SIMILARITY ANALYSIS OF VIDEO SEQUENCES USING AN ARTIFICIAL NEURAL NETWORK

by

FENG CHEN

(Under the Direction of Suchendra M. Bhandarkar)

ABSTRACT

Comparison of video sequences is an important operation in many multimedia information systems. The similarity measure for comparison is based on some measure of correlation with the perceptual similarity (or difference) among video sequences or with the similarity (or difference) in some measure of semantics associated with the video sequences. In content-based similarity analysis, the video data are expressed in terms of different features. The similarity matching is then performed by quantifying the feature relationships between target video and query video shots, with either an individual feature or with a feature combination. In this study, two approaches are proposed for the similarity analysis of video shots. In the first approach, mosaic images are created from video shots, and the similarity analysis is done by examining the similarity amongst the mosaic images. In the second approach, the key frames are extracted for each video shot, and the similarity amongst video shots is examined by comparing the key frames of the video shots. The features extracted include image histograms, slopes, edges, and wavelets. Both individual features and feature combinations are used in similarity matching using an artificial neural network models. The similarity rank of query video shots is determined based on the coefficients of determination and mean absolute errors.

INDEX WORDS: Video similarity, Similarity matching, Video shot, Mosaic image, Key frame, Individual feature, Feature combination, Artificial neural network.

SIMILARITY ANALYSIS OF VIDEO SEQUENCES USING AN ARTIFICIAL NEURAL NETWORK

by

FENG CHEN

B.S., Zhejiang University, China, 1983

M.S., Chinese Academy of Sciences, China, 1989

Ph.D., The University of Georgia, 1998

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF COMPUTER SCIENCE

ATHENS, GEORGIA

2003

SIMILARITY ANALYSIS OF VIDEO SEQUENCES USIN AN ARTIFICIAL NEURAL NETWORK

by

FENG CHEN

Major Professor: Dr. Suchendra Bhandarkar

Committee: Dr. Hamid Arabnia
Dr. Daniel Everett

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
May 2003

ACKNOWLEDGEMENTS

I would like to express my special thanks to Dr. Suchi M. Bhandarkar, my major professor, for his oversight and insight in helping me complete this thesis. Without his continued support, this project would not have been successful. Thanks also to Dr. Hamid R. Arabnia and Dr. Daniel M. Everett for serving on my committee.

I would like to express my deepest acknowledge to my family. Their patience, understanding, encouragement, and love have been my essential support.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

Comparison of video sequences is an important operation for many multimedia information systems. In general, a simple pixel-by-pixel comparison, or exact matching, between the corresponding frames does not work except for highly constrained situations (Adjeroh et al. 1998; Santini and Jain 1999; Lim et al. 2001). The similarity analysis is usually performed on video shots with different geometries (such as different frame sizes, orientations, and coordinate systems), different objects, different scenes, and different events, with or without similar properties amongst them. The measurement of similarity needs to be based on some measure of correlation with the perceptual similarity (or difference) among video sequences or with the similarity (or difference) in some measure of semantics associated with the video sequences.

In content based similarity analysis, the video data are expressed in terms of different features. The similarity matching can be performed by quantifying the relationships between the features that are extracted from the video data. Deriving a similarity measure can be considered as a three-step process. First, we need to find a set of features that are good enough to describe the characteristics of the video sequences. These features define the feature space for the similarity analysis. Next, the feature vectors need to be constructed from the video data. The components of the feature vectors can be the color or gray-level histogram, texture, shape, and/or other features

extracted from the video stream. Finally, the similarity matching between video sequences is performed based on certain criteria, such as a distance measure or correlation efficient, between the feature vectors derived from the video sequences in the feature space.

There are several issues that we need to discuss here. First, during extraction of feature vectors, the spatial information for some features may be lost because this information was not considered in the feature extraction process. For example, if the image content is characterized by the color or gray-level histogram, all the information about the spatial distribution of colors or gray-levels will be lost. One way to recover some spatial information is to divide the image frame into tiles and then compute the histogram for each tile. The spatial information can also be introduced by classifying the image frame into different regions and computing the histogram for each region. The second issue is that the various features such as color, texture, and shape are generally extracted using different computational methods in different feature spaces with different inherent distance metrics. As a result, content based similarity matching is usually performed based on a single individual feature. Since the different features may place different degrees of importance or focus on different aspects of the original video data, the result of similarity matching may be different when the different features are used in the single feature matching process although the underlying video data set is the same in every case. A more robust strategy is to combine the features in a reasonable way. For this purpose, the weighted linear method was developed to combine the similarity measurements from different individual features (Pentland et al. 1994, Ma and Manjunath 1998).

Another important issue is the selection of weights for the different components of feature vectors. In existing systems, the weights are usually arbitrarily specified by the user. For example, the user may specify "the weights for color feature, texture, and shape as 0.5, 0.3, 0.2, respectively". However, similarity matching using this kind of arbitrary weight specification may not yield results that are perceptually intuitive. The flexible adjustment of the weights based on the underlying visual data set may be necessary. The combination function of the feature vectors may be an another issue. Studies have already shown that the various features are not necessarily linearly related (Sheikholeslami et al. 1998; Santini and Jain 1999; Lee and Yoo 2000). Thus, a non-linear combination should be incorporated in the similarity matching. An artificial neural network (ANN) could be used for this purpose (Sheikholeslami et al. 1998; Lee and Yoo 2000; Lim et al. 2001). The final issue is the criteria used in similarity matching. In most of the current research, the similarity matching is based entirely on the distance between feature vectors. It is possible that alternative criteria may be used instead of the distance criteria.

Based on our knowledge, the most published studies were focused on the similarity analysis for images. Fewer studies were conducted on video analysis. In this study, we propose a new system model that combination of heterogeneous features of video shots for supporting content-based video retrieval. Our approach is based on ANN models that can be used to determine the non-linear relationship between different features extracted from video data. Although there are some studies that use ANN for image retrieval, little work has been done with video data. The input to the ANN model is the set of measurements of features that are extracted from video data. Unlike previous studies that use the distance criterion as a similarity measure, the coefficient of

determination between the target video shot and the query video shots is used as the primary criterion in our similarity analysis. In addition, the mean absolute error between the target video shot and the query video shots is also used as another criterion in the similarity analysis. The relative merits/demerits of these two criteria are also discussed in our study. The coefficients of determination have values between 0 and 1. A value of close to 1 represents high similarity, whereas a value of close to 0 represents low similarity between the target and the query video shots. The mean absolute errors have positive values with a high value standing for low similarity and low value standing for high similarity. Instead of arbitrarily signing the weight to each feature, the ANN can train on the training dataset, and then set weights for each of the features.

The general objective of this study is to perform similarity matching for video shots. Different similarity measure methods will be used in the study. The detailed goals for the study are listed as follows:

- Extract feature vectors based on color (and gray level) histogram, slope and slope direction, edge direction, and wavelets;
- Incorporate spatial information into the feature vectors;
- Allow for a combination of any of the feature vectors;
- Allow initial weight adjustment amongst the feature vectors;
- Explore a weighted non-linear feature combination for similarity matching using an ANN;
- Rank the video shots based on a similarity measure computed using the coefficients of determination and the mean absolute errors.

CHAPTER 2

LITERATURE REVIEW

With the rapid development of digital media, people have easy access to tens of thousands of visual databases containing images and videos. This trend is likely to continue, providing people with access to increasingly larger visual databases at a rapid speed. With growing sizes of visual databases, traditional methods of finding a specific piece of visual data break down. One commonly used searching method with traditional database tools is to index the visual database with keywords followed by keyword-based queries. Such an approach requires a person to tag all the visual images or videos with key words, a time-consuming task. This approach is often not workable in practice. The keyword approach has the problem that some visual aspects are inherently difficult to describe, while others are equally well described in many different ways (Niblack et al. 1993). Even though with good description, it may be difficult for the user to guess which visual aspects have been indexed. Similarity retrieval was introduced as an alternative strategy for querying, visual databases, containing images and videos.

Similarity retrieval is usually performed based on the features that can be extracted from the visual data, such as color, texture, shape, and motion (Pentland et al. 1994, Ma and Manjunath 1998, Antani et al. 2002). These features comprise the feature space in similarity analysis. Since these features describe the content of the visual data, the similarity retrieval is also called “content-based retrieval”. The literature review will

include three parts. In the first part, the basic description of similarity theory will be presented. Then we will review the current status of research related to similarity-based retrieval. Next we will give a review of related studies on various feature extraction techniques that pertain to our study. Finally, we will review the studies on applications of ANNs and other technologies to similarity analysis. Although our study focuses on similarity analysis for video data, much of the literature review deals with research on image similarity because the methods developed for image retrieval may also be, and actually are, used to retrieve video data with or without modification.

