Content-based video classification and comparison

John Alexander Bunch
University of Nevada, Las Vegas

Follow this and additional works at: https://digitalscholarship.unlv.edu/rtds

Repository Citation
https://digitalscholarship.unlv.edu/rtds/2153

This Thesis is brought to you for free and open access by Digital Scholarship@UNLV. It has been accepted for inclusion in UNLV Retrospective Theses & Dissertations by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.
CONTENT-BASED VIDEO CLASSIFICATION
AND COMPARISON

by

John Alexander Bunch

Bachelor of Science in Computer Science
Georgia Institute of Technology
Atlanta, Georgia
2004

A thesis submitted in partial fulfillment of the requirements for the

Master of Science Degree in Computer Science
School of Computer Science
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
August 2007
INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.
Thesis Approval
The Graduate College
University of Nevada, Las Vegas

JULY 26TH, 2007

The Thesis prepared by

JOHN ALEXANDER BUNCH

Entitled

CONTENT-BASED VIDEO CLASSIFICATION AND COMPARISON

is approved in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Examination Committee Chair

Dean of the Graduate College

Examination Committee Member

Examination Committee Member

Graduate College Faculty Representative

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
ABSTRACT

Content-Based Video Classification and Comparison

John Alexander Bunch

Dr. Evangelos A. Yfantis, Examination Committee Chair
Professor of Computer Science
University of Las Vegas

Automatic video analysis tools have dramatically increased in importance as the internet video revolution has blossomed. This thesis presents an approach for automatic comparison of videos based on the inherent content. Also, an approach for creating groups (or clusters) of similar videos from a large video database is given.

First, methods simplifying and summarizing the content of videos will be presented. Such methods include shot boundary detection and key frame feature extraction.

Next, a comparison of different distance measures between videos will be given. These distance measures will be used to construct video clusters, and results will be compared.
TABLE OF CONTENTS

ABSTRACT....................................................................................................................... iii

LIST OF FIGURES........................................................................................................ vi

LIST OF ALGORITHMS............................................................................................... vii

ACKNOWLEDGEMENTS............................................................................................... viii

CHAPTER 1  INTRODUCTION ............................................................................ 1

CHAPTER 2  PREVIOUS RESEARCH .............................................................. 5

CHAPTER 3  PROJECT DESCRIPTION............................................................... 11

CHAPTER 4  SHOT BOUNDARY DETECTION ............................................. 12
  4.1 Transition Types .............................................................................................. 12
  4.2 Feature Buffer .................................................................................................. 13
  4.3 Feature Extraction .......................................................................................... 13
    4.3.1 Color Based Features ...................................................................... 15
    4.3.2 Edge Based Features ....................................................................... 17
    4.3.2.1 Sobel Edge Detection Algorithm ...................................... 17
    4.3.3 Inter-Frame Features ...................................................................... 18
    4.3.3.1 Video Object Plane ......................................................... 19
  4.4 Detectors ........................................................................................................... 21
    4.4.1 Cut Detector .................................................................................... 21
    4.4.2 Dissolve Detector ............................................................................ 25
    4.4.3 Fade Out Detector .......................................................................... 26
    4.4.4 Fade In Detector ............................................................................. 27
  4.5 Verification and Collaboration ...................................................................... 28
  4.6 Key Frame Selection ...................................................................................... 29

CHAPTER 5  KEY FRAME DISTANCE CALCULATION .................................... 30
  5.1 Euclidian Distance Measure ........................................................................ 30
  5.2 Squared Normalized Distance Measure ...................................................... 32
  5.3 RGB Color Histogram Intersection Distance Measure .......................... 33
LIST OF FIGURES

Figure 1  Modular Software Design .........................................................11
Figure 2  Basic Feature Summation Method Graph.................................24
Figure 3  Adjusted Z-Score Feature Summation Method ..........................24
Figure 4  Dissolve Transition IQR Feature Graph ...................................25
Figure 5  Fade Out Transition IQR Feature Graph ..................................27
Figure 6  Fade In Transition IQR Feature Graph .....................................28
Figure 7  HSV Color Space Representation .............................................36
Figure 8  Pairwise Distances Between Feature Vectors ...........................41
Figure 9  Shot Boundary Detection Results ..........................................42
Figure 10 Clustering Algorithm Results ................................................46
# LIST OF ALGORITHMS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Histogram Intersection Calculation</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Luminance Calculation</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>Single Color Band Histogram Calculation</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>Histogram Based Mean Calculation</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>Histogram Based Variance Calculation</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>Sobel Horizontal Mask</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>Sobel Vertical Mask</td>
<td>18</td>
</tr>
<tr>
<td>8</td>
<td>Frame Difference Calculation</td>
<td>19</td>
</tr>
<tr>
<td>9</td>
<td>Gaussian Smoothing Mask</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>Simple Summation Cut Detection Algorithm</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>Z-Score Adjusted Summation Cut Detection Algorithm</td>
<td>23</td>
</tr>
<tr>
<td>12</td>
<td>Key Frame Selection Algorithm</td>
<td>29</td>
</tr>
<tr>
<td>13</td>
<td>Feature Vector Euclidean Distance Measure</td>
<td>32</td>
</tr>
<tr>
<td>14</td>
<td>Feature Vector Squared Normalized Distance</td>
<td>33</td>
</tr>
<tr>
<td>15</td>
<td>RGB Histogram Intersection Distance</td>
<td>35</td>
</tr>
<tr>
<td>16</td>
<td>BSAS Algorithm</td>
<td>38</td>
</tr>
<tr>
<td>17</td>
<td>TTSAS Algorithm</td>
<td>40</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Dr. Evangelos Yfantis, for his assistance and guidance. Throughout grad school, he has offered me many opportunities to grow as an academic and as a person.

I am grateful to Dr. Ajoy Datta, Dr. John Minor, and Dr. Angel Muleshkov for taking time out of their busy schedules to be members of my committee.

Before coming to UNLV to peruse a master’s degree, I was a high school teacher, so I understand the sacrifices that many of my former teachers have made to grow and cultivate knowledge in me. Mr. Stephen Griffin, my first computer science teacher at Lincoln High School, whose enthusiasm for the field of computing inspired me to enter the field and make it my life’s work. Mrs. Sue Doker, not only was she my algebra teacher and the sponsor of the Mu Alpha Tehta math club that defined my high school experience, but also she is a true personal friend and mentor. Mrs. Janet Stephens, whose knack for creating fun and interesting statistics lessons has provided me with a solid background for many of the ideas presented in this thesis.

While in college, I was shaped in large part to a growth in personal skills and leaderships skills, as the result of my membership in the Phi Kappa Tau...
Fraternity. While serving as chapter president, I got the rare opportunity to shape, define, and assist a group of 50 men in a common goal. I would like to thank my mentors and friends in this organization Mr. Cheyenne Throckmorton, Mr. Michael Rosack, and Mr. Scott Baker that allowed me this opportunity.

I would also like to thank, my friends and coworkers at UNLV’s Digital Image Processing Lab, Mr. Dean Curtis, Mr. Chris Jones, and Mr. Jia J. Tse.

Of course none of my existence in this world would be possible without my family. My dad, Mr. Dean Bunch Esq., has become my life advisor and has greatly supported me in my transition from adolescence to adulthood. In any quandary or decision, my dad will always have clear and levelheaded advice. My mom, Principal Mrs. Martha Bunch, is my emotional rock. She has been a “mother” to thousands of Lincoln High School students, but I am privileged enough to call her “mom.” My sister, Dr. Lynn Bunch M.D., inspires me to achieve. She has set the intellectual achievement bar high, and I hope to be as successful as she is someday.
CHAPTER 1

INTRODUCTION

My interest in the subject of video analysis was sparked by my prior work in the field of document analysis. During my work with documents, I compiled a large database of documents and attempted to formulate a methodology to identify documents with similarities.

As a result of this, I began to consider other areas in which my document analysis experience could be used. I recognized the huge number of images that are available for public viewing on the Internet. As the universe of images grows, the limitations of current search methods, which consider image filenames but neglect their content, become an increasingly large impediment to the efficient accessing and use of these files.

Inherently, searching by file name, rather than content, is prone to error. For example, if a query seeks images corresponding to the word “bush”, images of President George Bush will be returned along with images of plants. A search of this type, even assuming the file names accurately describe the content of the images on the file, fall prey to the ambiguities of searches such as the one
described above. It is simply not clear from the query what type of content the user is seeking.

Additionally, the intentional misnaming of files to produce incorrect query results is common. For example, a malicious user could include the name “George Bush” in the file name of an image of a chimpanzee to induce a query to return the false image. Thus, even if the file names are assigned with the best of intentions, searching based on file names, or verbal descriptors that have been assigned to a file containing images, is a process that is innately flawed. Content-based analysis of image files is much less susceptible to foibles of human classification.

