected versus actual outcomes, similar sequences, and scenes with motion, along with explaining anomalies and false positives.

4 RESULTS

This chapter presents the analysis of applying the edge detection segmentation algorithm to the fifty-four scenes selected from the tapes provided by AFRL. The first section provides a discussion of the results of the frame rate variation. In the following sections, an analysis is provided on the results of the various categories described in Chapter 3, including abrupt changes, rapid motion, zooms, and cloud cover. In the final two sections, motion and false positives/anomalies are discussed, respectively.

4.1 Frame Rate Variation

As explained in Chapter 3, the *dissolvem* program was executed on each of the selected scenes, and the resulting edge change fractions were collected for analysis. As part of the experimental process described in section 3.3.2, the frame rate (in frames per second or fps) of each sequence was varied across the following rates: 30 fps, 10 fps, and 5 fps. The frame rate was varied during the experiment to determine if a reduced frame rate could allow the edge detection algorithm to detect meaningful scene breaks. A reduction in frame rate from 30 fps to 5 fps could reduce storage and communications requirements for UAV video footage by upwards of 20 percent.

The results of varying the frame rates are inconclusive. In several of the UAV sequences, varying the frame rate did not affect the scene breaks detected

using the edge change fraction. For example, consider the following abrupt scene change captured at 30 fps in Figure 4-1, and 5 fps in Figure 4-2. It is obvious from these graphs that the edge change fraction did not change noticeably when the frame rate was reduced.

However, not all scenes tested in this research behaved as predictably. Figure 4-3 and Figure 4-4 represent another abrupt scene change, captured at 30 fps and 5 fps, respectively. In Figure 4-3, there is an obvious spike in the data, representing the scene change. The graph depicted in Figure 4-4 is extremely choppy in nature, making it difficult to determine where the abrupt scene change occurs when the frame rate is reduced. This is due in large part to other characteristics of the scene, i.e. the fast motion of the objects in view, as though the camera is close to the ground. Consequently, no conclusive recommendation can be made regarding the predictable advantage of reduced frame rates. Additionally, the analysis presented in the remainder of this document will use the results from 30 fps scene analysis.

4.2 Abrupt Changes

Abrupt changes in UAV surveillance footage are equivalent to a cut in edited video such as feature films or TV shows. As such, detecting abrupt changes is essential in successfully partitioning UAV footage for database insertion and retrieval. There are several situations that cause an abrupt change to take place in UAV footage, namely switching the camera feed or turning the

Scene14 - Abrupt Change 30 FPS

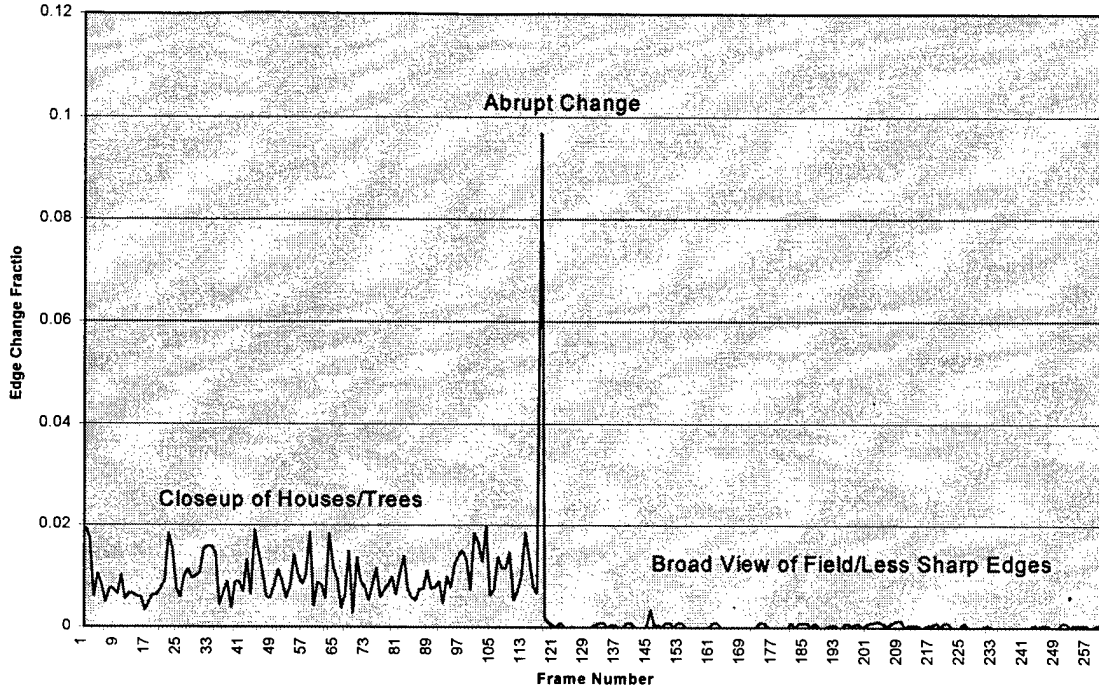


Figure 4-1. Abrupt Scene Change at 30 FPS

Scene14 - Abrupt Change 5 FPS

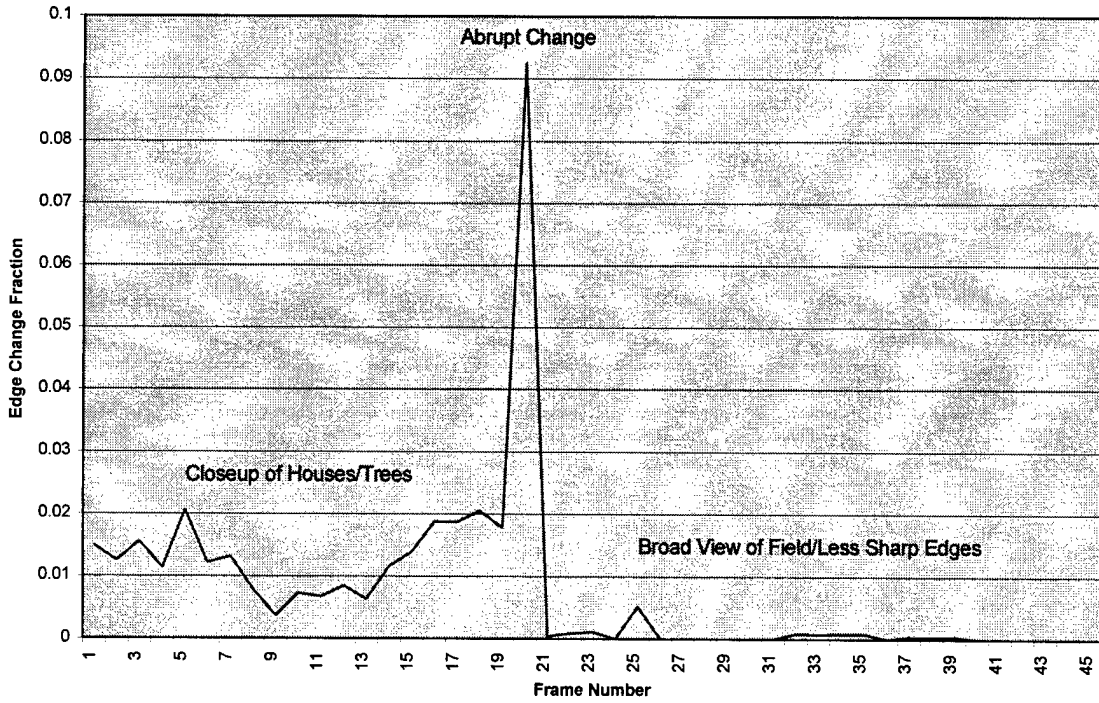


Figure 4-2. Abrupt Scene Change at 5 FPS

Scene44 - Abrupt Change 30 FPS

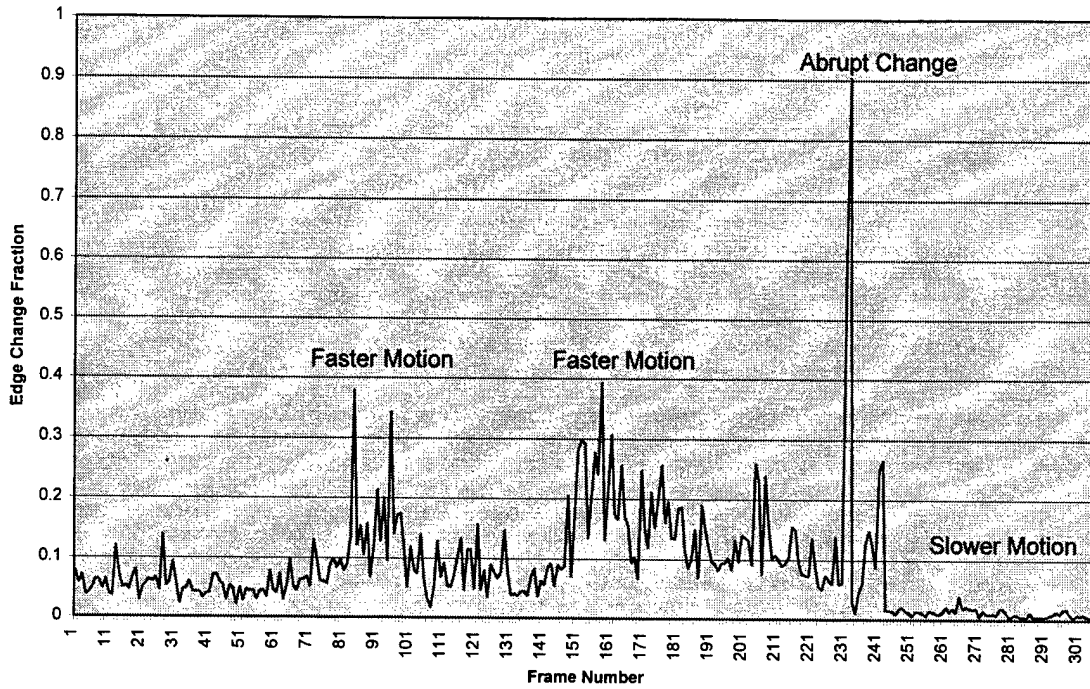


Figure 4-3. Abrupt Scene Change at 30 FPS

Scene44 - Abrupt Change 5 FPS

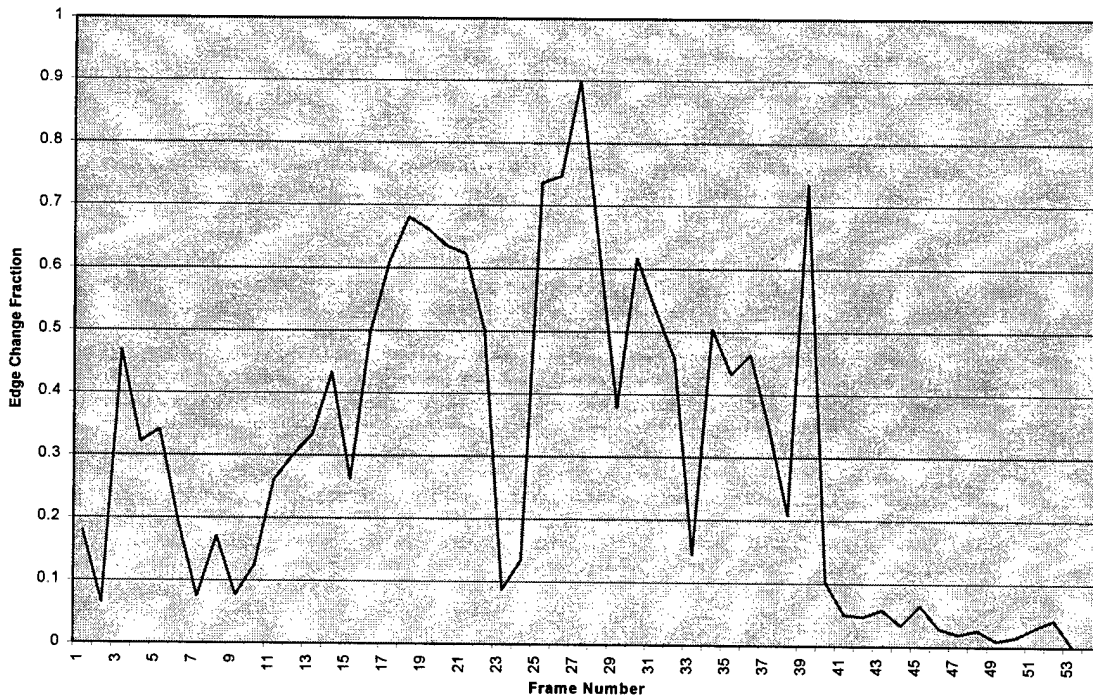


Figure 4-4. Abrupt Scene Change at 5 FPS

camera feed on/off. Eighteen scenes from typical UAV missions were selected that contained one of these elements. As described in section 3.3.2, the *dissolve* program was executed on each of the selected scenes and the edge change fraction was collected.