The basics of theories toward similarity measures

The early development of the similarity theory is rooted in the detection and recognition literature (Shepard 1964, Ashby and Perrin 1988). The general recognition theory assumes that the perceptual effect of a stimulus is random but that in any single trial it can be represented as a point in a multidimensional space. Similarity is a function of the overlap of perceptual distributions (Ashby and Perrin 1988). The most important models in similarity theory are dominated by geometric models. In these models, the distance (mainly geometric distance) has played a key role. The distance generally measures the dissimilarity in the similarity models. In order to use the distance for similarity, the essential assumption is that similarity and dissimilarity are inversely related (Tversky 1977, Ashby and Perrin 1988, Santini and Jain 1999).

In similarity theory, there are two different types of dissimilarity, or similarity (Tversky 1977, Ashby and Perrin 1988, Santini and Jain 1999), perceived dissimilarity (represented by d) and judged dissimilarity (represented by d). If A and B represent the

stimuli a and b in the feature space, then d(A, B) is the perceived distance between the two, and the judged distance can be represented as follows:

$$d(A, B) = g[d(A, B)] \quad [1]$$

where g is a monotonically non-decreasing function.

Stimuli are represented in a metric space. d(A, B) is the distance function of this space and is usually represented with geometric distance (such as Euclidian distance). In general, we say d satisfies the metric axioms (Tversky 1977, Krumhansl 1985, Ashby and Perrin 1988, Santini and Jain 1999) although the requirement for these axioms to be satisfied is also under debate (Tversky 1977, Rosh 1975, Tversky and Gati 1982, Ashby and Perrin 1988). In most studies, these axioms are assumed to be satisfied, and are detailed as follows:

a. Self-dissimilarity:

$$d(A, A) = d(B, B) \quad [2]$$

b. Minimality:

$$d(A, B) = d(A, A) \quad [3]$$

c. Symmetry:

$$d(A, B) = d(B, A) \quad [4]$$

d. The triangular inequality:

$$d(A, B) + d(B, C) = d(A, C) \quad [5]$$

Studies on similarity analysis of video sequences

Most studies defined the similarity of video shots as the similarity of images chosen from the video shot to represent each video shot (Zhang et al. 1995, Zhang et al. 1997, Shan and Lee 1998, Jain et al. 1999). These images are typically the sets of key frames (Yeung and Liu 1995, Zhang et al. 1997, Shan and Lee 1998, Shan and Lee 2001). In some early studies, the similarity between two video shots was defined as the maximum similarity between the pairs of key frames for these two video shots (Zhang et al. 1997). The problem of this definition with similarity measurement is that it leaves out the temporal information in the video shots. Shan and Lee (1998 and 2001) proposed a series of measures based on the similarity of the frame sequence which takes temporal ordering into consideration. They also presented the algorithms for both the symmetric similarity measures and the asymmetric similarity measures. The asymmetric similarity measure is used when the symmetry axiom is not satisfied. Besides the key frame-based similarity analysis, Zhang et al. (1997) proposed shot-based retrieval. In order to utilize the temporal information, they defined the similarity between two shots as the sum of color histogram similarity between key frames of the two shots, and the average and variance of the frame-to-frame histogram difference within each shot. They also used a motion feature to represent the temporal information by first detecting the direction distribution of motion vectors and then estimating the average speed and standard deviation in a given direction.

Adjeroh et al. (1998) showed that video sequence matching can be classified into three types: (i) scene-to-scene: check if two scenes are similar; (ii) scene-to-sequence: check if a scene is similar to the query scene hasoccurred in the database sequence; (iii)

sequence-to-sequence: check if two sequences are similar. They then reduced the three types of matching to one basic problem, that is, to find the solution for type (i): the scene-to-scene matching problem. In order to solve the similarity problem, they first represented the video sequence as a video string (vString), and then proposed the vString edit distance in their similarity study. Studies were also conducted to examine the similarity between images with wavelets (Hirata and Kato 1992, Stollnitz et al. 1996). Hirata and Kato (1992), in their research of image similarity with wavelets, represented the distance measure between two images as the difference of the image means and the differences of their wavelet coefficients (weights). A simpler but more efficient method was proposed by Jacobs et al. (1995) where they simply set the wavelet coefficients to zero and one. The similarity between two images then was equivalent to checking the number of zeros and ones that are matched between the two images. The dissimilarity distance between features was used as the underlying criterion in almost all the studies in similarity analysis.

Feature extraction for similarity analysis

The features typically extracted from visual data can be grouped into four types: color based, texture based, shape based, and motion based features. The combination of features has also studied by researchers and used in some retrieval systems. The following sections will give a review of previous studies on feature extraction. The literature review for color based, texture based, and shape based features is derived mainly from studies dealing with similarity analysis of images.

1. Color based features

Color plays a very important role in current content based visual retrieval systems. Color (image) histogram methods have been studied with different color spaces, including RGB (Androutsos et al. 1999, Kankanhalli et al. 1999, Sebe and Lew 2001), HIS (Jain et al. 1999, Sebe and Lew 2001), L*u*v* (Kankanhalli et al. 1996, Del Bimbo et al. 1998), L*a*b*, YC_bC_r (Inoue et al. 2000, Qiu 2002), and Munsell space (Zhang et al. 1997). The basic approach underlying the use of color features is to use a global color histogram. The similarity between two images is evaluated by comparing their color frequency distributions (Swain and Ballard 1991, Rubner et al. 1998). The main problem with this approach is the loss of spatial distribution of colors over the images. Because of this, several studies were conducted in order to introduce spatial information in the color histogram to a certain degree. One approach to introducing spatial information in the histogram is to partition the image into regular or irregular blocks. Each block is associated with an individual local histogram (Del Bimbo et al. 1998, Colombo and Del Bimbo 1999, Jain et al. 1999, Inoue et al. 2000). Colombo and Del Bimbo (1999) in their research first segmented the image into irregular blocks based on color distribution. Each block is considered as a homogeneous area with a fixed set of spatial relations with other blocks. Jain et al. (1999) used another approach in their research. In order to incorporate spatial information in the color space, they divide an image into five quarters, including top left quarter, top right quarter, bottom left quarter, bottom right quarter, and center quarter. They compute the histogram but reduce it to 64 levels for each quarter. Introducing spatial information typically increase the data volume. Inoue et al. (2000) convert the image from RGB system into gray-scale image by using the Y value of the

YCrCb color system. The Y value is a linear combination of the R, G, B values at each pixel location. Therefore, the image data volume in their similar analysis is greatly reduced, and the processing was more efficient.

An alternative approach is to partition the image into connected entities with homogeneous chromatic content (Del Bimbo et al. 1998, Smith and Chang 1999, Kankanhalli et al. 1999). This approach clusters image histograms around dominant components by performing an automated segmentation, and then determines entities as image segments with connected pixels under common dominating colors. In general, a few colors are sufficient to partition the histogram into cohesive clusters, which can be represented as a single average color without significant loss of information for the purpose of evaluation of similarity. Berretti et al. (2002) proposed a slightly different approach to represent color clusters. In their research, they also partitioned image space in color clusters collecting pixels with common chromatic attributes, but regardless of their spatial distribution within separate segments. They argued that this approach improves perceptual robustness and facilitates matching and indexing. There are different classifiers used in image clustering. Kankanhalli et al. (1996) in their similarity analysis used the minimum distance classifier to adjust for pixel population. They also used a classifier based on the Markov random field process to examine the spatial correlation property of image pixels.

Cha and Srihari (2002) divided the color features for similarity measurements into three types, namely nominal, ordinal, and modulo. In the nominal feature type, each value of the measurement is named. In ordinal feature type, the values can be ordered. Most measurements are of the ordinal type. An example of this type is the gray-level in gray

scale images. In the modulo feature type, the measurement values form a ring due to the arithmetic modulo operation. The modulo type histograms are obtained along the angular values such as directions or hue in color images. By analyzing the computational complexities for different methods for computing distance measurements including their own, they found that the computational complexities for the nominal, ordinal, and modulo feature types can be O(b), O(b), and $O(b^2)$, where b is the number of levels in histograms.

2. Texture based features

Texture is generally identified as the visual characteristics of invariance of certain local attributes, such as homogeneous regions, or significant local variations in local attributes in images. These characteristics may comprise of specific visual patterns or spatial arrangements of pixels. Typically, textures have strong statistical and/or structural properties with "repeating patterns". This allows that the retrievals based on texture features are effective for uniform texture images or regions. There are a number of methods for analyzing image textures. Smith and Chang (1994) used the quadrature mirror filter to represent the image texture with a quad-tree based segmentation of image. Jain et al. (1999) defined the texture in an image as the edge direction histogram. The Canny edge detector was used to extract the edges from the image.