I further realized that the image search techniques I used in document analysis could be applied to large scale video databases. The YouTube phenomenon, which has allowed every Internet user to post videos for general viewing, has exponentially increased the potential applicability of such video analysis. Now virtually anyone can post videos on the Internet, from the casual filmmaker to presidential candidate Barrack Obama who announced on his candidacy via video on YouTube.

As I reviewed the literature, I found that video analysis was not a novel concept; the subject has been extensively discussed in the literature. However, a review of the techniques attempted to date indicates room for improvement.
Video retrieval and classification systems had been researched for many years before anyone imagined the advent of the YouTube era and the explosion of videos. Such retrieval and classification systems evolved and became more sophisticated. Current systems are composed of several stages, each serving an important purpose.

Most algorithms contain at least four main steps: shot boundary detection, key frame extraction, feature extraction, and clustering. Two additional steps which could be implemented are region extraction and concept detection. However, due in part to the difficulties associated with effectively implementing these last two features, they are not always included in an analysis and classification system. (Smeaton, 2003)

Shot detection divides the video into scenes with a common fluidly connected background. Determining the boundary of each shot, its beginning and ending frames, is possible using the known properties of scene transitions. Once the shots have been determined, the system should be capable of closely examining each shot to identify a representative frame that has the most in common with all the other shot's frames. Post-extraction, the system will focus its analysis on only these key frames to expedite the analysis process.

Subsequently, feature extraction yields a feature vector, a list of numerical values that represent the characteristics of each key frame. An example of such a feature would be the predominant color in the frame.
The feature vectors for the entire video library are then fed into the clustering algorithm. Clustering is the process of identifying sets of related objects. In this instance clustering finds groups of videos that have similar content.

In addition to these four major processes, region detection and concept detection processes are occasionally used. Region detection identifies large areas of key frames that have similar content. For example, region detection would separate the ground and the sky in a landscape scene. Identification and classification of these regions can be useful when comparing two videos.

Concept detection is used to identify the types of objects in a video, such as a human, a bird, a podium, or a television. Although humans easily recognize complex properties and subtle patterns, such identification is difficult for a computer.

I postulate that techniques that have been used successfully in document image analysis systems will also prove useful for clustering video content. I will apply my prior knowledge of document analysis techniques to review current video extraction systems and to improve upon methodologies used in video analysis.
CHAPTER 2

PREVIOUS RESEARCH

The field of video analysis has made substantial advances in the past few decades evolving from the examination of single images to systems that analyze hours of unedited, or “raw”, video. The first seminal work that is referenced in many subsequent papers is the work done on color indexing by Swain and Ballard. This work describes a methodology for creating a database of objects based on the color histograms of those objects and later effectively searching for those objects. This work became very important when later video analysis systems attempted to define videos based on objects that are present in video frames. (Swain & Ballard, 1991)

Among other advancements that Swain and Ballard introduced was a novel approach for determining the level of similarity between multiple color histograms. This approach examines each color band and calculates the number of pixels shared by the two histograms. It then uses a normalizing factor of the total image size such that a comparison value of zero denotes no similarity and a comparison value of close to one denotes high similarity.
Algorithm 1: Histogram Intersection Calculation

$$H(I,M) = \frac{\sum_{j=1}^{255} \min(I_j,M_j)}{\sum_{j=1}^{255} M_j}$$

where $I_j,M_j$ are color histogram values

This direction is apparent in a 1995 paper by Zhang, et al., “Automatic Parsing and Indexing of News Videos”. This paper draws upon some of the algorithms described by Swain and Ballard but also focuses on segmenting news footage into different shots. Zhang decided to examine news video for its relatively straightforward temporal syntax. News videos tend to follow a predictable pattern wherein a “stand up” showing the reporter is followed by an insert of images of the news event, and then returning for the “close” with the reporter at the end of the story. Zhang selected a very narrow scope because the video analysis techniques developed at the time were not strong enough to examine open-ended or unconstrained video feeds. (Zhang, 1995)

In 1997, John Smith and Shih-Fu Chang from Columbia University published a paper describing the creation of a system that allowed for a continuous search of images and videos on the web and categorizing them based on their visual content. Smith and Chang augmented the work done by Swain and Ballard by adding region-sensitive color analysis which identifies
common color themes in two different videos allowing for colors that may be in
different locations in the frame. They continued to use file names as a partial
descrIptor of the contents. If “dog” was part of the filename, the authors
would give some weight to the assumption that a picture of a dog was included
in the video depicted in the file. As stated in the introduction, this approach is
subject to error but was often found to be a very strong predictor of the
contents of the file. Therefore, Swain and Ballard determined that the use of
the file name in addition to content-based analysis aided in the analysis and
classification process and was superior to the use of content-based analysis
alone. (Smith 97)

Smith and Chang signaled the shift from video primarily considered to
be a broadcast medium with images produced only by professionals, to the
concept of “video for the masses” produced by amateurs and distributed freely
without editorial control on the Internet. The emergence of sites such as
YouTube and Google Video confirmed the accuracy of their belief as to the
direction and importance of video on the Internet in the years ahead.

In 2001, Salembier and Smith described the MPEG-7 video format that
for the first time provided the video creator with a means to attach descriptors
to the file in an attempt to describe the visual contents. Other information
such as the producer, the date of creation, and location can also be included.
(Salembier, 2001)
This facility, if used consistently, would make the task of finding all videos with commonalities very straightforward. However, limitations remain in this approach. For instance, additional time is required to annotate the video file properly and not all video producers will spend the time to do this. Also, many videos produced prior to the introduction of this format are incapable of annotation. Nevertheless, the addition of this facility to video files is a major step forward in cataloging video content.

One of the most common uses of videos resulting from the use of a camera is surveillance. Video surveillance cameras are numerous but rarely monitored actively in real time. This leads to large amounts of video that may contain important information but are never accessed. Hu, et al. proposed a system: (1) that can look for situations that are out of ordinary, and (2) that can be prompted to search for specific objects or combinations of objects. (Hu, 2007)

For example, cars ordinarily drive on the right side of the road, so if an image of a car driving on the left side appeared, the system could automatically alert the police to the presence of an erratic driver. If the user is looking for a specific person of interest with a particular piece and color of clothing (e.g. a blue baseball cap), large amounts of video data could be searched to attempt to find this piece of clothing and accordingly the person wearing it.
In addition to the published articles, major advancements in the area of video analysis, classification, and retrieval have debuted and were discussed at a series of conferences known as TRECVID. These conferences are sponsored by the National Institute of Standards and Technology (NIST), and in 2003, they were an outgrowth of a conference known as TREC that was devoted to text retrieval. The goal of TRECVID is to encourage research in the area by providing a large test collection, uniform scoring procedures, and a comparison of results. (Smeaton, 2003)

Since the inception of TRECVID, the conference has been instrumental in providing direction for areas of new interest. In an effort to lead the field, TRECVID annually announces its current and most challenging problems as well as a constantly evolving list of tasks. Some tasks have remained on the list for years due to their importance and level of difficulty. Shot boundary detection, high level feature extraction, and search make up the set of recurring tasks. The fourth set of tasks, those that have changed over the years, are story segmentation (2003-2004), low level feature extraction (2005), and the management of unedited BBC footage known as “rushes” (2006-2007). (Smeaton, 2003) (Kraaij, 2005) (Over, 2006) (Over, 2007)

In addition to the changes in tasks at TRECVID, the number of languages to be examined has also expanded. The conference began using English but added Chinese and Arabic in 2005 as languages to include in the
data set for tests of the various programs being evaluated. This broadening of
interest in non-English languages reflects the global nature of the conference
that attracts teams from around the world.

Hereafter, I will explore certain techniques displayed and evaluated at
TRECVID as they relate to Shot Boundary Detection (Chapter 4), Key Frame
Distance Calculation (Chapter 5) and Clustering (Chapter 6).
CHAPTER 3

PROJECT DESCRIPTION

Chapters 4, 5, and 6 describe the methodologies used to create a software system for the automatic comparison of videos. The system is a concatenation of four modules. Each module performs calculations to decompose the task into smaller and more manageable problems. Figure 1 shows a summary of this system.

Figure 1: Modular Software Design
CHAPTER 4

SHOT BOUNDARY DETECTION

A video is a connected set of individual frames. Within this set of frames are discontinuities or significant changes in scene or action. A single shot is defined as a set of video frames that continuously flow from one into another with only small changes from frame to frame. In video analysis, treating an entire shot as a single frame can expedite analysis and computation.

Locating the start and end frames is necessary to simplify the analysis. Section 4.1 will analyze the methodologies used to fragment a video stream into individual shots.

4.1 Transition Types

Video analysis would be considerably simplified if every time a video transitioned from one shot to another, this transition was accomplished in a uniform manner by all videographers. However, video producers use many different techniques to transition from one scene to another. Some scenes jump directly from the end frame of one scene to the start frame of the next scene with no smooth transition. This is referred to as a “cut” transition.
Other transitions occur more gradually. For example, a “fade out” is a scene transition in which the first scene slowly fades to black before the second scene begins. Unlike a cut scene transition, fading occurs over the span of multiple frames. When performing shot boundary detection, understanding the difference between these transitions is important to create algorithms that can identify each type of transition while taking into account the videographic differences of each.