As expected, each of the eighteen abrupt change scenes behaved in a similar fashion. In each case, the abrupt camera change produced a noticeable spike (relative to the surrounding data points) in the edge change fraction over a small number of frames (usually 2 or 3 frames). Figure 4-5 depicts a typical abrupt scene change. In Figure 4-5, the edge change fraction spikes over a series of two frames to a value of 0.763. To detect scene changes, the authors of [ZABIH97] recommend setting the edge change fraction threshold at 0.15. Out of the eighteen scenes tested, a threshold of 0.15 would have successfully identified nine scenes. Although the other nine scenes produced a spike in relation to their surrounding points, the actual data values were less than the recommended threshold of 0.15. Figure 4-6 provides an example of an abrupt scene change with a spike under the recommended threshold.

There are several reasons an absolute threshold of 0.15 would not capture an abrupt change such as the example in Figure 4-6. First, the footage in Figure 4-6 contains telemetry data (such as latitude, longitude, elevation, etc.) overlaid on the video within boxes. These boxes are always present, and cover a large portion of the screen with sharp edges. As the edge change fraction is calculated

Scene53 - Abrupt Change 30 FPS

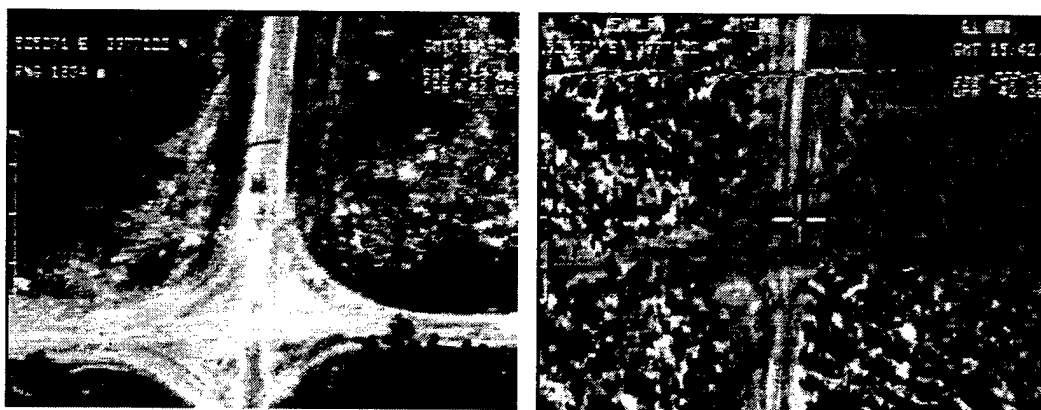
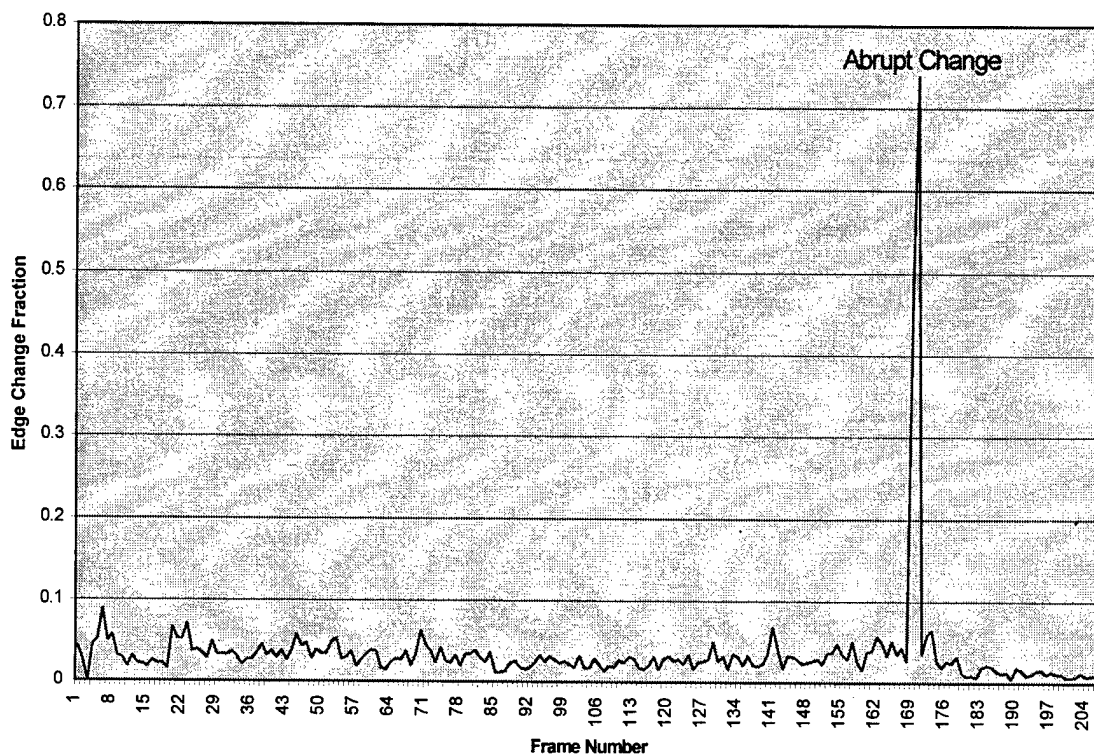


Figure 4-5. Typical Abrupt Change

from frame to frame, the sequences containing the boxed telemetry data will have less change due to the static boxes (as compared to the sequences without the boxed telemetry data). Additionally, certain sequences contain less complex data than other sequences (such as a view of a field versus a view of a city block):

Scene22 - Abrupt Change 30 FPS

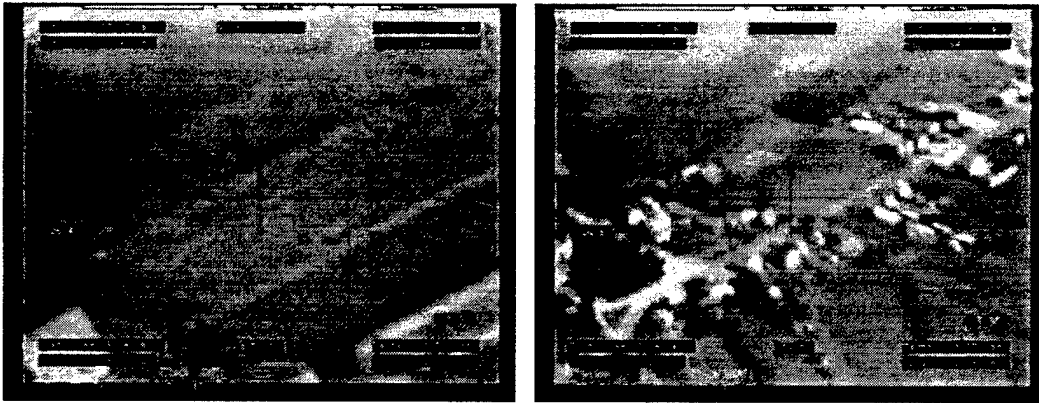
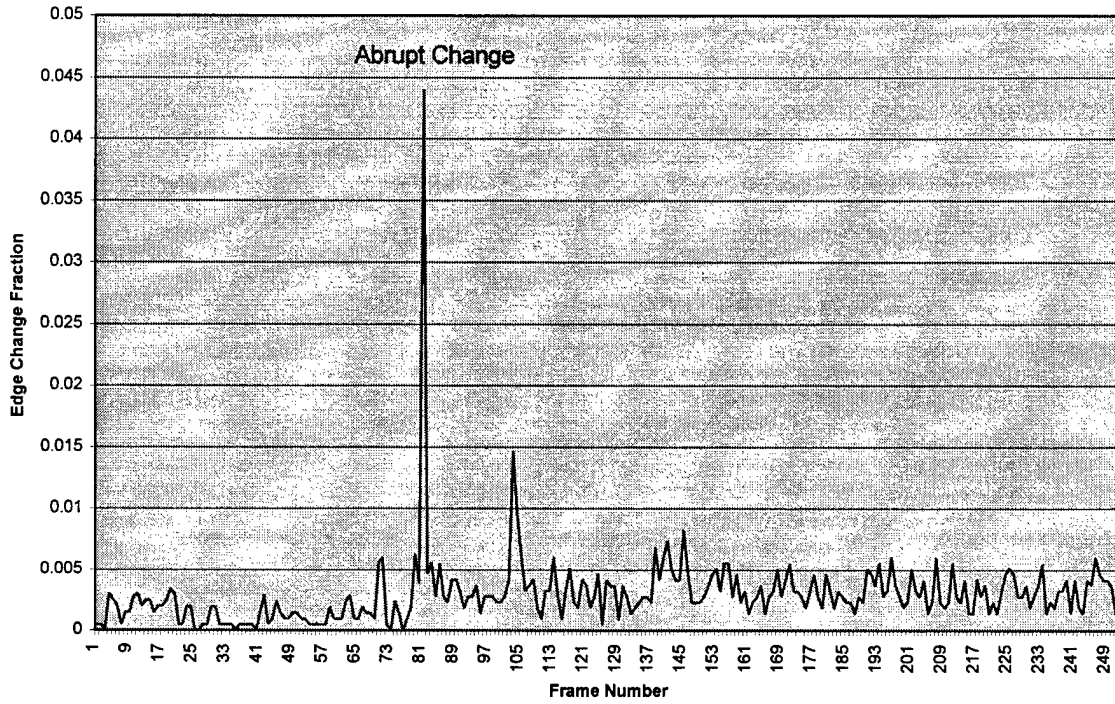


Figure 4-6. Abrupt Change with Low Threshold Value

When a scene change occurs within a sequence with few sharp edges, the edge change fraction will be noticeably lower than a scene change in a sequence with many sharp edges. For these reasons, a relative threshold (one comparing

surrounding data points) would provide more meaningful scene breaks than an absolute threshold.

4.3 Rapid Motion

Fourteen sequences were selected from the tapes provided by Air Force Research Laboratory's (AFRL) Information Intelligence Directorate containing rapid camera motion. Rapid motion usually occurs when one of the visible light cameras on the UAV is quickly rotated to acquire a target or area of interest. Consequently, the scenes occurring after rapid motion are of great interest to intelligence analysts. Per the methodology described in Chapter 3, *dissolvem* was executed on each of the sequences and the resulting edge change fractions were collected for analysis.

Each of the rapid motion segments within the fourteen scenes was detected as scene changes. The typical nature of a rapid motion scene change is a relatively large change in the edge change fraction over a number of frames (from as little as 20 to as many as several hundred, depending on the duration of the camera movement). The rapid motion can be categorized into two broad categories. The first category consists of choppy changes in the edge change fraction over the duration of the camera movement. This is typical when the camera movement is not uniform in nature. Rather, the camera may exhibit start/stop type motion, creating spikes in the edge change fraction interleaved

with frames containing little or no change. Figure 4-7 provides an example of a rapid motion scene change with choppy edge change fractions.

Almost uniform camera movement at an extremely high speed characterizes the second category of rapid motion. In contrast to the first category, this type of movement causes the resulting edge change fraction to exhibit a consistently high rate of change over the duration of the camera movement. Figure 4-8 provides an example of the second category of rapid motion scene change.

4.4 Zooms

In a typical UAV mission, once the target or area of interest is acquired, a zoom in may be performed to enhance the level of detail. Likewise, a zoom out can be performed to provide a broader view of an area. In each of the above cases, intelligence analysts would be interested in the scenes occurring after a zoom takes place (or in some cases, the scene before a zoom out takes place). For this reason, it is important to determine whether the edge detection segmentation method can detect zooms as scene changes.

Fifteen sequences were selected from the UAV footage provided by AFRL for experimental purposes. As with the other visual effects described earlier in this chapter, the *dissolve* program was executed on each of the scenes, and the edge change fraction was collected for analysis. The majority of the zoom sequences behaved favorably when segmented with the edge detection method.