The Gabor filter and wavelets are also used to generate the texture features (Manjunath and Ma 1996, Ma and Manjunath 1998, Puzicha et al. 1997, Mandal et al. 1999, Idrissa and Acheroy 2002). Manjunath and Ma (1996 and 1998) extract the image texture by filtering image with a bank of Gabor filters. The texture gradients are then

computed to segment the image into homogeneous regions. The basic assumption in their studies is that the texture regions are locally homogeneous. Puzicha et al. (1997) use Gabor filters to generate the image texture coefficients. The probability distribution function of the texture coefficients is used in conjunction with several distance measures to determine image similarity. The distances used are the Kolmogorov-Smirnov distance, x^2-statistics, the Jeffery divergence, weighted mean and variance among others. Idrissa and Acheroy (2002) also apply Gabor filters to obtain the texture features, but they use the fuzzy c-means clustering for unsupervised classification. Jacobs et al. (1995) describe how a Haar wavelet decomposition of the query and database images can be used to satisfy a content-based query both quickly and efficiently. Mandal et al. (1999) propose a fast wavelet-based histogram technique to index texture images. The proposed technique reduces the computation time compared to other wavelet-based histogram techniques.

3. Shape based features

Shape is an important aspect of the semantic content of images and is thus an important component for image retrieval. The extraction of shapes is usually based on an edge map. Shapes are created by linking the edge fragments in the edge map. Without edge linking, shape features cannot be extracted (Zhou and Huang 2001). Shape features are also subjected to change if the distance between camera and object changes or the object is rotated (Mokhtarian and Abbasi 2002). Because of these factors, shape features are usually difficult to track or computationally too expensive to be implemented in a general purpose content-based image retrieval system for real world images (Zhou and

Huang 2001). In general, shape features are usually effective for the retrieval of images with simple objects.

Different approaches have been taken for matching shape features in the content-based image retrieval studies. One approach is to adopt deformable image templates to match user-drawn sketches to the database images (Del Bimbo and Pala 1996, Scarloff 1997, Pala and Santini 1999). Since the user sketch may not be an exact match of the shape in the database, elastic deformation of the user template is used to match the image contours. An image for which the template has to undergo minimal deformation, or expend minimum energy, is considered as the best match (Del Bimbo and Pala 1996, Scarloff 1997, Pala and Santini 1999). Gunsel and Tekalp (1998) define a shape-based similarity metric based directly on the elements of the mismatch matrix derived from the eigenshape decomposition. A proximity matrix is formed using the eigenshape of representation objects. The distance between the eigenvectors of the query and target object proximity matrices formed the mismatch matrix, and the similarity is determined based on the mismatch matrix.

A different approach is to use implicit polynomials for effective representation of geometric shape structures (Alferrz and Wang 1999, Petrakis and Milios 1999, Mokhtarian and Abbasi 2002). This method is based on fitting a polynomial to a curve patch. A vector consisting of the parameters of this curve is used to match the image to the query. The assumption is that a typical database would contain the boundary curve vectors at various resolutions to make the matching robust. Alferez and Wang (1999) presented a method to index shapes which is invariant to affine transformations, rigid-body motion, perspective transforms and change in illumination. They used a

parameterized spline and wavelets to describe the objects. Petrakis and Milios (1999) used a dynamic programming based approach for matching shapes at various levels of shape resolution. Mokhtarian and Abbasi (2002) have investigated shape similarity retrieval under the affine transform using the concept of boundary curvature scale space.

Other approaches to shape-based retrieval are also found in the literature. Jain and Vailaya (1998) analyze the problems involved in trademark registration, and propose a computational strategy in which multiple feature description schemes of the same visual shape are used to improve retrieval accuracy without significantly increasing computational cost. They represent the trademark images in terms of invariant moments and the histogram of the edge directions, and integrate the dissimilarity of these features by using a weighted mean. Ciocca and Schettini (2002) use a relevance feedback approach to improve the effectiveness of image retrieval from a trademark database. Fudos and Palios (2002) present an efficient matching algorithm built around a novel similarity criterion and based on shape normalization about the shape's diameter, which reduces the effects of noise during the shape extraction procedure. The matching works by gradually inflating the query shape until the best match is discovered. Zhou and Huang (2001) define structure features as the features that lie between texture and shape features, and then propose an approach for image retrieval with edge-based structural features. These structural features can be extracted from the edge map of the original image with a "water-filling algorithm".

4. Combination of features

Features such as color, texture, and shape are normally generated using different computation methods. Thus, different features may reflect different properties of the visual data and have different underlying similarity measurements (Sheikholeslami et al. 1998, Lee and Yoo 2000). Because of this, each feature may only play a partial degree of importance in similarity analysis with that single feature. In order to take advantage of different features, several approaches have been proposed that combine different features together into a single feature vector. The similarity analysis then can be done based on the feature combination.

There are two major types of feature combination techniques encountered in current research, ones based on a linear relationship and those based on a non-linear relationship between features. The linear combination technique uses a weighted linear function to combine the similarity measurements of different feature classes. For given similarity measurements (m1, m2, …, mr) with respect to feature classes (f1, f2, …, fr) and the corresponding weights (w1, w2, …, wr), the overall similarity is calculated as $\mathrm{Sum}^{r}_{i=1}\ w_i\ m_i$ (Lee and Yoo 2000). Many studies have been done using the linear feature combination (Jain et al. 1999, Zhong and Jain 2000, Chan and Chang 2001). Jain et al. (1999) compute the dissimilarity distances (indices) of video shots with color, edge direction, and motion features. An integrated dissimilarity index was then defined by a linear combination using three weights which are normalized for color, edge direction, and motion respectively. Zhong and Jain (2000) present a method for combining color, texture and shape for retrieval of objects from an image database without preindexing the database in the discrete cosine transform (DCT) domain where the stored images are in

the JPEG-compressed format. Mojsilovic et al. (1999) extract various color related features such as the overall color, directionality, regularity, purity, and build a rule based grammar to identify each image. The image is described by a set of features and weights associated with the features. The weights are computed as a function of the features' frequency. The similarity is determined based on a linear weighted combination of features. Chan and Chang (2001) introduce a run-length feature, which integrates the information on color and shape of the objects in an image. They test the run-length feature to examine the similarity of images with relatively simple objects, and find that this feature can effectively discriminate the directions, areas, and geometrical shapes of objects.

Studies have shown that various feature classes are not necessarily linearly related. A model with a non-linear combination function may produce a more accurate similarity comparison between visual data such as images and videos (Sheikholeslami et al 1998, Lee and Yoo 2000). Artificial neural network (ANN) models are a good choice for the non-linear combination of various features. The input to the ANN is the set of measurements of individual image features and the output from the ANN is the similarity criterion that signifies the similarity of images. Sheikholeslami et al (1998) developed an approach for merging heterogeneous features using an ANN. They considered the texture and color feature classes. The back propagation algorithm with a single hidden layer was used to train the ANN. The experiments from their study show that the retrieved images based on merged heterogeneous features conform to human perception more than those derived from individual features. Lee and Yoo (2000) also introduce an ANN-based flexible image retrieval system. They use the radial basis function (RBF) network to

combine the values of the heterogeneous features. The nonlinear relationship between features is then developed to support the similarity comparison between images. Lim et al. (2001) present an ANN-based learning algorithm for adapting the similarity matching function to the user query's preference based on his/her relevance feedback. The relevance feedback is given as ranking errors (misranks) between the retrieved and desired lists of multimedia objects.

CHAPTER 3

METHODOLOGY

Video parsing and data preparation

The purpose of video parsing in this study is to preprocess the data set to extract features, which are then used for similarity matching. In this study, different video clips are collected by downloading them over the Internet. Two types of data sets are prepared, one to be used for mosaic based similarity analysis, and the second to be used for frame based similarity analysis. The purpose of video parsing is to generate video shots from the downloaded video clips. The video parsing procedure is as follows. For each video sequence, the locations of scene cuts are first determined. Then, based on the detected scene cuts, the video shots are extracted from the video clip. The video shots derived from video clips are then used for further processing. After extracting all video shots (that will be used for feature extraction and similarity analysis), these video shots are further used to create the panoramic images in case where the panoramic images can be created. The panoramic images are then used for mosaic based similarity analysis. For key frame based similarity analysis, the key frame images for each video shot are extracted and are ordered based on their temporal order within the video shots. The feature extraction is performed for both the mosaic images and the key frame images.

Feature extraction

In our study, the features extracted from the video shots include the image histogram, slope and slope direction, edge information, and the wavelet transform. These four feature subvectors comprise the feature space for this study. The following sections will describe each of the feature subvectors in further detail. The final feature vector is the concatenation of the four feature subvectors.