4.2 Feature Buffer

In shot boundary detection a comparison of the current frame to frames that occurred previously may detect whether a distinct enough change occurred to warrant the creation or categorization of a new shot.

To detect a cut transition, examining the frame immediately prior to the current frame is all that is necessary, because a cut transition is such an abrupt transition. However, all other types of transitions require searching further back into frames prior to the frame immediately preceding the current frame. Setting up a feature buffer accomplishes this. This feature buffer will hold up to 10 frames in memory. The feature buffer allows comparisons between the current frame and prior frames to take place more quickly.

4.3 Feature Extraction

Feature extraction is the process of quantifying relationships and attributes of very complicated sets of data into subsets of data that a computer
can more easily use. Feature extraction is a concept that is used in many
different areas of computer science, because computers are much more able to
deal with numbers than with complicated abstract concepts.

To a computer, a video is a set of meaningless data. Our minds have
been trained to easily identify complex patterns, objects, and commonalities in
videos. Feature extraction attempts to bridge this gap of understanding between
human capabilities and computer abilities. Feature extraction processes data
from a video and puts it into relatively simple data (compared to the very
complex data needed to construct a video clip) that can be analyzed and
compared to other sets of data.

Selecting a finite list of features helps describe the object. After all, we
are trying to get the computer to emulate human recognition in identifying
videos that are similar.

Two main feature types are examined in each frame. First, aspects of the
image colors are examined. Color features are then augmented by features
based on edge content. These two types of features are distinct and necessary
because edges are not rooted in what particular colors are present in an image.
Therefore, if two frames have similar colors but are characterized by a change in
edge content, the system should recognize the difference.
Different detectors account for different types of transitions. However, requiring each detector to review the entire video to identify each type of transition is a lengthy process.

To avoid this delay, we created a system with only one level of feature extraction. This information is passed to the detectors to quickly process the data and determine the location and nature of each transition.

4.3.1 Color Based Features

To analyze the colors in a frame requires examining the frame color histogram computed on each color band, in contrast to the overall conventional color histogram that will be discussed later. This catalogs the number of pixels that occur for each color value. [Algorithm 3] Each color image is separated into three primary colors: red, green, and blue, and each of these three colors is assigned a numerical value between 0 and 255. In addition to the three standard color bands, we calculate the luminance, or intensity, of each pixel using the standard formula. Luminance is represented as a value between 0 and 255.

\[
Luma = 0.2126R + 0.7152G + 0.0722B
\]

where \(R, G, B\) are the primary color values

Algorithm 2: Luminance Calculation
for (each color band)
  for (each pixel)
    histogram[band][value] ++

Algorithm 3: Single Color Band Histogram Calculation

Using this color histogram, we present statistical computations that summarize the color attributes in a relatively succinct manner. These attributes will drastically reduce the number of values needed to assess a frame’s content.

The first of these statistical tools is a simple mathematical average. In each color band, we calculate the average value based on the histogram that has already been calculated for that color. [Algorithm 4] This statistic will tell us, among other things, which color (red, green, or blue) predominates in the image.

\[
\bar{x} = \frac{1}{h \cdot w} \sum_{i=0}^{255} i x_i
\]

where \( \{x_0, x_1, \ldots, x_{255}\} \) are the set of pixels with a value of \( i \) and \( h \) is the image height and \( w \) the image width

Algorithm 4: Histogram Based Mean Calculation

The next statistic under review is the variance. As with the mean, the variance is computed separately for each color band. The variance indicates how
each color band is arranged around the mean. The variance can determine if a
frame has very smooth colors or if the colors are more jagged.

\[
\sigma = \sqrt{\frac{1}{h \cdot w} \sum_{i=0}^{255} (i - \bar{x})^2 \cdot x_i}
\]

Algorithm 5: Histogram Based Variance Calculation

The last statistic that is calculated is the interquartile range. The
interquartile range (IQR) measures the range between the first and third
quartiles, i.e., the distance between the 25th percentile and the 75th percentile.
This is a measure of statistical variability that is better than the simple range
because the IQR is not affected by outliers.

4.3.2 Edge Based Features

Edges are areas within a frame that go abruptly from one color to
another. The edges identify areas of particularly drastic change. If similar edges
between two frames can be established, that these frames are and can be
deduced to be in the same shot. Several methods can create this linkage. We
have chosen to simply examine the number of horizontal edge pixels and the
number of vertical edge pixels as descriptors of frame edge content.

4.3.2.1 Sobel Edge Detection Algorithm

Algorithms for detecting edges vary in complexity and accuracy.
Arguments can be advanced in favor or opposed to any of the different edge
detection algorithms. For this count, we elected to use the Sobel edge detection algorithm because of its relatively low computation overhead and for its ability to distinguish between horizontal and vertical edges.

The Sobel edge detection algorithm works by using two convolution masks to determine the gradient of a pixel in either the vertical or horizontal direction. [Algorithm 6,7,8] This gradient value determines the amount of change between the target pixel and its neighboring pixels. A simple threshold can be applied to the resulting gradient values to determine whether a pixel should be considered an edge or non-edge pixel.

\[
\text{Horizontal Mask} = \begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1 \\
\end{bmatrix}
\]

Algorithm 6: Sobel Horizontal Mask

\[
\text{Vertical Mask} = \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1 \\
\end{bmatrix}
\]

Algorithm 7: Sobel Vertical Mask

4.3.3 Inter-frame Features

Inter-frame features are those that are calculated between two separate frames. Short-term features compare the current frame with a frame...
immediately prior to the current frame. Long-term features compare the current
frame to frames up to half a second before the current frame.

4.3.3.1 Video Object Plane

More recent video standards, such as MPEG-7 have enabled content-
based functionality as mentioned in Chapter 2. Video Object Planes (VOP’s)
are on the forefront of this movement. In this standard, each video sequence is
segmented into regions of semantically connected content, or VOP’s. Each
VOP contains information about those regions’ motion and shape. For our
purposes, a change in the number, location, or shape of the VOP’s would
potentially signal a new shot.

The processes of calculating VOP’s is multi-stage. First, the areas of
motion between frames are determined. This is done by calculating the
difference of the values between each pair of pixels in the frame. Algorithm 9
shows this difference calculation for images H and I.

\[
D(x, y) = |H(x, y) - I(x, y)| \quad \text{for each color band}
\]

Algorithm 8: Frame Difference Calculation

Next, the Canny edge detection algorithm is used to find the exact
outline of the movement in the neighboring frames. The Canny algorithm first
uses a Gaussian smoothing mask, which helps to filter out any excess noise
from the camera that might be misdetected as motion in the frame.
After smoothing the image, the Sobel Edge Detection algorithm is applied as described in Subsection 4.3.2.1. These values will be used to help establish which pixels are edge pixels. However, due to the VOP’s need for a precise outline, it is necessary to perform some further processing to limit these edge pixels to an outline that is a single pixel wide.

To achieve this single pixel wide outline, non-maximal suppression is used. This process helps differentiate between neighboring pixels that are both considered edge pixels. The process only selects the pixel with the largest value within a gradient direction, as computed with the Sobel algorithm.

The subsequent image will only have maximal edge pixels, but will also have gaps within an edge gradient. Therefore, it is necessary to fill in these gaps. To do so, the end of each edge gradient is examined to look for other edges that continue in this gradient but are buffered by a small amount of non-edge pixels. These non-edge pixels are then filled in as edge pixels.
Once the Canny detection is performed, the VOPs are calculated by looking for large areas of connected pixels. The difference between input, Sobel, difference, and Canny images can be seen in Appendix A.

4.4 Detectors

4.4.1 Cut Detector

Cut detection is the most important transition detector because it is the most common transition type. (Smeaton 2003) Cuts are normally used back and forth within one scene between characters who are talking with one another, raising the possibility of many cuts within a single scene. Often the transitions from scene to scene are more subtle, such as fades and dissolves. Due to the high volume of cut transitions, a detector must function quickly and effectively. The only characteristic of a cut transition is a very abrupt change, and these abrupt changes can be seen in any of the extracted features.

Our first attempt to find an accurate way to detect abrupt feature changes was to use a simple summation of all differences of feature values for neighboring frames. [Algorithm 11] This method is flawed for a number of reasons.

Firstly, various features have different magnitudes. For example, a vertical edge count ranges from zero to the total number of image pixels, whereas red band average intensity ranges from zero to 255. Algorithm 11
would weigh changes in edge pixel counts much higher than red band average intensity.