Scene7 - Stationary Shots Separated by Rapid Motion 30 FPS

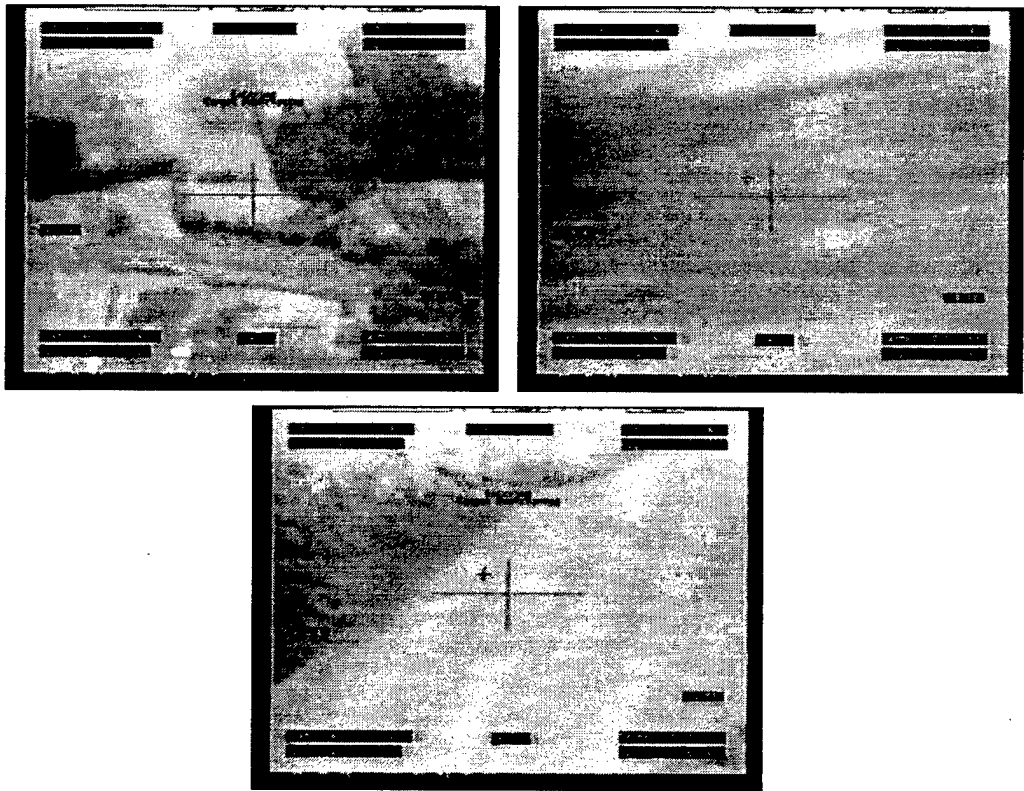
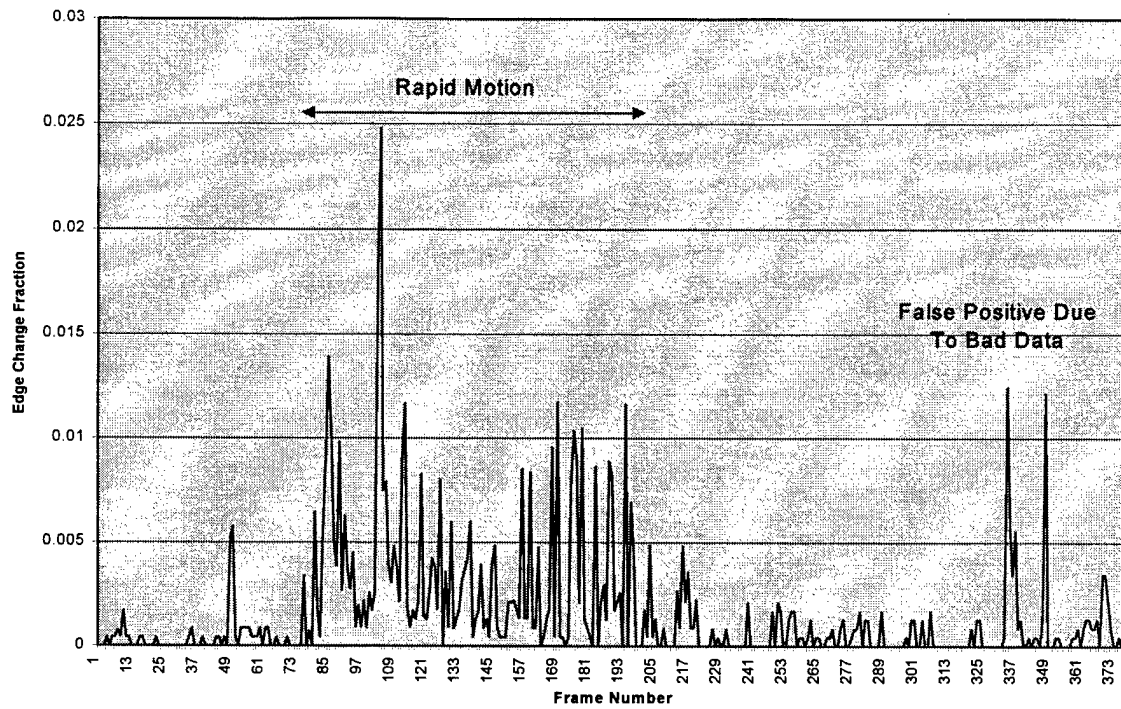


Figure 4-7. Rapid Motion Scene Change/Choppy Edge Change Fraction

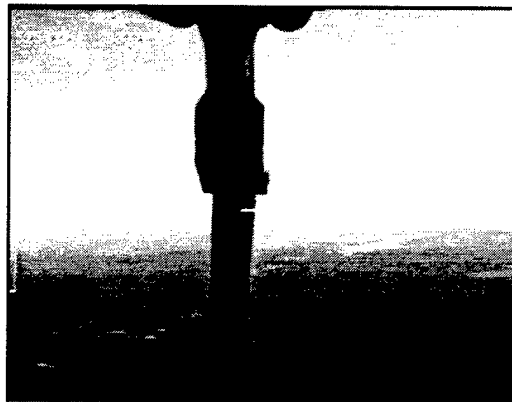
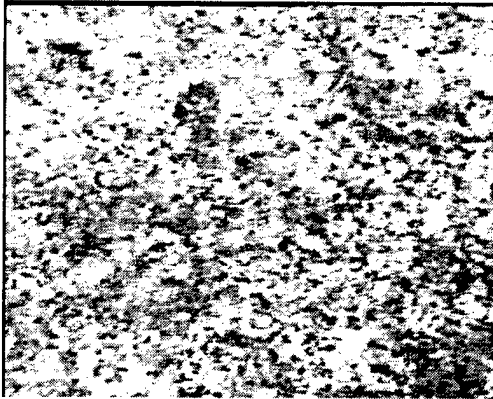
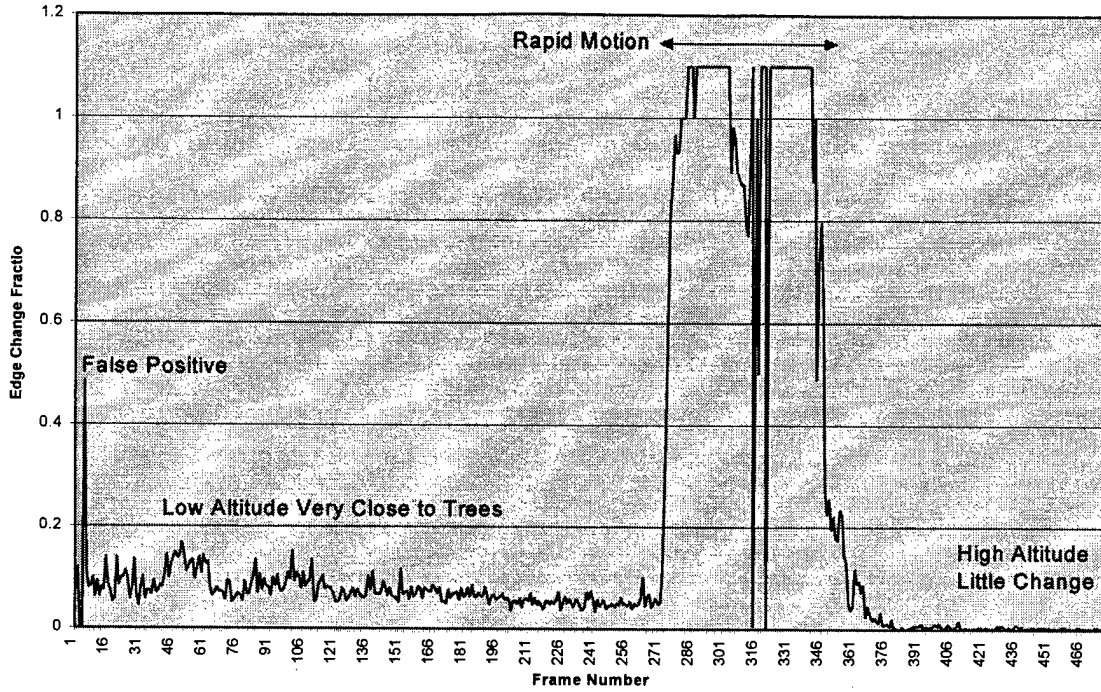


Figure 4-8. Rapid Motion/Uniformly High Edge Change Fraction

Out of the fifteen scenes, fourteen were identified as scene changes for a 93.3 percent correct identification rate.

The nature of a zoom is very similar to the first category of rapid motion when the resulting edge change fractions are analyzed. The typical zoom sequence consists of a moderate to large change in the edge change fraction over a number of frames (see Figure 4-9 for an example of a zoom in, and Figure 4-10 for an example of a zoom out). Like the rapid motion category, the number of frames can be from as little as twenty to as many as several hundred, depending on the duration of the camera action. Additionally, the edge change fractions do not change uniformly during the zooms. This is in large part due to non-uniform camera motion during the zoom. Consequently, the resulting edge change fractions are choppy in nature, exhibiting spikes interleaved with frames containing little or no change.

As discussed earlier, one of the fifteen scenes was not detected as a scene break when analyzing the resulting edge change fractions. The main reason for this is the nature of the camera movement of this particular sequence. In contrast to the other zooms tested in this research, this zoom was extremely slow in developing. For this reason, the edges changed slowly from frame to frame, causing very small changes in the edge change fraction. This small change mimics gradual motion. As a result, this sequence was not identified as a scene change. Using the current algorithm, detecting this event will be problematic.

Scene2 - Zoom In/Slow Motion Over Street/Buildings 30 FPS

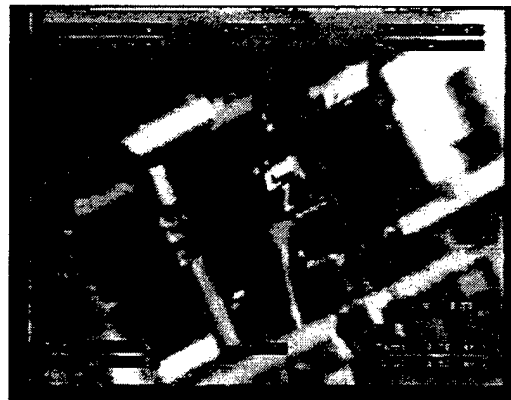
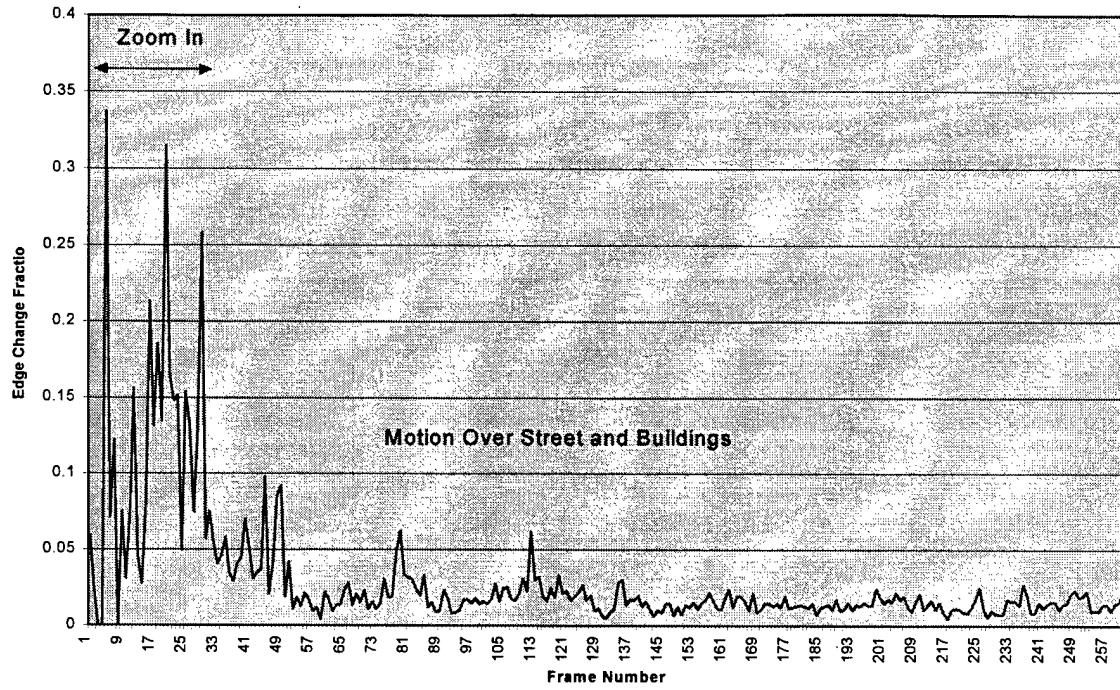