1. Image histogram

For an 8-bit image, the range of pixel intensity values is between 0 and 255. If we consider the image *I* that has a pixel intensity value from 0 to N-1 (for the case of an 8-bit per pixel image, N = 256), then the image histogram *H(I)* is a feature vector (h_1, h_2, …, h_N). For an RGB color image, the image histogram will be a feature vector that consists of all the three bands (h^R_1, h^R_2, …, h^R_N, h^G_1, h^G_2, …, h^G_N, h^B_1, h^B_2, …, h^B_N). There are two issues we need to consider regarding the representation of this feature vector. Since the goal of similarity matching is to compute the similarity between visual data sets as a whole, a detailed representation of the image histogram for each h_i (i = 1, 2, …, N) may not improve the result of the similarity analysis but rather, may introduce noise and result in a larger data volume. Based on this consideration, the above color histogram was grouped based on an interval value of V over the range [0, 2^b-1] (where b = 8 in our case). The second issue is that the representation of the image histogram will lose any spatial information contained within the visual data. In order to introduce spatial information within the image histogram feature to a certain degree, the image is further divided into tiles, and the image histogram for each tile is then computed.

The color histogram represents the pixel value distribution within the image. The number of distinct components for the image histogram feature can be determined using the parameters: number of bits per color per pixel (b), the value of the color or gray-scale interval (V) and the number into which each dimension of the image is divided into (D), as follows:

$$N_{pat} = C * D^2 * 2^b / V \qquad [6]$$

where N_{pat} is the total number of components, C is the number of image bands involved (for an RGB color image C=3, for a gray-scale image C=1), and D, b, and V are defined as discussed before. Note that dividing the image into D intervals along each of the dimensions results in D^2 image tiles.

2. Slope and slope direction

Slope is the maximum change between the intensity value of the center pixel and its neighboring pixels within a mask of predetermined size (such as 3x3). The slope direction is the direction of this maximum change. In general, the slope can be considered as a measure of the local intensity variations (changes) within the image, where as slope direction indicates the direction of the intensity variation (change).

In this study, we compute the slope and slope direction within a 3x3 mask. The computation of slope is done as follows (Moore et al 1993):

$$S_k(i,j) = \begin{cases} |P(i,j) - P(ii,jj)| & \text{if } (i = ii \text{ and } j \neq jj) \text{ or } (j = jj \text{ amd } i \neq ii) \\ & \text{and } |i-ii| = 1 \text{ and } |j-jj| = 1 \\ |P(i,j) - P(ii,jj)| / \text{Sqrt}(2) & \text{if } i \neq ii \text{ and } j \neq jj \\ & \text{and } |i-ii| = 1 \text{ and } |j-jj| = 1 \end{cases} \qquad [7]$$

$$S(i,j) = \text{Max}[S_k(i,j)] \qquad k = 0,1,2,\ldots,7 \qquad [8]$$

where S(i,j) is the slope at center location (i,j), P(i,j) is the pixel value at center location (i,j), P(ii,jj) is the pixel value for the neighboring location surrounding the center location (i,j).

Since the result of computation is a floating point number, the final result is converted into an integer and is scaled to lie between 0 and 255.

The slope direction at location (i,j) is computed based on the value of k in equation [8] (when S(i,j) is determined) and the sign of [P(i,j) – P(ii,jj)], as follows:

$$SD(i,j) = \begin{cases} k & \text{if } P(i,j) - P(ii,jj) > 0, \text{ and } S_k(i,j) = S(i,j) \\ k + 8 & \text{if } P(i,j) - P(ii,jj) < 0, \text{ and } S_k(i,j) = S(i,j) \\ 16 & \text{if } P(i,j) - P(ii,jj) = 0, \text{ and } S_k(i,j) = S(i,j) \end{cases} \qquad [9]$$

The slope feature, including slope magnitudes and slope directions, represents the local changes in pixel values in the image. The total number of distinct components for slope and slope direction are computed based on the histograms of slope magnitudes and slope direction values. Similar to the procedure for computing the total number of the distinct components for image histogram, the total number of distinct components for slope feature is computed using the same values of C, V, D, and as b in equation [6]. For computing the number of components for the slope direction, only the values of C and D are needed. Therefore, the total number of distinct components for slope and slope direction is:

$$SN_{pat} = C * D^2 * 2^b / V + C * 17 * D^2 \quad [10]$$

where SN_{pat} is the total number of components for slope and slope direction feature, and C, D, V, and b have the same meanings as in equation [6].

3. Edge feature

The edge information we used in our study is the edge direction. In our study, the Canny edge detector is used (Canny 1986). The final result is the edge information that is represented by the edge direction. The steps for performing the Canny edge detector are as follows:

1. smooth image to eliminate noise: in our study, a 5x5 Gaussian filter (s = 1.4) is used to remove noise.
2. compute edge magnitude: a 3x3 Sobel operator is used to compute the gradient values in horizontal and vertical directions. The magnitude is computed as follows:

   $$|G| = |Gx| + |Gy| \quad [11]$$

3. compute edge direction: in this step, the angle with respect to the positive X axis direction is computed based on the following equation:

   $$A = \text{invtan}(Gy/Gx) \quad [12]$$

   Four directions are determined, in which the direction is set to 1 for A with value between 0 and 22.5 as well as between 157.5 and 180, set to 2 for A with value between 22.5 and 67.5, set to 3 for A with value between 67.5 and 112.5, set to 4 for A with value between 122.5 and 157.5.

4. apply non-maximum suppression: when the edge directions are known, non-maximum suppression is performed in the gradient direction of each pixel to suppress any pixel value (set it to 0) that is not considered to be an edge.
5. apply double thresholding: the purpose of this step is to eliminate streaking. Streaking is the fragmentation of an edge contour caused by the filter operator output fluctuating above and below the threshold. The double thresholding technique uses two threshold values, T1 and T2, representing a high and a low value respectively. Any pixel that has a value greater than T1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixel that is connected to this edge pixel and that has a value greater than T2 is also selected as an edge pixel. Other pixels are considered to non-edge pixels (set to 0).

The total number of components for the edge features is computed within non-overlapping tiles in the image. In other words, the distribution of edge directions is computed for each tile. Since four directions are detected, the total number of patterns for the edge feature is:

$$EN_{pat} = 4 * C * D^2 \qquad [13]$$

where EN_{pat} is the total number of distinct components for the edge feature, and C and D have the same meanings as in equation [6]. The distribution of edge distributions is a measure of the oriented image texture within the image.

4. Wavelets

Wavelets are a mathematical tool used for hierarchical decomposition of functions and have been applied in many problems in computer graphics. There are different types of wavelets. In this study, the Haar wavelets are used as features for similarity analysis. The main reasons for selection of the Haar wavelets include their relative simplicity, orthogonality, and easy normalization. In V^0 space, the scaling function and the Haar wavelets are expressed as follows:

$$\phi(x) = \begin{cases} 1 & \text{for } 0 = x < 1 \\ 0 & \text{otherwise} \end{cases} \qquad [14]$$

$$\varphi(x) = \begin{cases} 1 & \text{for } 0 = x < 1/2 \\ -1 & \text{for } 1/2 = x < 1 \\ 0 & \text{otherwise} \end{cases} \qquad [15]$$

In V^j space, they can be expressed recursively, as follows:

$$\phi_i^j(x) = \phi(2^j x - i) \qquad i = 0, 1, \ldots, 2^j-1 \qquad [16]$$

$$\varphi_i^j(x) = \varphi(2^j x - i) \qquad i = 0, 1, \ldots, 2^j-1 \qquad [17]$$

The normalized wavelets can be expressed by incorporating the constant $\text{sqrt}(2^j)$ to the right of the above two equations, as follows:

$$\phi_i^j(x) = \text{sqrt}(2^j)\,\phi(2^j x - i) \qquad i = 0, 1, \ldots, 2^j-1 \qquad [18]$$

$$\varphi_i^j(x) = \text{sqrt}(2^j)\,\varphi(2^j x - i) \qquad i = 0, 1, \ldots, 2^j-1 \qquad [18]$$

In order to utilize the wavelets, we would like to express a function as a linear combination of wavelet functions, as follows:

$$F^j(x) = \text{Sum } c_i^j \ \phi_i^j(x) \qquad \text{for } i = 0, 1, \ldots, 2^j-1 \qquad [19]$$

Usually, $F^j(x)$ has a unique decomposition:

$$F^j(x) = F^{j-1}(x) + G^{j-1}(x) \qquad \text{for } i = 0, 1, \ldots, 2^j-1 \qquad [20]$$

where

$$F^{j-1}(x) = \text{Sum } c_i^{j-1} \ \phi_i^{j-1}(x) \qquad \text{for } i = 0, 1, \ldots, 2^j-1 \qquad [21]$$

$$G^{j-1}(x) = \text{Sum } d_i^{j-1} \ \varphi_i^{j-1}(x) \qquad \text{for } i = 0, 1, \ldots, 2^j-1 \qquad [22]$$

Further decomposition can be done recursively using equation [20]. The coefficient c_i^j is called "moving average". It can be further decomposed into c_i^{j-1} and d_i^{j-1} until it reaches the V^0 space. The coefficient d_i^j is called "weight" or "difference" which can be computed directly. The method for the computation of the coefficients is detailed in Chui (1992).