Secondly, features tend to change in drastically different manners. For example, red band IQR for a particular video may have a tendency to stay unchanged so any slight variation should be assigned substantial weight in the overall feature evaluation.

```
for(ea ch set of neighboring frames)
if ( \sum (all features) > threshold) classify as cut transition
```

Algorithm 10: Simple Summation Cut Detection Algorithm

Our second method for finding an accurate way to detect abrupt feature changes corrected the flaws described above. This method uses two passes through the video. The first pass calculates and stores the changes in each feature in each set of neighboring frames. The program then finds the average change in each feature and the standard deviation of the change in each feature. The second pass subsequently uses these values to calculate, for each set of frames, the standard deviations away from the mean (Z-Score) each feature has changed. The standard deviation calculations are summed to yield a final value against which to threshold.
for (each set of neighboring frames)

difference[frame#][feature#] = feature[frame#] - feature[frame#+1]

average[frame#][feature#] = \frac{\sum \text{difference[frame#][feature#]}}{\text{number of frames}}

sd[frame#][feature#] = \sqrt{\frac{\sum (\text{difference[frame#][feature#] - average[frame#][feature#]})^2}{\text{number of frames}}}

// PASS 2

for (each set of neighboring frames)

\text{if} \left( \frac{\sum (\text{difference[frame#][feature#] - average[frame#][feature#]})}{sd[frame#][feature#]} > \text{threshold} \right)

classify as cut transition

Algorithm 11: Z-Score Adjusted Summation Cut Detection Algorithm

By employing two passes, the adjusted Z-score feature summation method gains knowledge about the features and how they are changing throughout the video and is less susceptible to changes that may look drastic locally but are less dramatic when viewed in the context of the entire video.

After implementing both methods, we found a drastic improvement when using this method as compared to the basic summation method. In the figures below, the basic method displays many spikes, while the Z-score adjusted method displays only the three spikes corresponding to the three cut transitions in the video. [Figures 2,3]
Figure 2: Basic Feature Summation Method Graph

Figure 3: Adjusted Z-Score Feature Summation Method
4.4.2 Dissolve Detector

A dissolve is difficult to detect because the shot gradually changes from the current scene to the next scene. If only the current and previous frames are examined, it would not be possible to detect this type of transition because the change is too gradual. [Figure 4] Therefore, a frame which is many frames prior to the current one must be analyzed to determine whether or not a large amount of changes have occurred. The feature buffer explained in section 4.2 provides the ability to examine frames prior to the immediate past frame.

![Dissolve Transition IQR Feature Graph](image)

Figure 4: Dissolve Transition IQR Feature Graph
The solution we implemented is to sample every 20\textsuperscript{th} frame. This method is effective because it completely skips over the gradual transition and examines only the two frames from the different shots. However, this method does not come without flaws. (Yeo 1995)

Firstly, this method makes the assumption that shots are longer than 20 frames. Any shot less than 20 frames runs the risk of being skipped. However, at the common frame rate of 24 frames per second, such a shot would be shorter than one second and highly unlikely.

Also, this method has a tendency to trigger the detector within a shot with areas of high action or drastic change. Different problems arise depending on the threshold value. If the threshold is set too low, then the detector triggers falsely in areas where the shot does not actually change. If the threshold is set too high, then the detector often does not pick up on a new shot that has transitioned in. An appropriate threshold value must be found through trial and error. Setting this threshold a little too low is better than a little too high, because a few redundant shots are better than a few omitted shots.

4.4.3 Fade Out Detector

A fade out is one of the easiest transitions to detect due to the very obvious state that occurs at the end of the fade, an entirely black screen. The fade out detector triggers when the color IQR range is very closely centered around all black. The only difficulty in fade out detection is properly finding the
last frame of the prior shot, which is important because only frames fully 
representative of the shot should be included. Selecting a frame prior to the 
introduction of black pixels into the shot accomplishes this.

Figure 5: Fade Out Transition IQR Feature Graph

4.4.4 Fade In Detector

Fade in detection is very similar to fade out detection and is triggered by 
a scene that has a fully black IQR. The same process of determining when the 
next shot has fully transitioned out of the fade is necessary to determine a valid 
starting frame for the shot.
4.5 Verification and Collaboration

Due to multiple detectors for the different types of scene transition, two detectors could both possibly detect the same transition as two different transition types. This situation must be accounted for and corrected in the analysis. The rule implemented to resolve this conflict or tie is to give precedence to the longer transition, because if a cut transition is identified in the same time period as a fade transition, the fade may have been severe enough to be identified as a cut. In this case, the change from normal content to black
happened quickly enough for the cut detector to identify it as well, so the fade transition is given precedence over the cut transition due to its longer time span.

4.6 Key Frame Selection

At this point in the process, the video has been broken up into different shots that indicate similar content throughout each shot. Examining every frame in each shot to categorize what is occurring would be difficult. Selecting one key frame from each shot for future analysis solves this problem. The algorithm we implemented finds the frame within the shot closest to the mean of all feature values used during shot boundary detection.

```
for (each frame in shot)
    s = \sum |x_i - \bar{x}_i| \text{ where } \{x_1, x_2, ..., x_n\} \text{ are the frame features}
if (s < \text{min})
    \text{min} = s
    \text{keyframe} = \text{current frame}
```

Algorithm 12: Key Frame Selection Algorithm
CHAPTER 5

KEY FRAME DISTANCE CALCULATION

Once the video has been processed for shot boundary detection and key frame selection, the number of frames needed for analysis is reduced by a factor of around one hundred, and more time and effort can be concentrated on these few frames. The final result is those groups or clusters of key frames identified with great similarity.

To find these clusters of key frames, we must first develop ways to calculate the level of similarity between two key frames. This chapter will discuss a few different techniques for calculating a distance between key frames. This distance will be considered the level of similarity, zero being identical and a large number denoting little to no similarity. The next chapter will discuss the methodology behind using these distance measures to identify clusters.

5.1 Euclidian Distance Measure

Most distance measures in use by clustering algorithms are calculated based on a set of representative values, or feature vector. There are two main factors that go into any distance measure that falls into this category. First,
which features will comprise the desired set of features? Second, how will the
distance between a pair of feature vectors be calculated?

The first question is a relatively simple one. Consider that shot boundary
detection used a set of features to track changes in video content. This list
comprised of the following 14 values: red mean, red inter-quartile range, red
standard deviation, blue mean, blue inter-quartile range, blue standard deviation,
green mean, green inter-quartile range, green standard deviation, luma mean,
luma inter-quartile range, luma standard deviation, vertical edge count, and
horizontal edge count.

This set of features requires no computational overhead, because these
features had already been calculated during the shot boundary detection and are
already stored in the shot key frame class. This advantage can not be
overlooked, especially with large video databases. Using this set of features
streamlines the process.

The second question of distance has many answers since no uniform way
exists to calculate the distance between a set of vectors. This depends on what
properties define two vectors as close. For example, a distance measure could
be set up to compare only the magnitudes of two vectors. In most cases this
method makes little sense because two vectors with the same magnitude could
point in opposite directions. This would indicate that the vectors may have little
in common. For this reason, a distance measure must find some appropriate value of similarity between two vectors.

The most common distance measure for a wide variety of uses is a Euclidian measure. This simple measure is generally a good place to start because it gives each feature an even weight in determining the magnitude of the difference between two vectors.

\[ EUCLIDIANDISTANCE(a, b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \]

Algorithm 13: Feature Vector Euclidean Distance Measure

5.2 Squared Normalized Distance Measure

This measure falls into the same category as the Euclidian Distance Measure, a feature vector based distance calculation. As with the Euclidian Distance Measure described in section 5.1, the feature vector is borrowed from the standard set of features used in the Shot Boundary Detection task.

Where this method deviates from the previous method is in the way it calculates the distance. For each feature in a pair of feature vectors, this method divides the difference of the features by their sum and then squares that value. [Algorithm 15] This differs from the Euclidian distance in that it divides by the sum before it squares the term.
for two feature vectors $a$ and $b$

$$SQUARED\_NORMALIZED\_DISTANCE = \sqrt{\sum_{i=1}^{n} \left(\frac{a_i - b_i}{a_i + b_i}\right)^2}$$

Algorithm 14: Feature Vector Squared Normalized Distance

This difference may seem subtle, but it may make a large difference in the distance calculation. For example, in the Euclidian measure, a feature that has values of 1 and 3 in two input feature vectors would be considered a small change in respect to a feature that has values of 100 and 120. This is because the distance between 120 and 100, or 20, is significantly larger than the difference between 3 and 1, or 2. In the squared normalized distance however, the feature that has values of 1 and 3 would be considered much larger than that of a feature that has values of 100 and 120. Now consider, the difference over the sum of 3 and 1, or one half, compared to the difference over the sum of 120 and 100, or one eleventh.

Overall the main difference is that the squared normalized measure uses a relative change distance and the Euclidian measure uses absolute distance.

5.3 RGB Color Histogram Intersection Distance Measure

This method differs from the previous two in that it does not use a feature vector as the basis to compute distance. Instead, this method relies on a color histogram. Previously in section 4.3.1, we discussed calculating color
histograms for each color band (Red, Green, and Blue). Using separate band histograms is fine for the purpose of identifying changes in frame content, because a drastic change in the number of red pixels with intensity of 255 signals an overall change. However, using the same methodology to conclude two frames are similar is flawed.