Figure 4-9. Zoom In Scene Change

Scene4 - Zoom Out 30 FPS

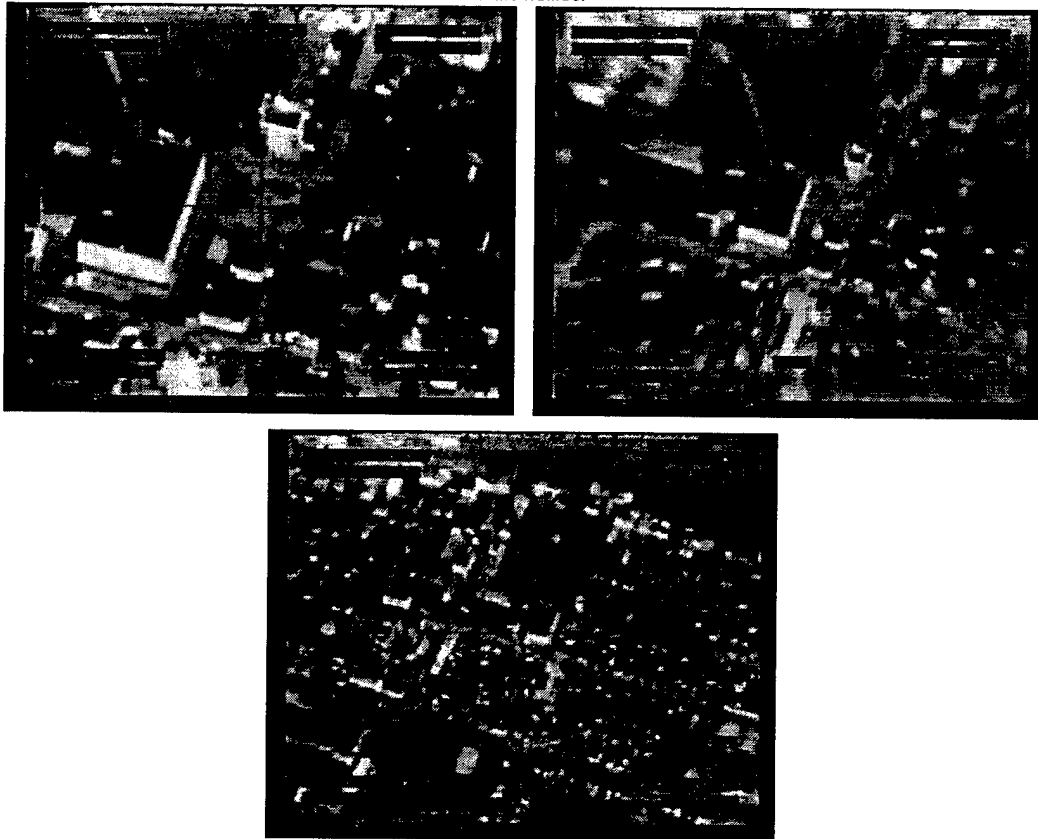
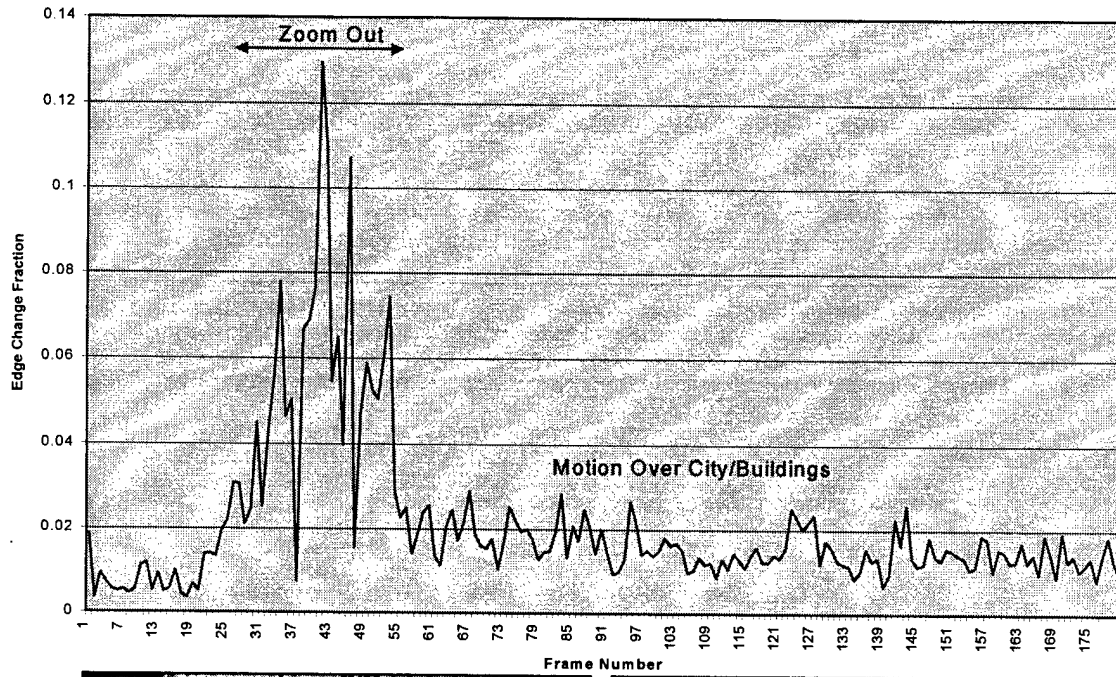


Figure 4-10. Zoom Out Scene Change

4.5 Cloud Cover

As with any aerial surveillance platform, cloud cover can be a frequent occurrence on UAV reconnaissance missions. In some cases, the cloud cover may be extremely light, causing no serious impediment to the UAV mission. On the other hand, sometimes cloud cover can be heavy in nature, partially or totally obscuring the target from view. If segmentation is to be used to partition video for database insertion, it must be determined how the edge detection segmentation method would react to sequences with cloud cover.

Consequently, cloud cover sequences were selected from the UAV tapes provided by Rome Laboratory. However, due to the limited sampling provided, only four sequences were identified as test cases for this research. The first of the four cloud sequences contained a light cloud passing through the target area during observation. Since the cloud was faint and the majority of the edges in the sequence changed relatively little, the cloud was not detected by segmentation (see Figure 4-11).

The three remaining cloud cover sequences all contained moderate to heavy cloud cover. In two of these sequences, the edge detection method appears to behave favorably to the clouds (for an example, see Figure 4-12). In both cases, as the cloud cover enters the sequence, large changes in the edge change fraction are produced and persist until the cloud exits (or allows some visibility of the target area). In these two cases, the edge change fraction appears

Scene6 - Stationary Shot/Cloud Passing in Front of Camera 30 FPS

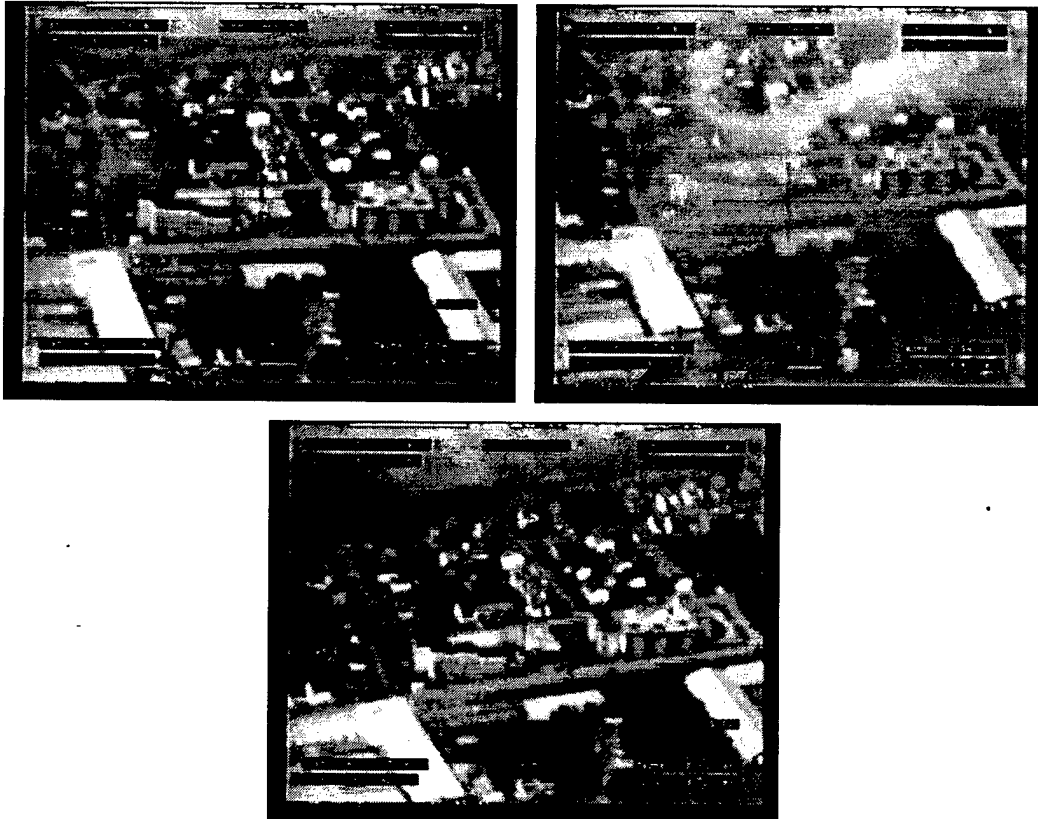
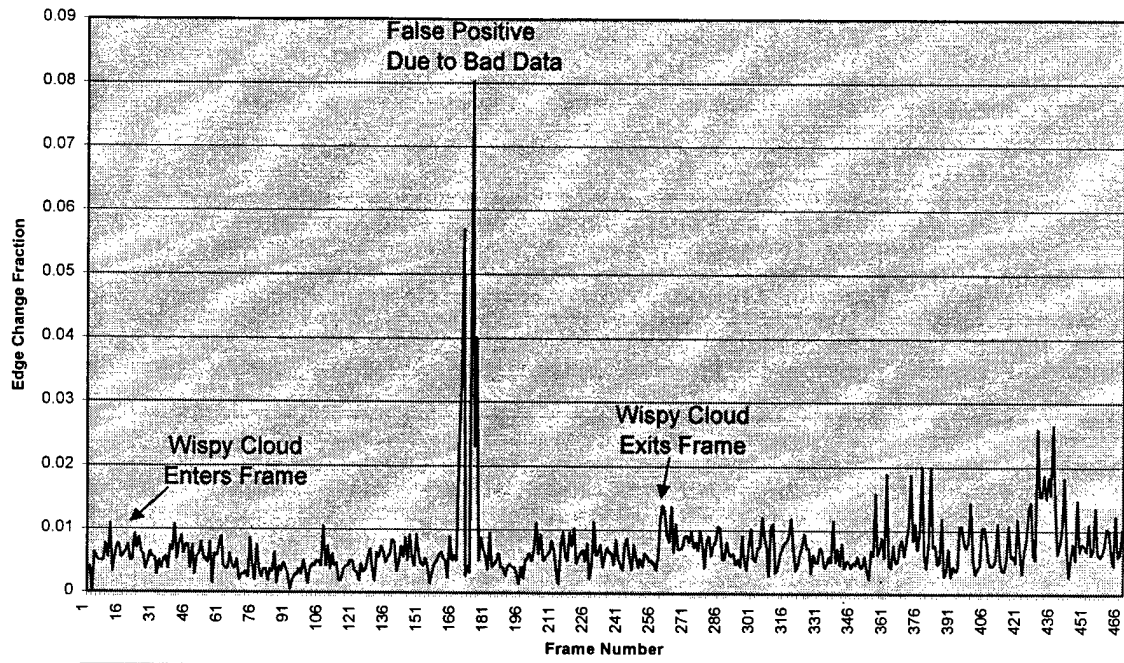