When we apply the wavelet transform to an image, we can treat the image pixel values as the coefficients. Since an image is a two dimensional data set and the wavelet transform is separable, we need to perform two transforms. First, we can perform the horizontal transform on pixel values in each row of the image. Next, we perform the vertical transform on pixel values in each column of the image. There are two procedures by which we can perform the two dimensional wavelet transforms. The first procedure is to perform the transform for the first row, then perform the transform for the first column,

then go to the second row and the second column. The procedure is repeated until the last row and last column are reached. The second procedure is to perform the transforms for all the rows, and then perform the transforms for all the columns.

The wavelet transform will result in two types of information, the micro and the macro information. The micro information is represented by pixel values that are close to 0 in the transformed image. These values represent the micro changes in the image. In similarity analysis with wavelets, we would not consider these micro changes since we would like to focus on the macro, or overall information in the image. Therefore, the values that are close to 0 will be set to 0 using appropriate threshold values. The retrieval values represent the dominant spectral properties of the image.

The transformed wavelet image is divided into non-overlapping 16x16 tiles in order to introduce spatial information after the wavelet transformation. For each tile, the number of coefficients with positive values and the number with negative values are counted. Therefore, the total number of distinct components for the wavelet feature is fixed to 512 (16 x 16 x 2).

4.1 Re-sampling

Note that the image size used for the wavelet transform needs to be of the order of 2^k (k = 0, 1, 2, …). However, this would not be true in practice. Therefore, before the wavelet transformation is applied, we need to check whether or not the any adjustment of image size is needed. If the image size is not in the order of 2^k, we have to adjust it into the order of 2^k.

The re-sampling method is used to adjust the image size to the order of 2^k if necessary. The re-sampling method works as follows:

1. check if the number of rows of the image is in the order of 2^k. If yes, then re-sampling for rows is not necessary.
2. if not, compute the following ratio:

$$\text{Ratio} = NS / NT \qquad [23]$$

where NS is the number of rows (or columns) for the original image, NT is the number of rows (or columns) for the re-sampled image (goal) with dimensions of the order of 2^k, where k is determined as follows:

for NS such that $2^{kk} < NS < 2^{kk+1}$

$$k = \begin{cases} kk & \text{if } |NS - 2^{kk}| < |NS - 2^{kk+1}| \\ kk+1 & \text{if } |NS - 2^{kk}| > |NS - 2^{kk+1}| \end{cases} \qquad [24]$$

3. determine the new pixel location with the following equation:

$$i_{res} = (\text{int})\ (\text{Ratio} * i_{org} + 0.5) \qquad i_{res} = 0, 1, \ldots, NT-1 \qquad [25]$$

$$i_{org} = 0, 1, \ldots, NS-1$$

where i_{res}: the pixel location after resampling (for the resampled image)

i_{org}: the pixel location before resampling (for the original image)

if Ratio > 1, some pixels from the original image will be repeated in the resampled image. Otherwise, if Ratio < 1, some pixels from the original image will be removed from the resampled image.

4. check if the number of columns of the image is of the order of 2^k. If yes, re-sampling for columns is not necessary. Otherwise, repeat steps 2 and3 for the columns.

Weight adjustment

In our study, the initial weight for each feature can be adjusted during feature extraction. Weight adjustment amongst different features during feature extraction is based on the values of C, D, V, and b, that are defined in equation [1]. By using different values of D and V, different features extracted will have a different number of distinct components. Therefore, the weights for different features will be also different. For example, for an 8-bit gray scale image, if we set D = 3, V = 8, then the number of distinct components of the image histogram, gradient, edge, and wavelet features will be:

Image histogram: $N_{pat} = D^2 * 2^b / V = 9 * 32 = 288$

Slope features: $SN_{pat} = D^2 * 2^b / V + 17 * D^2 = 288 + 153 = 441$

Edge features: $EN_{pat} = 4 * D^2 = 4 * 9 = 36$

Wavelet features: $WN_{pat} = 512$

If all the feature components are weighted uniformly, the weights for image histogram, slope, edge, and wavelet features will be 0.23, 0.34, 0.03, 0.40 respectively. Table 1 shows some examples of the different weights with the values of D and V with 8-bit gray scale images.

Weights can also be adjusted during the training with the ANN. The artificial neural network itself contains trained weight matrices to merge the heterogeneous features. To train the ANN and find the weights, different features are fed into the system. Once the network is trained, the features will have the proper weights so they can be used in merging heterogeneous features. In this approach, users do not need to worry about assigning weights to features. The final weights are actually the automatic combination of the weights assigned during feature extraction and those during network training.

Table 1. The change of initial weights with different values of D and V

		Weights			
V	D	image histogram	Slopes features	Edges	wavelets
8	3	0.23	0.34	0.03	0.40
8	4	0.27	0.42	0.04	0.27
8	8	0.34	0.53	0.04	0.09
16	4	0.19	0.39	0.05	0.37
16	5	0.22	0.44	0.06	0.28

Examination of correlation coefficients between features

Although the different features are extracted with different computation methods, these features may or may not be related. In order to examine the relations between the features, the correlation coefficients between different features are computed in our study. If the correlation coefficient is high between two feature classes, a method of removing the redundancy between different types of features may be needed. Otherwise, the features extracted from different feature extraction methods can directly used for similarity analysis.

Similarity matching with an artificial neural network

Artificial neural network mimics the brain's own problem solving process. An ANN can take previously solved examples (as knowledge) to build a system of interconnected "neurons" that makes new decisions, classifications, and forecasts. For an unknown problem, an ANN takes the data for the problem, trains on the data, and learns from the training process to obtain the necessary knowledge about the problem and builds the ANN for the problem.

In our study, an ANN technology is used for the similarity matching. The advantages for using an ANN in similarity matching are twofold, one that we can combine multiple features extracted with different methods, and the second that the combination of these features can be non-linear. The following discussion includes the basic architecture of an ANN and the design in our similarity matching.

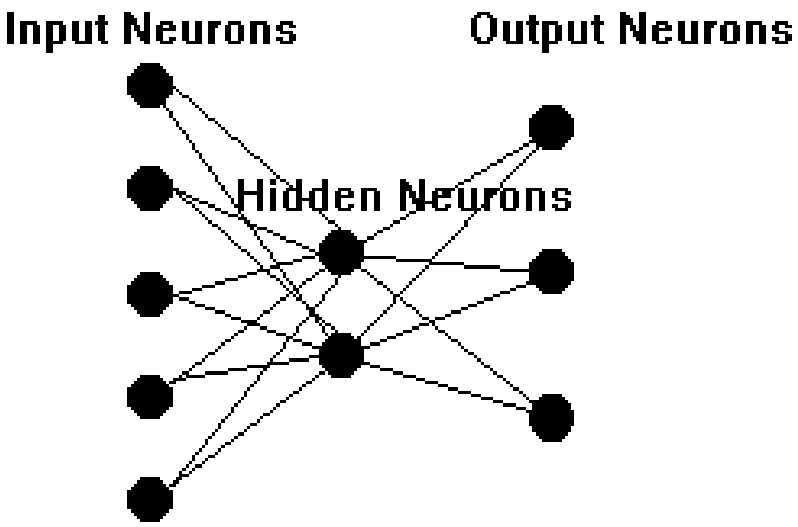


Figure 1. The basic architecture of an artificial neural network

A typical ANN is a Backpropagation network (Ward System Group, Inc. 1996). The architecture of a Backpropagation network usually has an input layer, one or more hidden layers, and an output layer of neurons (Figure 1). Input values from the input layer are weighted and passed to the hidden layer(s). Neurons in the hidden layer(s) "fire" or produce outputs that are based upon the sum of weighted values passed to them. The hidden layer(s) pass(es) values to the output layer in the same fashion. The output layer produces the results. In our study, the Ward network (Ward System Group, Inc. 1996) is used. The Ward network consists of three different Backpropagation network architectures with multiple hidden layers (Figure 2). In the Ward network, different activation functions applied to hidden layer slabs (a group of neurons) detect different features in the patterns processed through a network. The output from the network considers the combination of these features to produce a better result. The Ward network will also determine the optimal number of hidden neurons for each hidden layer slab

based on the number of input variables, the number of output variables, and the number of patterns within the training set.

In an ANN system, the patterns input to the network are typically divided into three sets, the training set, the test set, and the production set. These three data sets are extracted from the original input data patterns. There is no overlap amongst the three data sets, thus the following condition is always satisfied:

$$N_{tin} = N_{trn} + N_{tst} + N_{pat} \qquad [26]$$

where N_{trn} is the number of patterns in the training set, N_{tst} is the number of patterns in the test set, N_{pat} is the number of patterns in the production set, and N_{tin} is the number of total input patterns.