To define similarity between two frames, we must match exact colors (RGB combinations). For instance, two frames with a similar number of pixels with the value <0,127,255> would be a good indicator of similarity. However, the total number of pixel values possible using RGB is 256³, or about 16 million. The chances of two images of the same object having many equal pixel values is very low because many issues, including lighting and orientation, can slightly change pixel values. Due to the high level of precision, this can have drastic effects.

To overcome this problem, we lump ranges of pixel values into “bins”. For example, with bins of size 16 in each band, the first bin would count all pixels in the range <0-15, 0-15, 0-15>. So, both <1,2,3> and <15,15,15> would both be counted in this bin. Using this method we can bring the total amount of combinations down to a much more reasonable number, \( \left( \frac{256}{16} \right)^3 \), i.e. around 4000.
Swain and Ballard found that by using color histograms and a distance measure called histogram intersection, objects could be effectively indexed and recalled. Histogram intersection compares two histograms by summing the minimum value for each bin. Then by dividing by a normalizing factor, two identical histograms will have an intersection value of 1, and two disjoint histograms will have a value of 0.

\[
HISTOGRAM\_INTERSECTION(I, M) = \frac{\sum_{j=1}^{n} \min(I_j, M_j)}{\sum_{j=1}^{n} M_j}
\]

where \(I_j\) and \(M_j\) are color histogram values

Algorithm 15: RGB Histogram Intersection Distance

5.4 HSV Color Histogram Intersection Distance Measure

As deduced by the names, the HSV color histogram method is exactly the same as the RGB color histogram method except for the color spaces. A color space is a way of representing a color in terms of intensity values. RGB is the most commonly used color space, because it is the model used by CRT monitors that add the three primary colors (Red, Green, and Blue) to produce any color. Other color spaces, such as HSV, have been proposed to have advantages over RGB. The main advantage of using HSV over RGB is that HSV is a better color space for image retrieval. This is because the Euclidian
distance in the HSV cone approximates the human perceived distance much better than RGB.

HSV stands for hue, saturation, and value and is represented graphically as a hexagonal cone. [Figure 7] Hue represents the color type from red, yellow, green, and so on. Hue is generally represented as a number between 0 and 360, or the angle on the color circle. Saturation is a value between 0 and 100%. The value 0 denotes a color which is completely unsaturated and looks like a shade of grey. The value 100 denotes a color that is completely saturated and contains no white component. Value, sometimes called brightness, also is a value between 0 and 100%. As the value goes from 0 to 100%, the pixel gets brighter.

Figure 7: HSV Color Space Representation
CHAPTER 6

CLUSTERING

Clustering is an unsupervised method for data analysis with a focus on determining points of data within a set that are closely related. Clustering is used in a wide variety of fields in and out of computer science, such as taxonomy, pathology, distributed computing, and artificial intelligence. Many clustering methodologies exist to attack different types of problems.

In general, clustering is broken up into six main steps: feature selection, proximity measure, clustering criterion, clustering algorithm, and interpretation of results.

Feature selection and proximity measure have been discussed in prior chapters. In this chapter we will focus on clustering criterion and clustering algorithms. In the next chapter we will interpret results.

For our purposes, clustering groups sets of shots with common features and a small distance measure, as discussed in Chapter 5. Understanding why the input is shots as opposed to videos is important. Videos contain many shots that may have little similarity. Therefore, clustering based on the shots is the wiser route. However, clustering based on shots does not mean that we cannot
find similarities. For example, if we are able to cluster shot A in video X and shot B in video Y, we could certainly say shots A and B are similar, but also say that videos X and Y are similar based on shots A and B.

6.1 Two-Threshold Sequential Clustering Scheme

The clustering algorithm, or scheme, is a set of rules and procedures about when to add data points to clusters, create new clusters, and potentially merge existing clusters. The decision of which clustering algorithm to use is an important one, because results between different algorithms that may appear similar can vary quite drastically.

We made the design decision to use the Two-Threshold Sequential Algorithmic Scheme (TTSAS). TTSAS has advantages over some simpler clustering algorithms such as Basic Sequential Algorithmic Scheme (BSAS).

BSAS uses a basic one-pass distance threshold where if a data point is within the threshold limit of an already existing cluster, it groups that point with that cluster. Otherwise, the data point becomes part of a new cluster.

```plaintext
for(all feature vectors in set)
    find MINIMUM _DISTANCE to cluster
    if(MINIMUM _DISTANCE < THRESHOLD)
        add feature vector to existing cluster
    else
        add to new cluster
```

Algorithm 16: BSAS Algorithm
BSAS is weak, because it allows no room for uncertainty. Early in the process, firmly establishing clusters is impossible. However, in BSAS, new clusters are created without regard for how far the feature vector is from the threshold. The only consideration is that the vector is over the threshold. Also, BSAS results change depending on the order that the algorithm examines the feature vectors.

We have chosen TTSAS because it is resistant to the problems faced by BSAS. TTSAS gives more discretion for the algorithm to create or not create a new cluster. It uses two thresholds. The lower bound threshold is a value for which, if a feature vector is a smaller distance away from the cluster than this value, it is added to this cluster. The upper bound threshold is a value for which, if a feature vector is a larger distance away from all clusters than this value, it is considered to be a new cluster. If the distance is between these two thresholds, it is put back in the queue of feature vectors to be classified. As clusters get added and as members get updated to existing clusters, the distance calculations change. The next time a queued feature vector passes through the system, it has a better chance of being classified. If no vectors are classified in a whole iteration through the queue, we have reached a non-terminating situation, and we must correct this by creating a new cluster with the first member in the queue and continuing on with the process. Doing this ensures that all vectors are classified in a cluster by the end of the procedure. [Algorithm 19]
add all feature vectors to queue
for (each vector queue)
    if (no vectors were added in previous for loop)
        add first vector to new cluster
        remove first vector from queue
    find MINIMUM _ DISTANCE to any cluster
    if (MINIMUM _ DISTANCE < lower threshold)
        add vector to MINIMUM _ DISTANCE cluster
        remove vector from queue
    else if (MINIMUM _ DISTANCE > upper threshold)
        add vector to new cluster
        remove vector from queue
    else
        append vector to queue

Algorithm 17: TTSAS Algorithm

6.2 Clustering Criterion

After the clustering algorithm has been established, how loosely or
tightly connected clusters are to be grouped must be determined. If the criteria
are too tight, this results in each feature vector being assigned to its own
individual cluster with no others similarly assigned. Such a result would not
provide any information concerning the set of data. If the criteria are too loose,
this results in a small number of clusters that contain all of the groups of the
data. Similar to the “tight” criteria, this does not provide enough
differentiation to yield any useful information. For the TTSAS, the two
thresholds set the level of sensitivity.
To determine these thresholds, we plotted a set of all distances for each pairwise distance between feature vectors in the shot system. [Figure 8]

Examining the plot, a cutoff was determined where all distances below were clearly related. Next, a cutoff where all distances above were clearly unrelated was established.

![Figure 8: Pairwise Distances Between Feature Vectors](image)

Examining the graph, we conclude that the significant dip at 15 is a good value for the lower bound threshold and due to the leveling out of the values at the higher levels, 60 represents an appropriate upper bound threshold.
EXPERIMENTAL RESULTS

Each portion of the system must be independently analyzed to determine its degree of effectiveness in recognizing and classifying portions of video and differentiating between classifications. The following charts and graphs demonstrate the relative effectiveness of each process.

7.1 Shot Boundary Detection Results

Two videos were analyzed: the first was a speech of President Bush and the second was a portion of a football game. The following chart shows the success of the program in the shot boundary detection of these videos for the purposes described previously.

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Actual Shots</th>
<th>Recognized Shots</th>
<th>False Positives</th>
<th>Unrecognized Shots</th>
<th>Total Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bush Speech</td>
<td>75 s.</td>
<td>22</td>
<td>21</td>
<td>0</td>
<td>1</td>
<td>94.45%</td>
</tr>
<tr>
<td>Football Play</td>
<td>406 s.</td>
<td>46</td>
<td>46</td>
<td>3</td>
<td>3</td>
<td>93.47%</td>
</tr>
</tbody>
</table>

Figure 9: Shot Boundary Detection Results

The data shown above was compiled by a human visual inspection of the videos and a manual count of the shots contained in both. This manual count
was then compared to the count which resulted from the shot boundary detection section of the program. However, the count of shots that was reported by the program is not positive proof of the actual number of shots which the program recognized, as evidenced by the football play results.

This is caused by two types of potential errors that can occur in the program’s analysis, which can potentially offset one another. First, the transition detector can fire in error resulting in an extra shot. Second, the transition detector could fail to detect a transition, resulting in a missed shot. Therefore, in the case of the football play even though the gross number of shots determined (46) was correct, only 43 out of 46 (93.5%) were correctly identified.