Figure 4-11. Light Cloud Cover

Scene29 - Cloud Cover 30 FPS

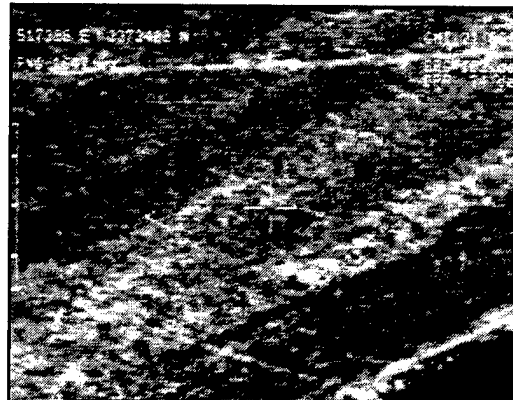
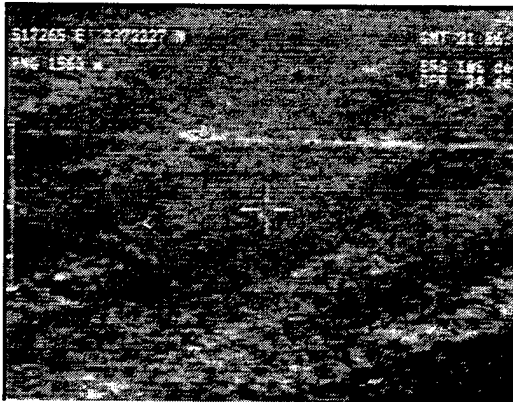
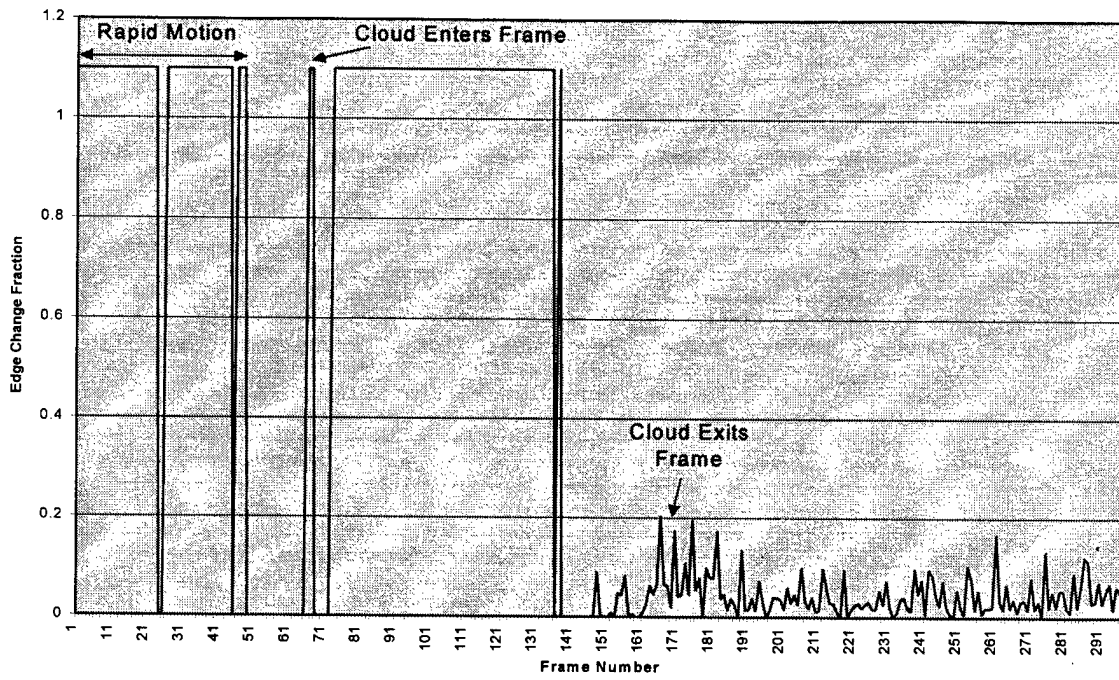


Figure 4-12. Heavy Cloud Cover

to react similar to that of a rapid motion sequence, with large changes in the edge change fraction over many frames (depending on the number of frames containing the cloud cover).

However, in the final heavy cloud cover sequence, the edge detection method behaves unfavorably. In this sequence, the edge change fraction changes radically for the duration of the sequence and no correlation with the results and the sequence can be made. Due to the anomalous results of the final heavy cloud cover sequence, and the limited number of samples tested, the results of this type of visual effect are inconclusive. Although segmentation appeared to behave favorably in several of the sequences, more research is required before a recommendation can be made.

4.6 Motion

The large majority of UAV reconnaissance missions require the vehicle to navigate across large segments of countryside and/or residential areas (sometimes including cities or small towns) while en route to the area of interest. As a result, large portions of the UAV surveillance footage (in some cases, several hours) may contain little or no useful information while the UAV is approaching its destination. Therefore, it is important to determine the reaction of segmentation to motion of various degrees over scenery. If the UAV reacts unfavorably to certain types of motion, it may cause many false positives, prohibiting the effective partitioning of the data.

As discussed earlier, fifty-four sequences were chosen for test cases in this research based on the various visual effects they contain. Each of these sequences contains motion of varying degrees. In accordance with the methodology described in section 3.3.3, the edge change fractions of the sequences were collected and analyzed. Based on that analysis, the motion of the UAV camera can be classified into two categories: *Slow/Gradual Motion at High Altitudes* and *Fast Motion at Low Altitudes*. These two categories are described in the following sections.

4.6.1 Slow/Gradual Motion at High Altitudes

In many UAV flights, the vehicle travels at a relatively slow speed and a high altitude. In this scenario, the camera motion tends to be gradual or slow in nature, causing very little change from frame to frame. Several sequences tested in this research contained this type of motion. In each case, the resulting edge change fractions behaved in similar fashion. As expected, the edge change fractions contained little or almost no change in sequences with gradual motion. This can be considered a favorable outcome, since it would be nonsensical to segment sequences based on gradual or slow motion. Figure 4-13 provides an example of gradual motion.

4.6.2 Fast Motion at Low Altitudes

In some instances, the UAV may travel at a low altitude to provide detail of a target or area of interest. Since the vehicle is traveling at a low altitude,

Scene5 - Slow Motion Over Road/Buildings/Plains 30 FPS

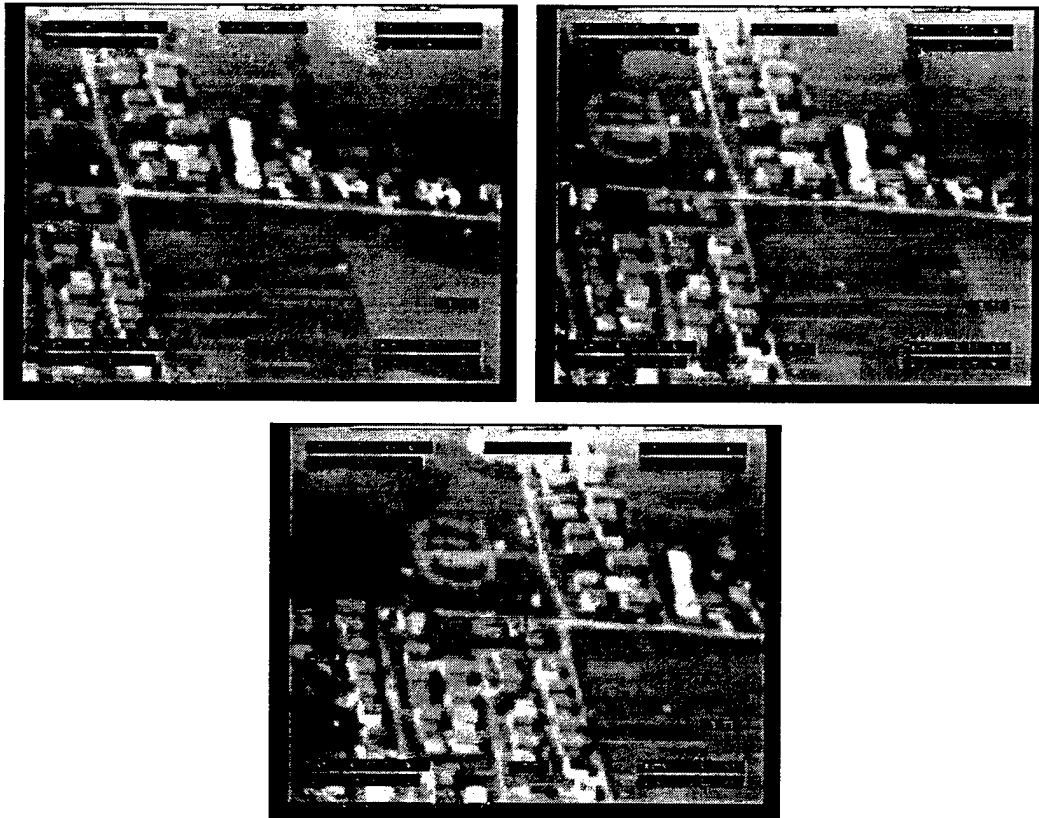
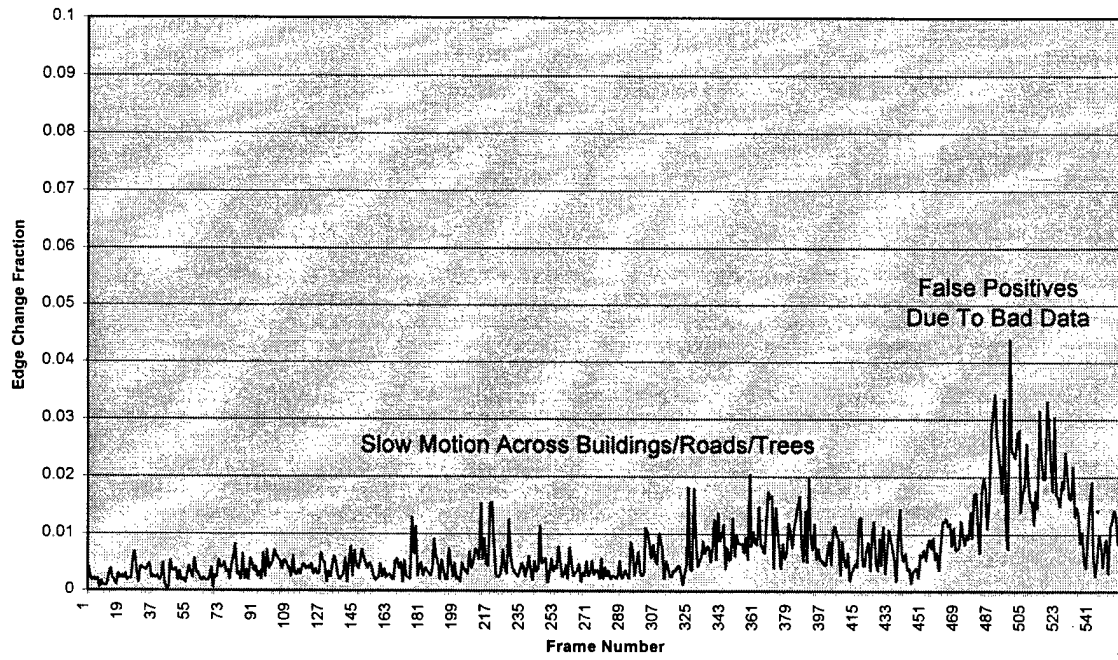


Figure 4-13. Gradual Motion

objects passing in front of the camera appear to be moving fast. Additionally, the visible light camera on the UAV may be zoomed in to enhance the level of detail of a specific area. When a camera is zoomed in, any type of camera movement causes the objects to move quickly. Motion in both of the above scenarios can be considered fast motion. Several sequences chosen for experimental purposes contained fast motion. Based on the results of the analysis of the resulting edge change fractions, fast motion appears to cause large changes in the edge change fractions (see Figure 4-14 for an example of fast motion). In some cases, the fast motion appears to have a similar effect to rapid motion. Since some UAV missions may encounter this type of visual effect with regularity, this can be considered an unfavorable outcome. Fast motion appears to cause false positives (this is further described under section 4.7.3), which would cause us to segment video data inefficiently.

4.7 False Positives and Anomalies

The edge detection segmentation algorithm proposed by [ZABIH97] was originally intended to be used with edited video footage, such as feature films or news broadcasts. Consequently, applying this algorithm to continuous surveillance footage produced some false positives and anomalous results. The following sections provide a description of these results.