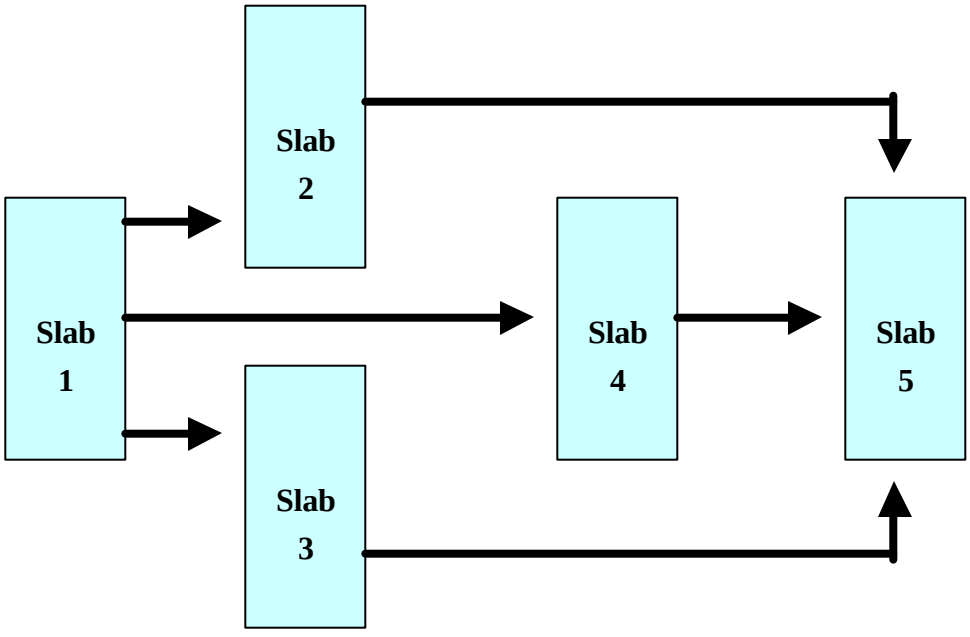


Figure 2. One architecture of the Ward neural network

The training data set is used to train the network. Through the training process, the ANN model is built. The test data set is used to test how well the ANN model has been developed with the training data. Both the training data set and the test data set are used for model development. The production data set is used to evaluate the developed model. Therefore, for similar problems, the developed model can be directly applied by presenting the new data patterns to the developed model and producing the required results.

There are two reasons for using an ANN in our study. First, the ANN is used to create a non-linear combination of features by automatically adjusting the weights for different features. The second reason is to generate the appropriate criteria for the similarity matching of video shots. In this study, the input data patterns to the ANN are the features generated from the feature extraction process. These data are divided into two sets, the training set and the test set. In order to have a good number of data patterns for both the training data set and test data set (with slightly more data for model training), the training data set and the test dataset are divided with a population percentage of 60 and 40 respectively. The production set is not created from the input data since the developed model is not used in any further application. The basic procedure for similarity matching is as follows:

- import data into the ANN.
- define the input and output layers: set the feature data extracted from the target video shot as input, set the feature data from the query video shots as outputs.

- randomly extract the training data set and the test data set for the all feature data: the number of patterns in training data set consists of 60% of the total data patterns, and the remaining 40% are used for the test data set.
- design the ANN: the Ward network (Ward System Group, Inc.) is selected. The number of hidden neurons in each backpropagation system is automatically determined by the Ward network based on the input, output, and the size of the training data set of the system.
- train the system (learning) and develop the ANN model.
- produce results: coefficients of determination and mean absolute errors between the features of the target video shot and the features of the query video shots are computed using the ANN model.

Criteria for similarity ranking

In our study, two criteria are used to examine the similarity, the coefficient of determination (R2) and the mean absolute error (MAE). The coefficient of determination is a statistical indicator that compares the accuracy of the model to the accuracy of a trivial benchmark model. A perfect fit between the models would result in an R2 value of 1, a very good fit near 1, and a very poor fit near 0. The computation of R2 is as follows:

$$R2 = 1 - S\,(y - y_{pre})^2 \,/\, S\,(y - y_{avg})^2 \qquad [27]$$

where y is the actual value, y_{pre} is the predicted value of y, and y_{avg} is the mean of the y values.

The correlation coefficient is the square root of the coefficient of determination (R2), and computed as follows:

$$r = +/- \text{ sqrt}(R2) \quad [28]$$

A very good fit would result in an r value close to 1 or -1, implying a highly positive or highly negative relation between the input and the output. On the hand, a poor fit would result in an r value close to 0, implying a poor relation between the input and the output.

The mean absolute error (MAE) is a statistic difference between the values of the actual outputs and the predicted outputs computed by the network. The computation of the MAE is as follows:

$$MAE = \text{sqrt} (S (y - y_{pre})^2 / (N-1)) \quad [29]$$

where y and y_{pre} have the same meaning as in equation [27], and N the total number of data samples.

CHAPTER 4

MOSAIC BASED SIMILARITY ANALYSIS

Some video shots, such as pan video shots, contain many views of the same scene taken over time, either from a moving or a sweeping camera. The scene itself contains the common information within all frames in the video shot. However, the information is implicitly distributed over frames with a very high redundancy. The mosaic representation transforms the video shots from a sequential frame-based representation into a scene-based representation to which every frame can be directly related. This representation allows direct and immediate access to the scene information, such as static locations and dynamically moving objects. It also eliminates the redundancy between frames and results in a highly efficient and compact representation of the video information (VideoBrush Corporation 2000).

The mosaic based similarity analysis is based on the following assumption: if the two video shots are similar, the mosaic scenes created from them are also similar. The converse is also true. Therefore, the similarity analysis between video shots is considered to be equivalent to similarity analysis between their mosaic scenes. The procedure for mosaic based similarity analysis is as follows:

- create mosaic scenes for the target video shot and the query video shots
- extract features from these mosaic scenes
- perform similarity matching with an ANN

- rank the query video shots based on the values of the coefficients of determination and/or the mean absolute error between the target video shot and each of the query video shots

Data preprocessing

Different pan video shots are collected in this study. After the target video shot is selected, four more video shots are created from the target video shot. The purpose of doing this is to create some video shots that are more or less similar to the target video shot. These created video shots plus other irrelevant video shots consist of a group of query video shots. Common sense suggests that the query video shots created from the target video shot should generally be more similar to the target video shot than other irrelevant video shots. The mosaic images are further created for all video shots for the purpose of feature extraction and similarity matching. The creation of the four new video shots is done as follows:

- from the target video shot, cut the first 1/3 portion and keep the rest to create the first new video shot (the name of the video shot will end with ‘se’)
- from the target video shot, cut the last 1/3 portion and keep the rest to create the second new video shot (the name of the video shot will end with ‘sb’)
- from the target video shot, cut the first and last 1/6 portion and keep the rest to create the third new video shot (the name of the video shot will end with ‘md’)
- from the target video shot, change the frame size and create a fourth new video shot (the name of the video shot will end with ‘sm’)

Figure 3 shows the organization of the video shots that are used for the mosaic based similarity analysis.