It is important to note that the Bush State of the Union Speech video was relatively simple because all of the transitions were cuts, as opposed to fades or other types. Also, the lighting was professionally staged and indoor, resulting in a much clearer picture than is the case with outdoor, natural light videos, or those where there is non-professional indoor lighting. The Football Play video depicts the famous University of California at Berkley versus Stanford game in which the last play includes many desperation laterals by Berkley resulting in a touchdown as the Stanford band runs onto the field before the game is concluded. This video is much more difficult to analyze because it contains cuts, dissolves and fades transitions, and outdoor lighting. Notwithstanding this
relatively complex video, as with the Bush speech, the detector is effective at a high rate.

7.2 Key Frame Distance Calculation Results

In order to test the effectiveness of the four distance measures used in the system, four sets of related images with four images in each set were used. Each of the 16 images was compared to each other, using each of the distance measures previously described. Both of the feature vector-based distance measures, Euclidian and Squared Normalized, performed better than the color histogram measures (RGB and HSV). Comparing the feature vector-based measures, Squared Normalized was superior to Euclidean; and between the color histogram measures, HSV was ranked over RGB.

To compute the effectiveness of each measure, the average distance between random images divided by the average distance between images in the same image "set" was computed. A larger gap between the numbers for random images and related images indicates a more effective measure. In summary, Squared Normalized was the most effective at 2.76; Euclidian was second at 2.68; with HSV at 2.43; and RGB last at 2.32. Appendix C contains the complete calculations.

Although it was determined that Squared Normalized is the best distance measure utilizing this mathematical calculation, it is unclear if this difference is statistically significant or would produce noticeably better results for clustering.
I hypothesize that the feature vector-based analysis performs better than the color histogram-based measures because the feature vectors are computed from a simple version of a color histogram that has many more complex calculations within it. Further, it is rational that Squared Normalized measure performed better than Euclidean, because it is a more complex distance measure, as described in Chapter 5. It also follows logically that HSV is better than RGB, because it is a superior color space when comparing images, as described in Chapter 5. Both of these conclusions are significant, demonstrating that the theory that a more complex measure results in a superior measure holds true in practice as well as theory.

7.3 Clustering Results

As noted in Chapter 6, it is difficult to specify or describe in advance the exact attributes of successful clustering due to the ambiguous nature of the task. Judgment is necessary to determine whether the resulting clusters are neither too large nor too small to provide meaningful information. In the program presented here, the objective is to have shots that are from the same video or contain similar content and need to be clustered together.

The clustering algorithm developed in this research was preliminarily run on a set of eight videos. Further research will involve additional videos. After running the clustering algorithm, the clusters were examined to determine how much of the content is consistent with other things within the cluster.
<table>
<thead>
<tr>
<th>Cluster Number</th>
<th>Shots In Cluster</th>
<th>Shots With Similar Content</th>
<th>Consistency Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 1</td>
<td>2</td>
<td>2</td>
<td>100%</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>35</td>
<td>28</td>
<td>80%</td>
</tr>
<tr>
<td>Cluster 3</td>
<td>38</td>
<td>28</td>
<td>74%</td>
</tr>
<tr>
<td>Cluster 4</td>
<td>16</td>
<td>13</td>
<td>81%</td>
</tr>
<tr>
<td>Cluster 5</td>
<td>8</td>
<td>12</td>
<td>67%</td>
</tr>
<tr>
<td>Cluster 6</td>
<td>4</td>
<td>4</td>
<td>100%</td>
</tr>
</tbody>
</table>

Figure 10: Clustering Algorithm Results
CHAPTER 8

CONCLUSIONS

This thesis presents a system for comparing videos based on their content. Each aspect of this project demonstrates positive results. However, each task that was performed using the program developed as a part of this project, and the associated research in the field, demonstrate that video analysis, classification, and differentiation are very much in their infancy.

Shot Detection showed the best results, which is to be expected because it is the easiest task. It would appear that algorithms exist to detect each different transition type. Further research in this area will include finding frames within a shot where the content has changed significantly enough to warrant a new shot and key frame, even though a traditional transition has not taken place.

Key Frame Distance Calculation also showed promising results. It was found that feature vector based calculations outperformed color histogram calculations. In the future, work with more advanced feature vectors could further increase the effectiveness of these measures. Also, it is clear that detection of precise objects or people using object and facial recognition will
advance this area significantly in the near future. Being able to identify that President Bush is in a video, is a much stronger descriptor of the video than the color traits of that video.

Clustering results were the most difficult to formulate, although they appeared to be positive. Future work should extend the ideas presented in this thesis to a large scale video database. A large video database would introduce increased problems with differentiability between videos due to the large cardinality of the database.

The analysis performed and program developed in this thesis demonstrate promise and the results suggest that further work in this area will be productive in advancing the reliability and usefulness of video analysis.
APPENDIX A

EDGE DETECTION PICTURES

Sample Input Image

Sample Sobel Image
VIDEO OBJECT PLANE PICTURES

Sample Difference Image

Sample Canny Image
FEATURE GRAPHS

Appendix A contains the graphs of the features stated in section 4.3.1.
The first four parts show each of the four main transition types in section 2.4.
The fifth part shows graphs from a 20 second video clip which contains 3 cut transitions. These cuts are at the 363\textsuperscript{rd}, 459\textsuperscript{th}, and 584\textsuperscript{th} frames respectively.
These graphs include the feature graphs included in the previous four parts and adds the non-adjusted sum graph as a reference for comparison against the Z-score adjusted summation graph.
PART I: CUT TRANSITION

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
PART II: DISSOLVE TRANSITION

Red Average

Red IQR
PART III: FADE OUT TRANSITION

Red Average

Frames

0 5 10 15 20 25 30 35 40 45

0 10 20 30 40 50 60 70

Red IQR

Frames

0 5 10 15 20 25 30 35 40 45

0 10 20 30 40 50 60 70

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
PART IV: FADE IN TRANSITION

Red Average

Red IQE
PART V: 20 SECOND VIDEO CLIP WITH 3 CUT TRANSITIONS

Red Average

Red IQR
MORE 20 SECOND VIDEO

Red Standard Deviation

Vertical Edge Count

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
MORE 20 SECOND VIDEO, Difference in regular sum versus adjusted sum
APPENDIX C

KEY FRAME DISTANCE CALCULATIONS

The following are spreadsheets used to calculate which distance measure was the best performer. These distance measures are described in chapter 5.
### Euclidian Metric:

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>230.4712</td>
<td>329.6324</td>
<td>536.5501</td>
<td>6027.268</td>
<td>7880.761</td>
<td>8484.422</td>
<td>2362.467</td>
<td>3212.511</td>
<td>2024.060</td>
<td>1826.859</td>
<td>3421.092</td>
<td>358.092</td>
<td>4797.229</td>
<td>2841.761</td>
<td></td>
</tr>
<tr>
<td>230.4712</td>
<td>0</td>
<td>298.9765</td>
<td>443.4541</td>
<td>5991.637</td>
<td>7821.793</td>
<td>8474.177</td>
<td>2239.805</td>
<td>3378.749</td>
<td>2008.905</td>
<td>1740.754</td>
<td>3324.769</td>
<td>3508.102</td>
<td>4755.031</td>
<td>2752.38</td>
<td></td>
</tr>
<tr>
<td>329.6324</td>
<td>298.9765</td>
<td>0</td>
<td>264.5664</td>
<td>3540.537</td>
<td>3595.842</td>
<td>7839.745</td>
<td>6521.659</td>
<td>1859.037</td>
<td>3181.237</td>
<td>1711.478</td>
<td>1417.992</td>
<td>3153.795</td>
<td>3153.258</td>
<td>4581.421</td>
<td></td>
</tr>
<tr>
<td>536.5501</td>
<td>443.4541</td>
<td>264.5664</td>
<td>0</td>
<td>5340.537</td>
<td>3198.148</td>
<td>2604.360</td>
<td>3984.118</td>
<td>6399.448</td>
<td>8421.993</td>
<td>6534.433</td>
<td>5019.043</td>
<td>3694.594</td>
<td>3772.979</td>
<td>2672.354</td>
<td></td>
</tr>
<tr>
<td>6027.268</td>
<td>7821.793</td>
<td>3540.537</td>
<td>5340.537</td>
<td>0</td>
<td>5530.278</td>
<td>6196.192</td>
<td>4352.984</td>
<td>5609.571</td>
<td>4012.707</td>
<td>3328.615</td>
<td>2731.222</td>
<td>2668.765</td>
<td>2603.466</td>
<td>2510.483</td>
<td></td>
</tr>
<tr>
<td>7880.761</td>
<td>8474.177</td>
<td>3198.148</td>
<td>2604.360</td>
<td>5530.278</td>
<td>0</td>
<td>2557.252</td>
<td>8424.199</td>
<td>10643.560</td>
<td>6713.760</td>
<td>7596.503</td>
<td>5653.374</td>
<td>5890.271</td>
<td>4793.869</td>
<td>4701.537</td>
<td></td>
</tr>
<tr>
<td>8484.422</td>
<td>1859.037</td>
<td>3181.237</td>
<td>3984.118</td>
<td>4352.984</td>
<td>8423.995</td>
<td>0</td>
<td>2116.776</td>
<td>1859.037</td>
<td>3198.148</td>
<td>2604.360</td>
<td>3984.118</td>
<td>6399.448</td>
<td>8421.993</td>
<td>6534.433</td>
<td></td>
</tr>
<tr>
<td>2362.467</td>
<td>3212.511</td>
<td>1711.478</td>
<td>6399.448</td>
<td>5609.571</td>
<td>3198.148</td>
<td>2604.360</td>
<td>0</td>
<td>3198.148</td>
<td>2604.360</td>
<td>3984.118</td>
<td>6399.448</td>
<td>8421.993</td>
<td>6534.433</td>
<td>5019.043</td>
<td></td>
</tr>
<tr>
<td>3212.511</td>
<td>2024.060</td>
<td>1417.992</td>
<td>3595.842</td>
<td>4012.707</td>
<td>3328.615</td>
<td>2731.222</td>
<td>2668.765</td>
<td>0</td>
<td>3198.148</td>
<td>2604.360</td>
<td>3984.118</td>
<td>6399.448</td>
<td>8421.993</td>
<td>6534.433</td>
<td></td>
</tr>
<tr>
<td>2024.060</td>
<td>1826.859</td>
<td>1417.992</td>
<td>7839.745</td>
<td>4012.707</td>
<td>3328.615</td>
<td>2731.222</td>
<td>2668.765</td>
<td>3198.148</td>
<td>0</td>
<td>2557.252</td>
<td>8424.199</td>
<td>10643.560</td>
<td>6713.760</td>
<td>7596.503</td>
<td></td>
</tr>
<tr>
<td>1826.859</td>
<td>3421.092</td>
<td>1417.992</td>
<td>6521.659</td>
<td>4012.707</td>
<td>3328.615</td>
<td>2731.222</td>
<td>2668.765</td>
<td>2604.360</td>
<td>2557.252</td>
<td>0</td>
<td>2557.252</td>
<td>8424.199</td>
<td>10643.560</td>
<td>6713.760</td>
<td></td>
</tr>
<tr>
<td>3421.092</td>
<td>358.092</td>
<td>1417.992</td>
<td>1859.037</td>
<td>4012.707</td>
<td>3328.615</td>
<td>2731.222</td>
<td>2668.765</td>
<td>3540.537</td>
<td>3181.237</td>
<td>2604.360</td>
<td>0</td>
<td>2557.252</td>
<td>8424.199</td>
<td>10643.560</td>
<td></td>
</tr>
<tr>
<td>358.092</td>
<td>4797.229</td>
<td>1417.992</td>
<td>3595.842</td>
<td>5609.571</td>
<td>3198.148</td>
<td>2604.360</td>
<td>3984.118</td>
<td>5340.537</td>
<td>4012.707</td>
<td>3328.615</td>
<td>2731.222</td>
<td>0</td>
<td>3198.148</td>
<td>2604.360</td>
<td></td>
</tr>
<tr>
<td>4797.229</td>
<td>2841.761</td>
<td>1417.992</td>
<td>7839.745</td>
<td>5609.571</td>
<td>3198.148</td>
<td>2604.360</td>
<td>3984.118</td>
<td>5340.537</td>
<td>4012.707</td>
<td>3328.615</td>
<td>2731.222</td>
<td>3198.148</td>
<td>0</td>
<td>3198.148</td>
<td></td>
</tr>
<tr>
<td>2841.761</td>
<td>2752.380</td>
<td>1417.992</td>
<td>6521.659</td>
<td>5609.571</td>
<td>3198.148</td>
<td>2604.360</td>
<td>3984.118</td>
<td>5340.537</td>
<td>4012.707</td>
<td>3328.615</td>
<td>2731.222</td>
<td>3198.148</td>
<td>3198.148</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**Average Distance Between Two Random Images:** 3881.931

**Average Distance Between Two Images in Class A:** 262.9654

**Average Distance Between Two Images in Class B:** 301.543

**Average Distance Between Two Images in Class C:** 1443.573

**Average Distance Between Two Images in Class D:** 1057.276

**Average Distance Between Two Images in Same Class:** 1445.337

**Note:** The symbol '0' in the distance matrix indicates the same image, infinity completely different images
<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Squared Normalized Metric:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1 (Average Distance Between Two Random Images)</td>
<td>0.27595</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2 (Average Distance Between Two Image in Class A)</td>
<td>0.054059</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3 (Average Distance Between Two Image in Class B)</td>
<td>0.111667</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4 (Average Distance Between Two Image in Class C)</td>
<td>0.074541</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1 (Average Distance Between Two Image in Class D)</td>
<td>0.099838</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2 (Average Distance Between Two Images in Same Class)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note**: The symbol `@` symbolizes the same image, 1 completely different images.
### RGB Histogram:

|     | A1 | A2 | A3 | A4 | B1 | B2 | B3 | B4 | C1 | C2 | C3 | C4 | D1 | D2 | D3 | D4 |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| A1  | 1  | 0.787044 | 0.572227 | 0.558307 | 0.490663 | 0.457617 | 0.663417 | 0.476471 | 0.201055 | 0.136315 | 0.162448 | 0.282214 | 0.322096 | 0.282214 | 0.322096 |
| A2  | 1  | 0.787044 | 0.535964 | 0.528047 | 0.445547 | 0.42255 | 0.529544 | 0.453255 | 0.238086 | 0.163056 | 0.199999 | 0.340495 | 0.343516 | 0.386966 | 0.273516 | 0.338491 |
| A3  | 0.572227 | 0.535964 | 0.518503 | 0.445547 | 0.42255 | 0.529544 | 0.453255 | 0.238086 | 0.163056 | 0.199999 | 0.340495 | 0.343516 | 0.386966 | 0.273516 | 0.338491 | 0.290676 |
| A4  | 0.558307 | 0.528047 | 0.518503 | 0.445547 | 0.42255 | 0.529544 | 0.453255 | 0.238086 | 0.163056 | 0.199999 | 0.340495 | 0.343516 | 0.386966 | 0.273516 | 0.338491 | 0.340651 |
| B1  | 0.19104 | 0.18604 | 0.174479 | 0.174042 | 0.166193 | 0.86261 | 0.626957 | 0.69265 | 0.667037 | 0.887722 | 0.125702 | 0.205223 | 0.255244 | 0.224365 | 0.209615 | 0.340534 |
| B2  | 0.17675 | 0.160034 | 0.14158 | 0.18319 | 0.168319 | 0.760183 | 0.66263 | 0.626957 | 0.69265 | 0.887722 | 0.125702 | 0.205223 | 0.255244 | 0.224365 | 0.209615 | 0.340534 |
| B3  | 0.164085 | 0.165687 | 0.137969 | 0.132386 | 0.132386 | 0.51586 | 0.51586 | 0.51586 | 0.51586 | 0.51586 | 0.51586 | 0.51586 | 0.51586 | 0.51586 | 0.51586 | 0.51586 |
| B4  | 0.186122 | 0.175313 | 0.177877 | 0.17705 | 0.62957 | 0.663903 | 0.73735 | 0.73735 | 0.73735 | 0.73735 | 0.73735 | 0.73735 | 0.73735 | 0.73735 | 0.73735 | 0.73735 |
| C1  | 0.050264 | 0.043383 | 0.069232 | 0.059521 | 0.09075 | 0.09075 | 0.09075 | 0.09075 | 0.09075 | 0.09075 | 0.09075 | 0.09075 | 0.09075 | 0.09075 | 0.09075 | 0.09075 |
| C2  | 0.034079 | 0.026315 | 0.048199 | 0.040756 | 0.040756 | 0.040756 | 0.040756 | 0.040756 | 0.040756 | 0.040756 | 0.040756 | 0.040756 | 0.040756 | 0.040756 | 0.040756 | 0.040756 |
| C3  | 0.040612 | 0.032754 | 0.05876 | 0.049397 | 0.055534 | 0.081921 | 0.081921 | 0.081921 | 0.081921 | 0.081921 | 0.081921 | 0.081921 | 0.081921 | 0.081921 | 0.081921 | 0.081921 |
| C4  | 0.075553 | 0.061036 | 0.093089 | 0.086124 | 0.090449 | 0.118425 | 0.112952 | 0.099797 | 0.795739 | 0.628971 | 0.628971 | 0.628971 | 0.628971 | 0.628971 | 0.628971 | 0.628971 |
| D1  | 0.080524 | 0.071917 | 0.100788 | 0.086799 | 0.131982 | 0.15906 | 0.147922 | 0.142396 | 0.142396 | 0.142396 | 0.682741 | 0.824086 | 0.824086 | 0.824086 | 0.824086 | 0.824086 |
| D2  | 0.100163 | 0.091423 | 0.111725 | 0.096742 | 0.163356 | 0.18693 | 0.18693 | 0.18693 | 0.18693 | 0.18693 | 0.18693 | 0.18693 | 0.18693 | 0.18693 | 0.18693 | 0.18693 |
| D3  | 0.072575 | 0.06748 | 0.080456 | 0.060379 | 0.143402 | 0.161455 | 0.168037 | 0.162266 | 0.162266 | 0.162266 | 0.162266 | 0.162266 | 0.162266 | 0.162266 | 0.162266 | 0.162266 |
| D4  | 0.085133 | 0.081296 | 0.096128 | 0.08422 | 0.134154 | 0.159788 | 0.161781 | 0.144541 | 0.144541 | 0.144541 | 0.144541 | 0.144541 | 0.144541 | 0.144541 | 0.144541 | 0.144541 |