Scene41 - Zoom In 30 FPS

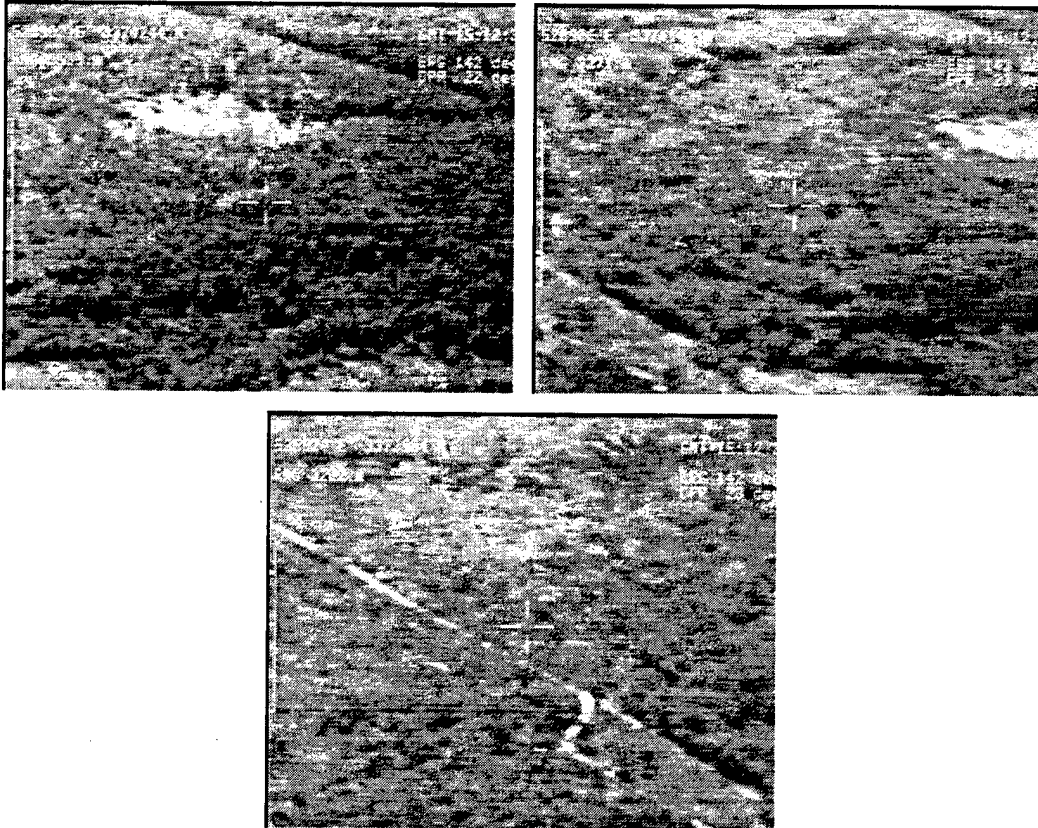
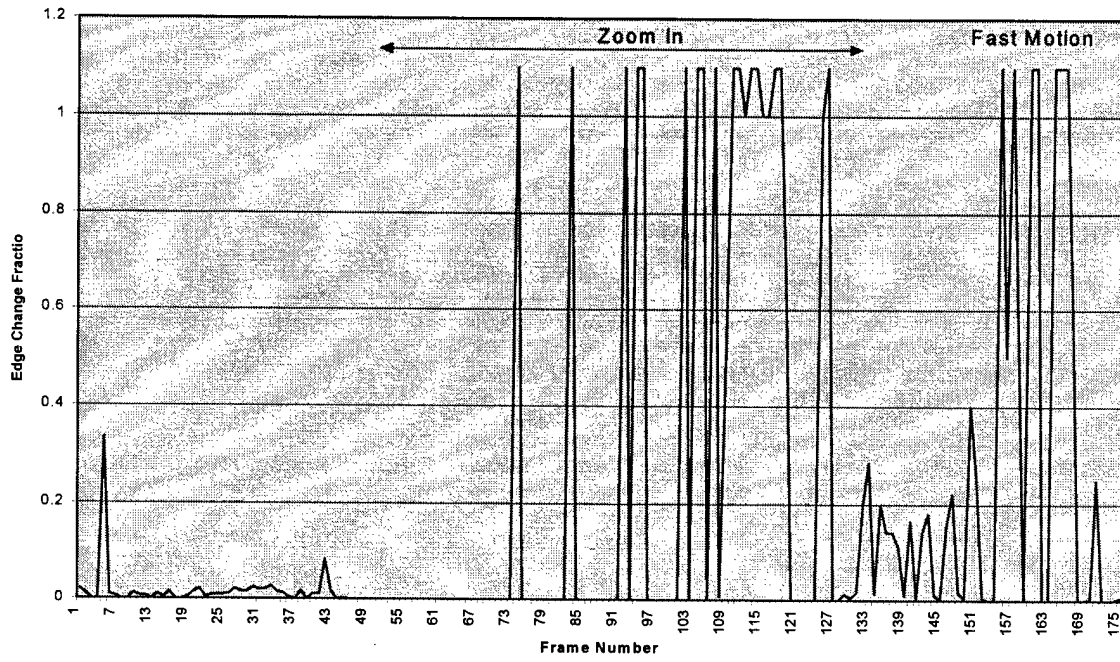


Figure 4-14. Fast Motion

4.7.1 Words/Telemetry Data

UAV footage contains reticulated telemetry data (such as latitude, longitude, elevation, etc.) overlaid on the video frames. In some instances, this data may disappear and reappear, or simply be updated on the frame. The appearance or disappearance of data, along with the updating of digits within that data, can cause a moderate change in the edges from frame to frame. Consequently, in many cases where the data appears or is updated, a noticeable spike is produced in the edge change fraction (see Figure 4-15 for an example). Based on these results, it appears as if the edge detection method reacts to entering and exiting text on the video frame as a scene change.

4.7.2 Corrupt Data

In many instances, a small imperfection in the analog videotape (such as a ripple, static, etc.) is transferred to the MPEG frames during the digitization process. In these cases, the imperfection may cause a dramatic change in the edges from frame to frame, even though no real change has occurred in the video sequence. As expected, the resulting edge change fractions contain a spike (in relation to the surrounding data points) where the corrupt data appears. Consequently, a false positive is produced in those places of a video sequence that contain corrupt data (see Figure 4-16).

Scene49 - Zoom In 30 FPS

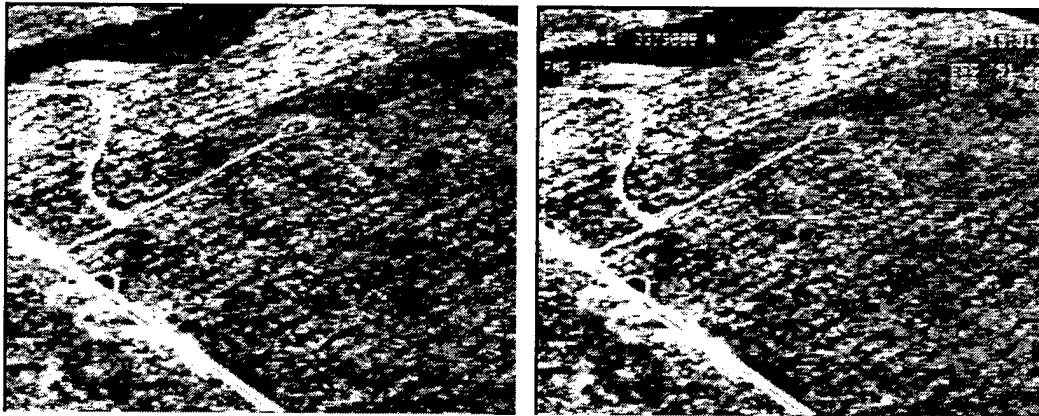
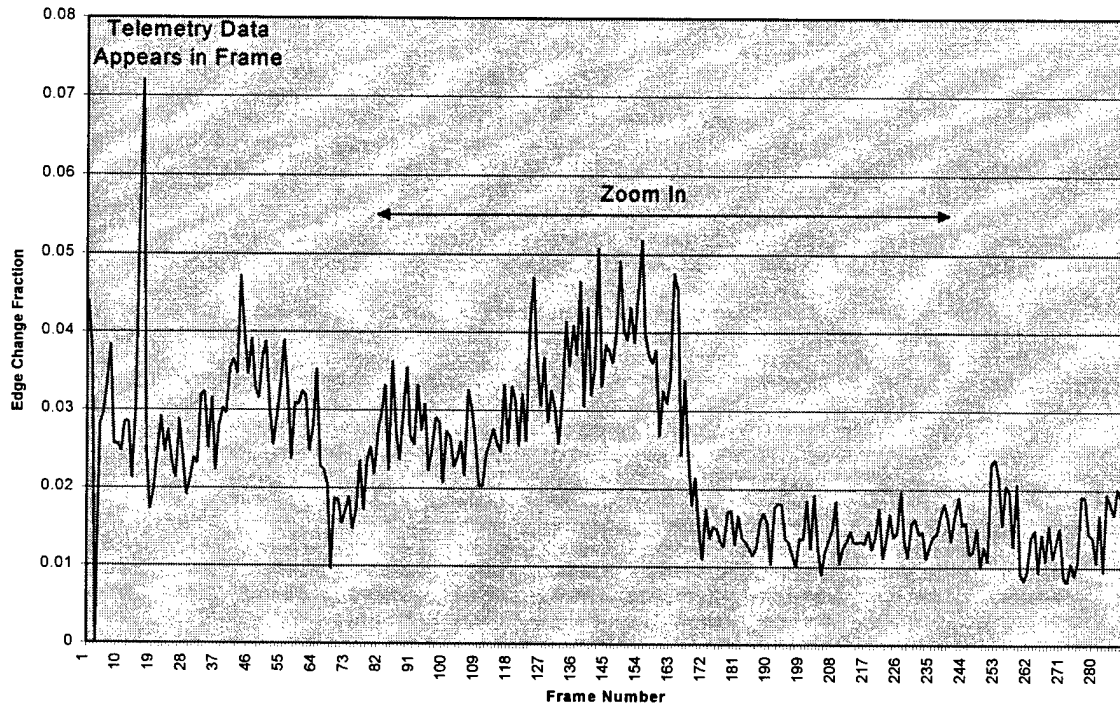


Figure 4-15. Telemetry Data Appears

4.7.3 Motion

As discussed in section 4.6.2, fast motion causes the edges to change in some cases dramatically from frame to frame. In the case of rapid camera

Scene32 - Corrupt Data 30 FPS

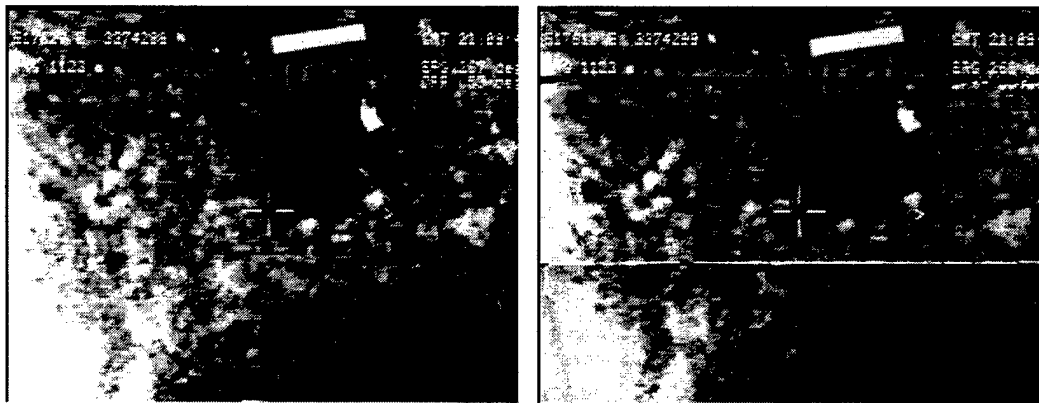
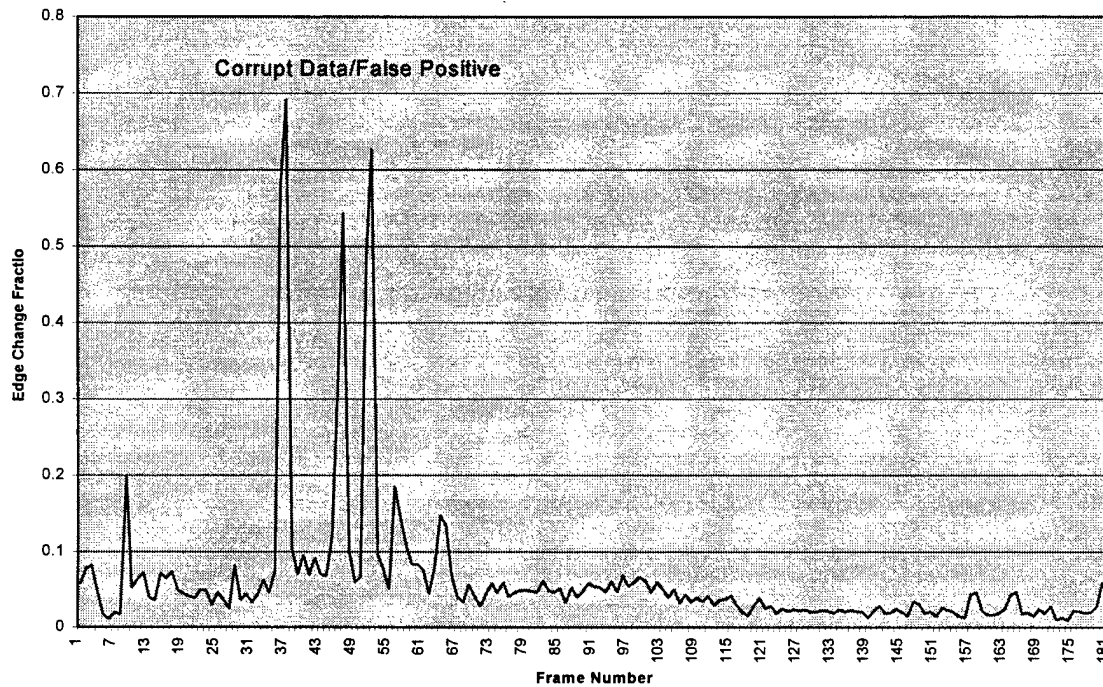


Figure 4-16. Corrupt Data

motion, where the camera is swiveled to acquire a target, acknowledging a scene change is warranted to capture the change in situation. However, when the UAV is simply flying at a low altitude over complex scenery (such as city buildings, trees, etc.), or zoomed in over a particular area, denoting a scene

change may not be warranted. Based on the analysis of the edge change fractions of scenes with motion, fast motion as described in section 4.6.2 causes moderate to large changes in the resulting edge change fractions. Since these changes may persist during the length of the fast motion, the false positives may cause the video data to be partitioned inefficiently.