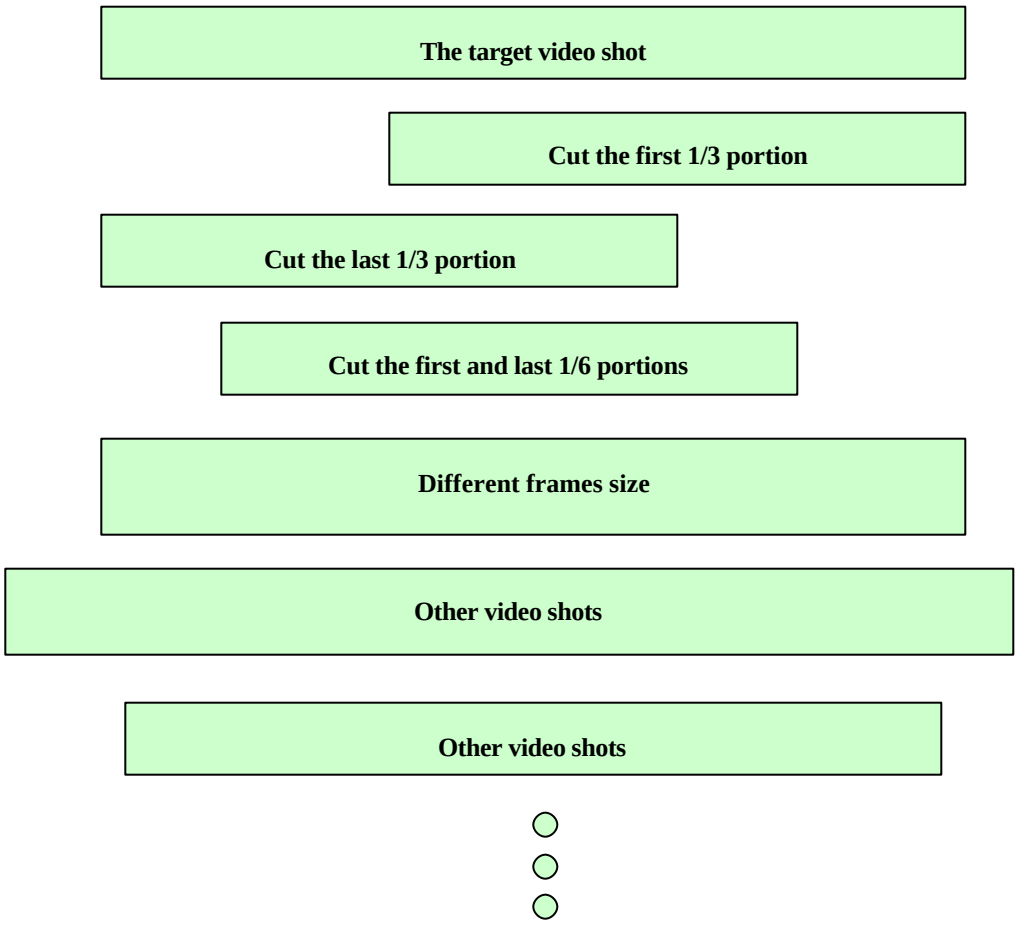


Figure 3. Organization of video shots for mosaic based similarity analysis

Results and discussion

Three groups of data sets are prepared. For each group, the mosaic images generated from the original video shots are shown in Appendix A. In group 1, the target

video shot is the tour of the interior area of a house. From this video shot, four shots are generated. The target and the generated video shots, with the other unrelated video shots, consist of the data set for the group.

In Group 2, the target video shot is about the landing of an unmanned plane to the ground. Four video shots are also generated from this video shot. The motion information of the unmanned plane is recorded on their mosaic images with the moving track of the unmanned plane (Appendix A, Figures b and d). These four generated video shots, with the other unrelated video shots, consist of the data set for group 2.

In group 3, the target video shot is a pan video of a coastal area with children playing. Four video shots are also generated from this video shot. These along with the other unrelated video shots, consist of the data set for group 3.

For the above three groups of video shots, we would like to examine if the video shots that are generated from the target video shot exhibit a higher degree of similarity to the target video shot compared to the other video shots. For the video shots that are not generated from the target video shots, we would like to investigate the variation of the degree of similarities with the contents of the features extracted from these video shots. Since a color video shot will generate three panoramic images, representing the red, green, and blue bands, respectively, this will lead to large volume of data. In order to reduce the data volume, the average of the three color bands is used. The similarity analysis with the R, G, B bands considered separately is also performed to compare with the results using the average of the R, G, B bands.

The results of similarity matching and subsequent discussion are presented in the following sections. In the first section, the results with the three color bands used

separately and with the average of the three color bands are compared. The next section will focus on the results using individual features (in isolation) and various feature combinations. The weight adjustment for the feature combination will be discussed. Finally, the similarity criteria based on the coefficient of determination and the mean absolute errors are also discussed.

1. Substitution of the average image intensity for the individual R, G, B intensity values

Table 2 shows the similarity matching results using a similarity criterion based on the coefficients of determination with a single feature and the feature combination with the individual values in the R, G, and B bands. Table 3 shows the similarity matching results using a similarity criterion based on the coefficients of determination with a single feature and the feature combination with the average R, G, B intensity values. From these tables, we can find that the rank order obtained by using the average R, G, B value is almost the same as the rank order obtained by using the individual R, G, B values separately. If a different order is obtained, it differs between two successive neighbors, where the coefficients of determination are very close in value. The difference in similarity matching results from both data sets is further examined by a paired t test. The result of the t test shows that similarity matching using the average R, G, B intensity values is not significantly different from the similarity matching results using the individual values in R, G, B bands under the 95% confidence level at the p value < 0.0001. Therefore, there is essentially no difference between the results from the individual R, G, B values and the results from the average value of the three bands. In the

future analysis, we use the average image intensity value of the R, G, B bands to reduce the data volume as well as the processing time.

2. Examination of correlation coefficients between features

Although the methods for features extraction are different, these methods can not guarantee that the features extracted are not related. On account of this, before we use the features for similarity matching, we may need to examine the correlations amongst these features. If the high correlation values are exhibited, removing the redundancy amongst the features may be necessary. Table 4 shows the correlation coefficients between feature classes for a number of video shots. From Table 4, we can see that the correlation coefficients between different types of features are very small. That is, the features extracted are unlikely to be related. Therefore, the extracted features are directly used in similarity analysis without having to de-correlate them.

3. Similarity matching with individual features and with a feature combination

Table 5 shows the results obtained by using an individual feature and feature combination. The query videos are ranked based on their coefficients of determination. In general, the query video shots that are generated from the target video shot show higher similarity than other irrelevant video shots. This is true for both individual features and feature combinations. Comparing with the results in the case of individual features, the ranked similarity orders are different with different features used, especially in the case of video shots that are not generated from the target video shots. The image histogram features show the pixel intensity distribution property of the image. In addition, the

similarity analysis with edge feature is more sensitive than others. The main reason for this is that we set relatively high thresholds for the Canny edge detector, and therefore the edge feature is basically the most significant local change. On the contrary, the similarity analysis with slopes and slope directions and with wavelets is less sensitive. The slope features mainly reflect the information about the pixel local variance in detail, where as the wavelets mainly reflect the global information with the details removed.

The non-linear feature combination is done by the ANN model by exploiting its capability for automatic weight adjustment. The initial weight setting also has a certain effect on the result (this will be discussed in the next section). The similarity analysis with feature combination using ANN would be a better choice than that with a single feature.

4. Similarity matching with weight adjustment

Although the ANN automatically adjusts the weights for different features, the results may also be affected by the initial weight setting. As we discussed before, the initial weight adjustment is based on the adjustment of the values of V and D. In this study, two sets of weight adjustment are used to examine the effects on the different weight combinations. The values of V and D in the first set are set to be 8 and 4, respectively. In the second set, both values are set to be 8. The weights for each feature class in the two sets are shown as follows:

Set 1: histograms : slopes features : edges : wavelets = 0.27 : 0.42 : 0.04 : 0.27

Set 2: histograms : slopes features : edges : wavelets = 0.34 : 0.53 : 0.04 : 0.09

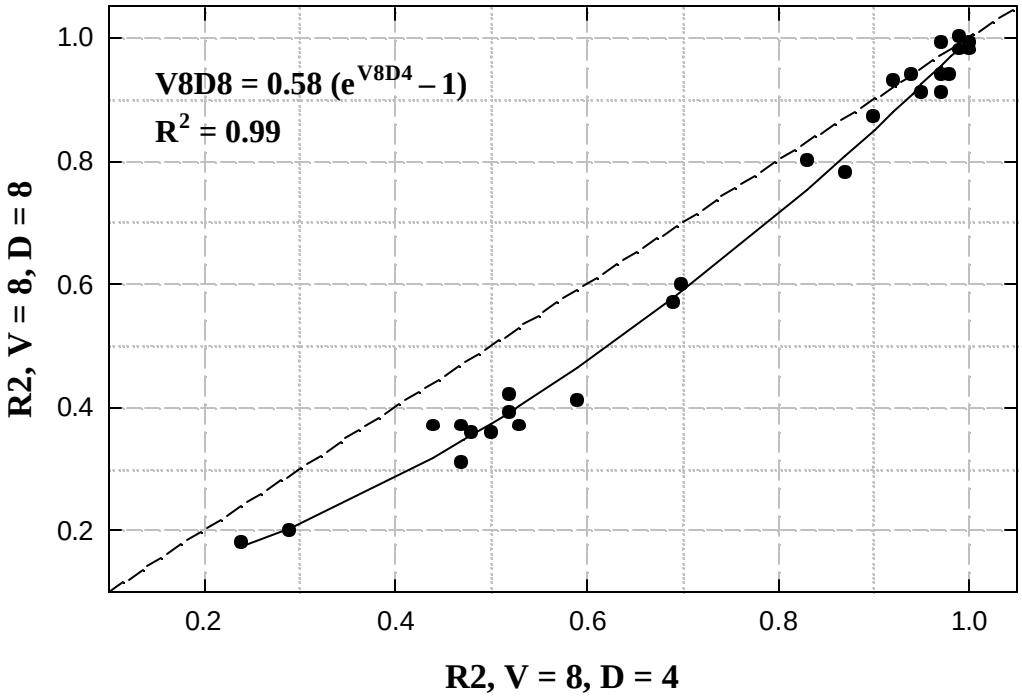


Figure 4. Effect of initial weight assignment

Table 6 shows the results with different initial weight combinations. For video shots that are generated from the target video shots, there is no difference in the similarity rank order with different initial weights. The main reason for this is that the similarity rank orders with the individual features are almost exactly the same as that with the feature combination (Table 3). For the query video shots that are not generated from the target video shot, the rank orders are a little different with different initial weight assignments. This means that the initial weight combination affects the similarity rank order. However, the effect is small. Further analysis shows that the results with different

initial weights exhibit good consistency but the relation is not linear (Figure 4.). The results of this experiment show that with ANN based similarity matching the initial weight assignment may not be big issue. The ANN will automatically adjust the weights to their optimal values.