Average Distance Between Two Random Images: 0.322997
Average Distance Between Two Images in Class A: 0.725011
Average Distance Between Two Image in Class B: 0.74612
Average Distance Between Two Image in Class C: 0.786105
Average Distance Between Two Images in Class D: 0.793271
Average Distance Between Two Images in Same Class: 0.762927
### HSV Histogram:

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>0.77106</td>
<td>1</td>
<td>0.473568</td>
<td>1</td>
<td>0.513411</td>
<td>0.43125</td>
<td>1</td>
<td>0.45133</td>
<td>0.404885</td>
<td>0.503477</td>
<td>0.125703</td>
<td>0.39154</td>
<td>0.79128</td>
<td>0.17913</td>
<td>0.338126</td>
<td>0.429463</td>
</tr>
<tr>
<td>A2</td>
<td>0.513411</td>
<td>0.473568</td>
<td>1</td>
<td>0.813229</td>
<td>0.429701</td>
<td>0.451377</td>
<td>0.580234</td>
<td>0.189189</td>
<td>0.627999</td>
<td>0.143654</td>
<td>0.280742</td>
<td>0.325866</td>
<td>0.361654</td>
<td>0.229011</td>
<td>0.320564</td>
<td>0.488125</td>
</tr>
<tr>
<td>A3</td>
<td>0.527396</td>
<td>0.493125</td>
<td>0.813229</td>
<td>1</td>
<td>0.43125</td>
<td>0.463462</td>
<td>0.439452</td>
<td>0.192969</td>
<td>0.65703</td>
<td>0.150443</td>
<td>0.271536</td>
<td>0.303177</td>
<td>0.343008</td>
<td>0.226663</td>
<td>0.308069</td>
<td>0.416914</td>
</tr>
<tr>
<td>A4</td>
<td>0.193751</td>
<td>0.193802</td>
<td>0.167852</td>
<td>0.167969</td>
<td>0.1</td>
<td>0.680496</td>
<td>0.784422</td>
<td>0.509326</td>
<td>0.069504</td>
<td>0.026805</td>
<td>0.0493125</td>
<td>0.097499</td>
<td>0.136734</td>
<td>0.195735</td>
<td>0.144485</td>
<td>0.161423</td>
</tr>
<tr>
<td>B1</td>
<td>0.190623</td>
<td>0.193303</td>
<td>0.177022</td>
<td>0.174805</td>
<td>0.680496</td>
<td>1</td>
<td>0.744659</td>
<td>0.502741</td>
<td>0.091568</td>
<td>0.041896</td>
<td>0.055578</td>
<td>0.166717</td>
<td>0.177475</td>
<td>0.220408</td>
<td>0.176147</td>
<td>0.19313</td>
</tr>
<tr>
<td>B2</td>
<td>0.163589</td>
<td>0.163727</td>
<td>0.145059</td>
<td>0.141107</td>
<td>0.50459</td>
<td>0.475628</td>
<td>1</td>
<td>0.40507</td>
<td>0.06276</td>
<td>0.023392</td>
<td>0.045488</td>
<td>0.092406</td>
<td>0.116615</td>
<td>0.145265</td>
<td>0.111009</td>
<td>0.126104</td>
</tr>
<tr>
<td>B3</td>
<td>0.194661</td>
<td>0.196671</td>
<td>0.172764</td>
<td>0.181229</td>
<td>0.506326</td>
<td>0.502741</td>
<td>1</td>
<td>0.384971</td>
<td>0.034586</td>
<td>0.073176</td>
<td>0.105509</td>
<td>0.139186</td>
<td>0.175715</td>
<td>0.141459</td>
<td>0.157028</td>
<td></td>
</tr>
<tr>
<td>B4</td>
<td>0.033867</td>
<td>0.031426</td>
<td>0.047454</td>
<td>0.048482</td>
<td>0.004482</td>
<td>0.058604</td>
<td>0.05276</td>
<td>1</td>
<td>0.542897</td>
<td>0.772347</td>
<td>0.615648</td>
<td>0.141459</td>
<td>0.157028</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>0.009206</td>
<td>0.009788</td>
<td>0.0157</td>
<td>0.016425</td>
<td>0.017165</td>
<td>0.026813</td>
<td>0.023392</td>
<td>0.022308</td>
<td>0.542897</td>
<td>1</td>
<td>0.62221</td>
<td>0.491507</td>
<td>0.066544</td>
<td>0.148764</td>
<td>0.009046</td>
<td>0.09351</td>
</tr>
<tr>
<td>C2</td>
<td>0.021611</td>
<td>0.019782</td>
<td>0.035964</td>
<td>0.037611</td>
<td>0.031305</td>
<td>0.041976</td>
<td>0.045488</td>
<td>0.046833</td>
<td>0.772347</td>
<td>0.62221</td>
<td>1</td>
<td>0.616548</td>
<td>0.29657</td>
<td>0.041208</td>
<td>0.036133</td>
<td>0.03431</td>
</tr>
<tr>
<td>C3</td>
<td>0.051631</td>
<td>0.044798</td>
<td>0.070186</td>
<td>0.067884</td>
<td>0.062399</td>
<td>0.081035</td>
<td>0.092406</td>
<td>0.067526</td>
<td>0.621797</td>
<td>0.491507</td>
<td>0.615648</td>
<td>1</td>
<td>0.141423</td>
<td>0.139072</td>
<td>0.14492</td>
<td>0.148636</td>
</tr>
<tr>
<td>C4</td>
<td>0.098151</td>
<td>0.084531</td>
<td>0.081396</td>
<td>0.075794</td>
<td>0.087151</td>
<td>0.113584</td>
<td>0.116615</td>
<td>0.068079</td>
<td>0.045156</td>
<td>0.006654</td>
<td>0.029657</td>
<td>0.141423</td>
<td>1</td>
<td>0.45553</td>
<td>0.642237</td>
<td>0.615055</td>
</tr>
<tr>
<td>D1</td>
<td>0.120311</td>
<td>0.10687</td>
<td>0.096443</td>
<td>0.085752</td>
<td>0.12527</td>
<td>0.141061</td>
<td>0.143286</td>
<td>0.114688</td>
<td>0.056611</td>
<td>0.14876</td>
<td>0.041268</td>
<td>0.139072</td>
<td>0.45553</td>
<td>1</td>
<td>0.590359</td>
<td>0.587516</td>
</tr>
<tr>
<td>D2</td>
<td>0.077021</td>
<td>0.073664</td>
<td>0.042673</td>
<td>0.056641</td>
<td>0.092471</td>
<td>0.112734</td>
<td>0.116098</td>
<td>0.096563</td>
<td>0.051452</td>
<td>0.059046</td>
<td>0.063133</td>
<td>0.144492</td>
<td>0.642372</td>
<td>0.509359</td>
<td>1</td>
<td>0.0618</td>
</tr>
<tr>
<td>D3</td>
<td>0.104229</td>
<td>0.095652</td>
<td>0.08014</td>
<td>0.077227</td>
<td>0.103311</td>
<td>0.123504</td>
<td>0.100498</td>
<td>0.047477</td>
<td>0.095651</td>
<td>0.03431</td>
<td>0.148636</td>
<td>0.615055</td>
<td>0.587516</td>
<td>0.6618</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**Average Distance Between Two Random Images:** 0.284864

**Improvement from Random to Related Images:** 2.433437

**Note:** 1 symbolizes the same image, 0 completely different images

**Average Distance Between Two Image in Class A:** 0.699054

**Average Distance Between Two Image in Class B:** 0.68274

**Average Distance Between Two Image in Class C:** 0.7083

**Average Distance Between Two Images in Class D:** 0.682701

**Average Distance Between Two Images in Same Class:** 0.693199
REFERENCES


68


VITA

Graduate College
University Of Nevada, Las Vegas

John Alexander Bunch

Address:
Las Vegas, NV 89119

Degrees:
Bachelor of Science, Computer Science 2004
Georgia Institute of Technology, Atlanta, GA

Publications:

Thesis Title:
Content-Based Video Classification and Comparison

Thesis Examination Committee:
Chairperson, Evangelos Yfantis, Ph.D.
Committee Member, Ajoy Datta, Ph.D.
Committee Member, John Minor, Ph.D.
Graduate Faculty Member, Angel Muleshkov, Ph.D.