4.8 Summary

This chapter presents the results of applying video segmentation to UAV video data. As described in Chapter 3, the edge detection segmentation algorithm was applied to fifty-four scenes selected from tapes provided by AFRL. The frame rate variation from the standard 30 fps to lower frames rates (10 fps and 5 fps) was inconclusive. Applying the edge detection segmentation algorithm to abrupt changes, rapid motion, and zooms produced favorable results, as these visual categories were all detected as scene changes. However, applying video segmentation to scenes with cloud cover is inconclusive. As expected, segmentation produced some anomalous results. These were due in large part to telemetry data, corrupt data, and camera motion. The impact of these results is explored in Chapter 5.

5 CONCLUSIONS AND RECOMMENDATIONS

As stated in Chapter 1, the focus of this research is to determine if the application of video segmentation to UAV video footage can provide meaningful segments for database storage and retrieval. To accomplish this, the edge detection segmentation algorithm proposed by [ZABIH97] was applied to fifty-four scenes containing various visual effects (abrupt changes, camera zooms, rapid and gradual motion, and cloud cover) per the methodology described in Chapter 3. Chapter 4 provides an analysis of the results of this experiment. Several conclusions can be drawn from this analysis, and can be grouped into three broad categories: near-term benefits, long-term benefits, and future research directions. The following sections present a discussion of each of these areas.

5.1 Near-Term Benefits

Many of the visual effects that can be considered scene changes take place in and around the target area on a typical UAV mission. While a UAV is en route to a particular target, the visible camera feed is for the most part static. That is, the camera does not typically perform zooms or camera movements until entering the area of interest. However, when a UAV enters an area of interest and the target is acquired, several actions are usually performed. These include camera zooming to enhance the level of detail, switching among the visible light cameras (causing an abrupt camera change), and swiveling a camera between

targets (stationary shots separated by rapid camera motion). As evident in the analysis presented in Chapter 4, the edge detection segmentation algorithm behaved favorably in this experiment when encountering abrupt changes, zooms, and stationary shots separated by rapid motion. In each of these cases, the visual effect was detected as a scene change by the edge detection algorithm.

There are several near-term benefits provided by detecting these visual effects as scene changes. The following sections provide a discussion of each of these benefits.

5.1.1 Key Frames for Mission Content

The first, and probably most substantial near-term benefit, is the ability to identify key frames from UAV mission data. Key frames can be extracted from UAV data by selecting the first frame from each new scene, as detected by video segmentation. Since the majority of new scenes are detected in and around the target area, segmentation can provide a good indicator of mission content. Building on the concept of key frames, a thumbnail sketch of the mission video can be constructed. Using some type of graphical user interface (GUI), such as a web page [PAGE99], intelligence analysts can browse an entire mission video in a matter of minutes (versus hours using the current method) by viewing only the key frames. Analysts could then select the key frames which require attention, and view only that portion of the mission video, saving image transmission and bandwidth time. Additionally, since segmentation typically produces false

positives instead of false negatives, the analyst can be assured that viewing only key frames has missed no critical scenes.

5.1.2 Video Partitioning for Imagery Mosaics

Another important near-term benefit is the ability to effectively partition UAV mission video for building imagery mosaics. Mosaics are still images of several (sometimes hundreds) of consecutive video frames that have been stitched together to provide a single panoramic image. In most cases it would be counterproductive to create mosaics of consecutive video frames under certain circumstances, such as camera zooms, abrupt changes, rapid motion, and other visual effects. However, a mosaic algorithm working with video segmentation could successfully create panoramic still images by creating mosaics *between* scene changes. When a visual effect is encountered by the segmentation algorithm and is identified as a scene change, the current mosaic process can be stopped and a still image created. Once the visual effect has passed, the mosaic process can begin again on a new image. Under this scenario, mosaics can be created automatically in conjunction with segmentation software. The mosaics can then be used in the same manner as key frames to provide an overview of mission content.

5.1.3 Segmentation Based on Telemetry Data

One unexpected result from this research is the ability of segmentation to detect entering and exiting telemetry data on video frames as scene changes.

Although at first this appears to be nothing more than a false positive, in some cases this could be of some importance to imagery analysts. Consider the situation where the text 'Entering Target Area' appears on a video frame, and causes a scene break to be detected by the segmentation software. In this scenario, a key frame could be created which points to the portion of the video containing the target. In conjunction with the thumbnail or mosaic process, this could be used in effect to fast forward to the segment of the video containing the target.

5.2 Long-Term Benefits

Each of the aforementioned near-term benefits could be implemented using existing computer software and hardware technology in a relatively short amount of time [PAGE99]. However, some of the benefits that segmentation provides will take a longer time to be realized, such as several months to several years. These are discussed in the following sections.

5.2.1 Key Frames for Indexing and Storage

The ultimate long-range goal of this research is to store UAV data in a video database management system (VDBMS), supporting all the typical functionality of an ordinary DBMS (indexing, querying, etc. to support as yet unanticipated UAV post-mission analysis requirements). However, as described in Chapter 2, video data has many differences when compared to traditional alphanumeric data stored in ordinary database systems. A major difference is

the temporal nature of video. To successfully store and retrieve video data in a VDBMS, one of the first steps that must be accomplished is the identification of significant video frames for representation, indexing, storage, and retrieval of the data. Significant video frames could be identified by automatic temporal sampling of the data, but since artificial intelligence and machine vision applications required for this are not yet mature or available, segmentation can be used as a "rough cut" to provide frames for indexing and storage. Using the concept of key frames described in the previous section, the first frame from each new scene detected can be used as a reference frame for indexing and storage/retrieval purposes. Although this is still a long way from a full-blown VDBMS, it provides the foundation for future research efforts.

5.2.2 Mission Profiles

Another important long-term benefit is the capability to provide mission profile information based on video segmentation characteristics. Consider a UAV mission that begins with a long flight over a body of water, with very little change in the visible light camera (see Figure 5-1 for an example edge change fraction plot). Using the edge detection segmentation algorithm, little change would be noticed. As the UAV crosses the shoreline and encounters trees and manmade structures, the video frames become more complex and edge detection algorithm registers some change (but usually not enough to trigger a scene change). In this time, some rapid camera motion, camera zooms, and abrupt

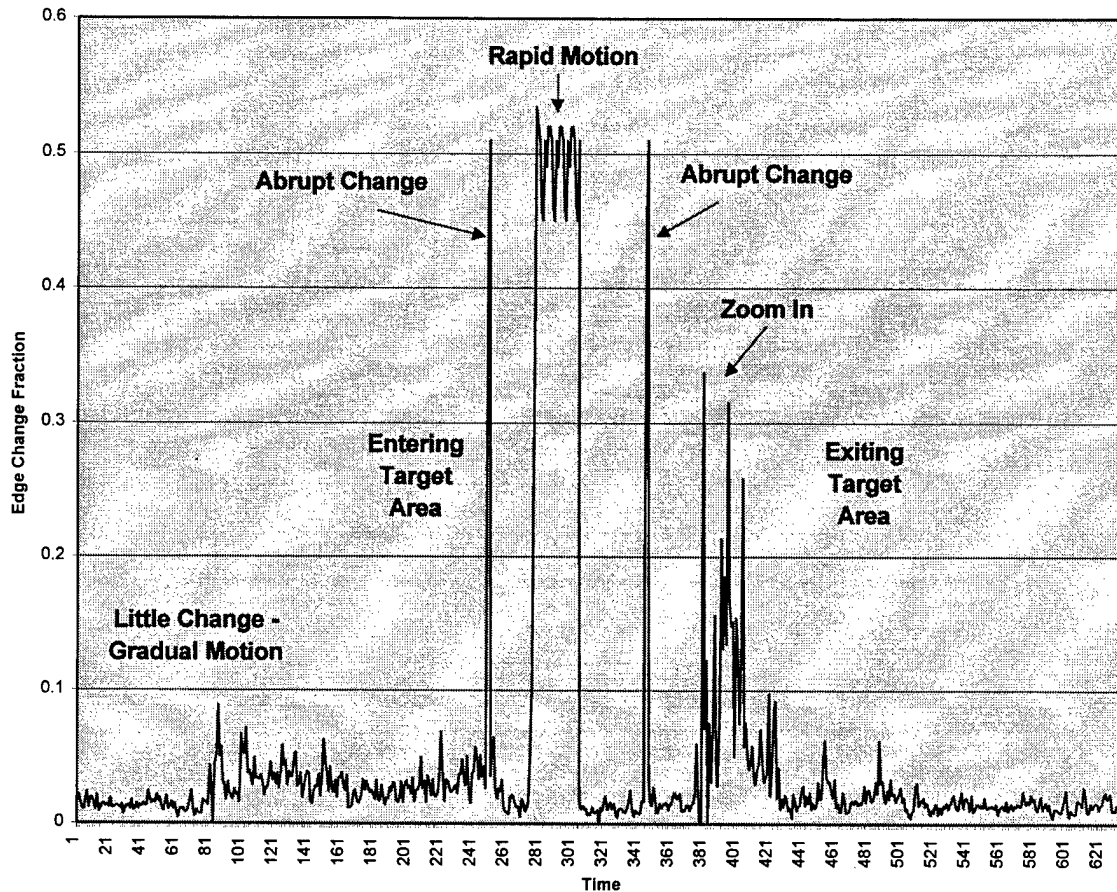


Figure 5-1. Example Mission Profile Plot Based On Edge Change Fractions

camera changes may occur causing a scene change, but usually these actions do not occur until the target area is entered. Once the target area is entered, several of the visual effects that cause scene changes occur, usually in a relatively short span of time. Finally, the target area is exited, and the flight to back to base produces little or no scene changes. Given this scenario, a profile of the mission can be generated by examining where scene changes occur in the mission timeline and what types of scene changes occur. Combining this information with other data sources, such as telemetry data (latitude, longitude, elevation,

etc.), mission dates and times, and other pertinent information, could eventually allow imagery analysts to perform semantic-based queries on these attributes. For example, an analyst may want to see all video clips on 2 Feb 99 containing a camera zoom in the area of latitude N 34 and longitude W 118.

Building on the idea of key frames and segmentation characteristics, these long-term benefits can be realized with some additional research and advances in computer software and hardware. The next section recommends some future research directions that can ultimately lead to the fulfillment of both the near-term and long-term benefits.

5.3 Future Research Directions

As described above, there are several near-term and long-term benefits to be realized from applying video segmentation to UAV video data. However, before these benefits can be of any practical use, some additional research must be accomplished. The first recommended course of follow-on research is to explore other candidate segmentation algorithms. As described in Chapter 2, there are many other algorithms in the literature, each of which exploits some characteristic of video data (such as color histograms, motion vectors, statistical analysis, etc.) to determine scene changes. Since many of these algorithms exploit different characteristics, they behave differently than the edge detection method under similar circumstances. In some situations, they may detect changes that the edge detection method did not, or vice versa. The next logical

step would involve layering segmentation data from several algorithms to provide the best or most probable scene changes. Additionally, other algorithms may behave favorably where the edge detection experienced problems (i.e., fast motion close to the ground). Consequently, a tool that would allow an analyst the capability to segment with different algorithms based on mission information (elevation, average speed, terrain encountered, etc.) could be of great benefit. For example, imagine a software package with two segmentation tools: tool 1 that works well with rapid motion and tool 2 that works well with abrupt changes but poorly with rapid motion. If an intelligence analyst knew the UAV mission was flown at high speeds close to the ground, tool 1 could be selected for segmentation.

Another area of recommended research involves segmentation characteristics or attributes. A major component of a DBMS is indexes, therefore, building indexes based on segmentation attributes should be explored. Since the possibility exists that imagery analysts may want to query UAV data based on visual effects (zooms, abrupt changes, rapid motion, etc.), indexes could be built based on the type of visual effect causing the scene change of the key frame in question. Also, other indexable attributes, such as telemetry data, date/time stamps, object recognition technologies, etc., could be combined to provide a layered indexing scheme, allowing analysts to query on each attribute individually, or combining attributes to provide more complex queries.

Following this path, the next areas to explore would include query processing and database retrieval.

The final recommended research direction includes segmentation based on audio cues. In current and future UAV missions, the UAV operator can record audio information on the video. Terms such as 'Entering Target Area' or 'Target Acquired' can be used to identify key frames for storage and retrieval purposes. In this scenario, the audio cues would serve as *synthetic* scene breaks, or scene breaks inserted into the UAV video by the operator. This allows for more effective control of where scene breaks occur, and all but eliminates false positives. Additionally, the audio segmentation can be layered with visual segmentation and the other attributes available to provide a robust query and retrieval capability for imagery analysts.

5.4 Summary

Based on the analysis performed in Chapter 4, the edge detection segmentation algorithm behaved favorably by detecting scene changes when encountering abrupt changes, zooms, and stationary shots separated by rapid motion. These results produce several near-term benefits, including the capability to identify key frames from UAV mission data, effectively partition UAV mission video for building imagery mosaics, and the ability to detect entering and exiting telemetry data on video frames as scene changes. Along with the near-term benefits, several long-term benefits are also produced, such as

the ability to provide key frames for indexing/storage and mission profile information based on video segmentation characteristics. To realize these benefits, several future research directions are recommended. These include the exploration of other candidate segmentation algorithms, using segmentation attributes as database indexes, and audio segmentation.

APPENDIX A - DATA AND SOFTWARE

AVAILABILITY

The data and software used in this research is available by contacting the AFIT School of Engineering Database Systems Research Point of Contact (POC).

Currently, the Database Research POC is:

Maj Michael L. Talbert
Air Force Institute of Technology
WPAFB, OH 45433-7765

Email: michael.talbert@afit.af.mil
Phone: DSN 785-6565 ext. 4280 COMM (937) 255-6565

BIBLIOGRAPHY

- [BOUTH97] P. Bouthemy, M. Gelgon, and F. Ganansia, "A Unified Approach to Shot Change Detection and Camera Motion Characterization," Publication Interne No. 1148, Institut De Recherche En Informatique Et Systemes Aleatoires, Nov. 1997.
- [CHRIS95] Stavros Christodoulakis and Leonidas Koveos, "Multimedia Information Systems: Issues and Approaches," *Modern Database Systems*, (New York: Addison-Wesley, 1995) 318-337.
- [CRIST96] John Cristy, *ImageMagick Image Manipulation Software*, <http://www.wizards.dupont.com/cristy/ImageMagick.html>, 1996.
- [DAILI95] Apostolos Dailianas, "Comparison of Automatic Video Segmentation Algorithms," *SPIE*, Oct. 1995, Vol. 2615: 2-16.
- [DAZZL97] DAZZLE™ Multimedia, SNAZZI Video Capture Software, <http://www.dazzlemm.com.sg/html/snazzi.html>, 1997.
- [DIMIT97] Nevenka Dimitrova and Forouzan Golshani, "Video and Image Content Representation and Retrieval," *The Handbook of Multimedia Information Management*, (New Jersey: Prentice Hall PTR, 1997) 95-138.
- [ELMAG97] Ahmed K. Elmagarmid, Haitao Jiang, Abdelsalam A. Helal, Anupam Joshi, and Magdy Ahmed, *Video Database Systems: Issues, Products, and Applications*, (Boston: Kluwer, 1997) 1-132.
- [FALOU96] Christos Faloutsos, *Searching Multimedia Databases by Content*, (Boston: Kluwer, 1996) 57-80.
- [GIBBS97] Simon Gibbs, Christian Breiteneder, and Dennis Tsichritzis, "Modeling Time-Based Media," *The Handbook of Multimedia Information Management*, (New Jersey: Prentice Hall PTR, 1997) 13-36.
- [GUPTA97] Amarnath Gupta and Ramesh Jain, "Visual Information Retrieval," *Communications of the ACM*, May 1997: 71-79.

- [HAMPA95] Arun Hampapur, *Design of Video Data Management Systems*, Ph.D. Thesis, The University of Michigan, 1995.
- [HIBIN96] Stacie Hibino and Elke A. Rundensteiner, "A Visual Multimedia Query Language for Temporal Analysis of Video Data," *Multimedia Database Systems: Design and Implementation Strategies*, (Boston: Kluwer, 1996) 123-159.
- [HJELS96] Rune Hjelsvold, Roger Midtstraum, and Olva Sandsta, "Searching and Browsing a Shared Video Database," *Multimedia Database Systems: Design and Implementation Strategies*, (Boston: Kluwer, 1996) 89-122.
- [JAGAD97] H.V. Jagadish, "Content-Based Indexing and Retrieval," *The Handbook of Multimedia Information Management*, (New Jersey: Prentice Hall PTR, 1997) 69-92.
- [KAYLE95] David C. Kay and John R. Levine, *Graphics File Formats*, (New York: McGraw Hill, 1995).
- [LEEIP95] John Chung-Mong Lee and Dixon Man-Ching Ip, "A Robust Approach for Camera Break Detection in Color Video Sequence," Technical Report HKUST-CS95-14, The Hong Kong University of Science and Technology, Department of Computer Science, Apr. 1995.
- [LIENH97] Rainer Lienhart, Silvia Pfeiffer, and Wolfgang Effelsberg, "Video Abstracting," *Communications of the ACM*, Dec. 1997: 55-62.
- [MENGJ95] Jianhao Meng, Yujen Juan, Shih-Fu Chang, "Scene Change Detection in a MPEG Compressed Video Sequence," Columbia University, Department of Electrical Engineering and Center for Telecommunications Research, 1995.
- [MPEGDC98] MPEG Developing Classes ©, *mdcdecoder*, <http://vision.cs.wayne.edu/mpeg/index.html>, Vision and Neural Networks Laboratory, Computer Science Department, Wayne State University, 1998.
- [MSSG96] MPEG Software Simulation Group (MSSG), MPEG-2 Video Codec, *MPEG2Decode*, <http://www.mpeg.org/MPEG/MSSG>, 1996.

- [OOMOT97] Eitetsu Oomoto and Katsumi Tanaka, "Video Database Systems—Recent Trends in Research and Development Activities," *The Handbook of Multimedia Information Management*, (New Jersey: Prentice Hall PTR, 1997) 405-448.
- [PAGE99] Timothy I. Page, *Incorporating Scene Mosaics as Visual Indexes Into UAV Video Imagery Databases*, M.S. Thesis, AFIT/ENG/GCS/99M-16, Air Force Institute of Technology, 1999.
- [PATEL97] Nilesh V. Patel and Ishwar Sethi, "Video Segmentation for Video Data Management," *The Handbook of Multimedia Information Management*, (New Jersey: Prentice Hall PTR, 1997) 139-165.
- [PICAR95] Rosalind W. Picard, "Light-years from Lena: Video and Image Libraries of the Future," Technical Report No. 339, Massachusetts Institute of Technology, Media Laboratory, Perceptual Computing Section, Oct. 1995.
- [PRABH97] B. Prabhakaran, *Multimedia Database Management Systems*, (Boston: Kluwer, 1997) 1-112.
- [SETHI95] Ishwar K. Sethi and Nilesh Patel, "A Statistical Approach to Scene Change Detection," *SPIE*, Feb. 1995, Vol. 2420: 329-338.
- [SIMON96] Brigitte Simonnot and Malika Smail, "Model for Interactive Retrieval of Videos and Still Images," *Multimedia Database Systems: Design and Implementation Strategies*, (Boston: Kluwer, 1996) 278-297.
- [VASCO97] Nuno Vasconcelos and Andrew Lippman, "A Bayesian Video Modeling Framework for Shot Segmentation and Content Characterization," Massachusetts Institute of Technology, Media Laboratory, 1997.
- [WANGA94] John Y. A. Wang and Edward H. Adelson, "Spatio-Temporal Segmentation of Video Data," *SPIE*, Feb. 1994, Vol. 2182.
- [WIEDE97] Peter H. Wiedemann, "On the Use of the Predator (MAE-UAV) System in Bosnia," Unmanned Aerial Vehicle Conference, Paris, France, June 1997.
- [XIOLE95] Wei Xiong, John Chung-Mong Lee, and Ding-Gang Shen, "Net Comparison: An Adaptive and Effective Method for Scene Change Detection," Technical Report HKUST-CS95-12, The Hong Kong

University of Science and Technology, Department of Computer Science, Apr. 1995.

- [XIOIP95] Wei Xiong, John Chung-Mong Lee, and Man-Ching Ip, "Net Comparison: A Fast and Effective Method for Classifying Image Sequences," *SPIE*, Feb. 1995, Vol. 2420: 318-328.
- [YEOLI95] Boon-Lock Yeo and Bede Liu, "A Unified Approach to Temporal Segmentation of Motion JPEG and MPEG Compressed Video," Princeton University, Department of Electrical Engineering, May 1995.
- [YEOYE97] Boon-Lock Yeo and Minerva M. Yeung, "Retrieving and Visualizing Video," *Communications of the ACM*, Dec. 1997: 43-52.
- [ZABIH97] Ramin Zabih, Justin Miller, and Kevin Mai, "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks," <http://www.cs.cornell.edu/Info/Projects/csrol/dissolve.html>, Cornell University, Computer Science Department, 1997.

VITA

First Lieutenant Bradley L. Pyburn is a prior-enlisted Air Force officer from Nashville, TN. He attended John Overton High School, graduating in the Top Ten in May 1990. Lt Pyburn received a scholarship to David Lipscomb University in Nashville, which he attended until his enlistment in the Air Force in August 1991.

Upon graduation from Basic Training at Lackland AFB, TX, in September 1991, he attended Communications-Computer Technical Training at Keesler AFB, MS. After graduation in December 1991, he was assigned to the Environmental Technical Applications Center (ETAC) at Scott AFB, IL. At ETAC he served as an Applications Programmer/Analyst in the Systems Branch, where he worked several key projects, one of which provided worldwide dial-in connectivity to ETAC's climatological databases. He was selected Airman of the Quarter for July-September 1992, and nominated Airman of the Year for 1993.

While serving at ETAC, Lt Pyburn attended McKendree College in Lebanon, IL. He graduated Magna cum Laude with a BS in Computing and Information Science/Mathematics in May 1994. After graduation, Lt Pyburn was accepted to Officer's Training School at Maxwell AFB, AL, where he graduated as a Distinguished Graduate in March 1995. Upon graduation, Lt Pyburn was assigned to the 480 Intelligence Group/27 Intelligence Squadron at Langley AFB, VA. From April-August 1995 he attended Basic Computer Officer Training at Keesler AFB, MS, where he graduated as a Distinguished Graduate. At the 480IG/27IS Lt Pyburn served as the Chief of the Network/Communications Management Branch, where he was responsible for over 14 million dollars in computer assets. He was selected Company Grade Officer of the Quarter for October-December 1996 and Company Grade Officer of the Year for 1996.

Lt Pyburn is currently stationed at Wright-Patterson AFB, OH, where he is attending the Air Force Institute of Technology working on his graduate degree in Computer Systems. Lt Pyburn's military awards include the National Defense Service Medal, Air Force Good Conduct Medal, Air Force Achievement Medal, and Air Force Commendation Medal. Lt Pyburn is also a member of the Ohio Eta Chapter of Tau Beta Pi. After graduation in March 1999, Lt Pyburn has been selected to teach Computer Science at the US Air Force Academy in Colorado Springs, CO. Lt Pyburn is married to the former Jacquelyn F. Wheeler of Nashville, TN, and they have three children: Taylor, Elissa, and Kirsten.

Permanent Address: 6518 Redmond Lane
Nashville, TN 37211