5. Similarity matching using coefficients of determination and mean absolute errors

The similarity rank order using coefficients of determination is based on the following assumption: if the coefficient of determination between the target video shot and the query video shot is high, the similarity between them is also high; otherwise, if the coefficient of determination between the target video shot and the query video shot is low, the dissimilarity between them is high. The rank order with mean absolute errors is based on the reverse assumption, that is, if the mean absolute error between the target video shot and the query video shot is small, the similarity between them is high; otherwise, if the mean absolute error between the target video shot and the query video shot is large, the dissimilarity between them is high.

Table 7 shows the results of similarity matching with coefficients of determination and with mean absolute errors. In general, both results show that the query video shots generated from the target video shot exhibit a higher similarity than the other query video shots. This means that the mean absolute error can also be used as the criterion for the similarity analysis. However, the actual rank order based on the mean absolute errors is different from the rank order based on the coefficients of determination, whether or not the query video shots are generated from the target video shot. From Table 7, we also find that the similarity results are more consistent with the coefficients of

determination than with the mean absolute errors. The similarity results resulting from the mean absolute errors also vary in a large range. In some cases, the similarity between the target video shot and the video shots generated from the target video shot is lower than that between the target video shot and the video shots not generated from the target video shot (Table 7). Thus, the coefficient of determination appears to be the better metric than the mean absolute error for the purpose of similarity analysis.

Table 2. The similarity matching results using the individual R, G, B values

House	**house**	**Housesm**	**housese**	**Housemd**	**housesb**	**touch**	**land**	**whitebd**	**Sea**
Feature combination	0.98	0.98	0.96	0.96	0.86	0.69	0.52	0.52	0.51
Image histogram	0.99	0.99	0.96	0.95	0.85	0.60	0.42	0.06	0.27
Edge direction	1.00	0.99	0.89	0.93	0.44	0.30	0.11	0.21	0.08
Slope feature	0.99	0.99	0.97	0.97	0.86	0.74	0.54	0.74	0.66
Wavelets	1.00	0.96	0.94	0.97	0.79	0.79	0.66	0.79	0.68

Land	**land**	**Landsm**	**landse**	**Landmd**	**landsb**	**touch**	**whitebd**	**house**	**Sea**
Feature combination	1.00	1.00	0.98	0.95	0.71	0.52	0.47	0.43	0.28
Image histogram	0.99	0.99	0.95	0.91	0.55	0.38	0.08	0.20	0.04
Edge direction	1.00	0.99	0.97	0.91	0.75	0.33	0.08	0.09	0.28
Slope feature	1.00	1.00	0.99	0.96	0.76	0.57	0.66	0.52	0.41
Wavelets	1.00	0.96	0.91	0.92	0.75	0.63	0.67	0.59	0.61

Sea	**sea**	**Seamd**	**seasm**	**Sease**	**seasb**	**house**	**whitebd**	**touch**	**Land**
Feature combination	1.00	0.98	0.97	0.94	0.90	0.62	0.48	0.48	0.24
Image histogram	1.00	0.96	0.95	0.89	0.77	0.40	0.16	0.12	0.04
Edge direction	1.00	0.98	0.97	0.93	0.90	0.12	0.12	0.41	0.30
Slope feature	0.98	0.95	0.96	0.93	0.91	0.74	0.63	0.69	0.35
Wavelets	1.00	0.95	0.94	0.93	0.91	0.70	0.61	0.73	0.68

Table 3. The similarity matching results using the average R, G, B values

House	**house**	**housesm**	**housese**	**Housemd**	**housesb**	**touch**	**land**	**whitebd**	**sea**
Feature combination	0.99	0.99	0.97	0.97	0.87	0.69	0.52	0.53	0.50
Image histogram	0.98	0.97	0.93	0.93	0.83	0.58	0.38	0.04	0.25
Edge direction	1.00	0.99	0.85	0.92	0.41	0.24	0.00	0.19	0.01
Slope feature	1.00	0.99	0.98	0.98	0.88	0.76	0.57	0.76	0.65
Wavelets	1.00	0.96	0.94	0.97	0.79	0.78	0.66	0.78	0.67

Land	**land**	**landsm**	**landse**	**Landmd**	**landsb**	**touch**	**whitebd**	**house**	**sea**
Feature combination	1.00	1.00	0.98	0.95	0.70	0.52	0.48	0.44	0.29
Image histogram	0.99	0.98	0.95	0.89	0.53	0.38	0.08	0.19	0.04
Edge direction	0.97	0.97	0.94	0.87	0.59	0.18	0.00	0.00	0.20
Slope feature	1.00	1.00	0.99	0.96	0.77	0.56	0.71	0.53	0.39
Wavelets	1.00	0.95	0.91	0.92	0.75	0.63	0.66	0.59	0.60

Sea	**sea**	**seasm**	**seamd**	**Sease**	**seasb**	**house**	**whitebd**	**touch**	**land**
Feature combination	0.97	0.94	0.92	0.90	0.83	0.59	0.47	0.47	0.24
Image histogram	1.00	0.95	0.95	0.88	0.77	0.38	0.23	0.09	0.04
Edge direction	0.99	0.96	0.94	0.91	0.87	0.00	0.05	0.35	0.18
Slope feature	1.00	0.98	0.98	0.96	0.96	0.75	0.61	0.69	0.35
Wavelets	1.00	0.95	0.95	0.93	0.91	0.69	0.62	0.73	0.68

Table 4. The correlation coefficients between feature classes

	Image histogram	**Edge direction**	**Slope**	**Slope Direction**	**Wavelets**
Image histogram	1	-0.003 ~ 0.044	-0.228 ~ 0.177	-0.160 ~ 0.141	-0.004 ~ 0.003
Edge direction		1	-0.049 ~ 0.000	-0.020 ~ 0.006	-0.001 ~ 0.004
Slope			1	0.098 ~ 0.342	-0.003 ~ 0.004
Slope Direction				1	-0.003 ~ 0.002
Wavelets					1

Table 5. Comparison of similarity rank order using individual feature and a feature combination

house	**house**	**Housesm**	**housemd**	**housese**	**Housesb**	**Touch**	**whitebd**	**land**	**Sea**
Feature combination	0.99	0.99	0.97	0.97	0.87	0.69	0.53	0.52	0.50
Image histogram	0.98	0.97	0.93	0.93	0.83	0.58	0.04	0.38	0.25
Edge direction	1.00	0.99	0.92	0.85	0.41	0.24	0.19	0.00	0.01
Slope feature	1.00	0.99	0.98	0.98	0.88	0.76	0.76	0.57	0.65
Wavelets	1.00	0.96	0.97	0.94	0.79	0.78	0.78	0.66	0.67

Land	**land**	**Landsm**	**landse**	**landmd**	**Landsb**	**Touch**	**whitebd**	**house**	**Sea**
Feature combination	1.00	1.00	0.98	0.95	0.70	0.52	0.48	0.44	0.29
Image histogram	0.99	0.98	0.95	0.89	0.53	0.38	0.08	0.19	0.04
Edge direction	0.97	0.97	0.94	0.87	0.59	0.18	0.00	0.00	0.20
Slope feature	1.00	1.00	0.99	0.96	0.77	0.56	0.71	0.53	0.39
Wavelets	1.00	0.95	0.91	0.92	0.75	0.63	0.66	0.59	0.60

Sea	**sea**	**Seasm**	**seamd**	**sease**	**Seasb**	**House**	**whitebd**	**touch**	**Land**
Feature combination	0.97	0.94	0.92	0.90	0.83	0.59	0.47	0.47	0.24
Image histogram	1.00	0.95	0.95	0.88	0.77	0.38	0.23	0.09	0.04
Edge direction	0.99	0.94	0.96	0.91	0.87	0.00	0.05	0.35	0.18
Slope feature	1.00	0.98	0.98	0.96	0.96	0.75	0.61	0.69	0.35
Wavelets	1.00	0.95	0.95	0.93	0.91	0.69	0.62	0.73	0.68

Table 6. Effects of initial weight assignment on the similarity rank order

House	**house**	**housesm**	**housemd**	**housese**	**Housesb**	**Touch**	**Whitebd**	**land**	**sea**
V = 8, D = 4	0.99	0.99	0.97	0.97	0.87	0.69	0.53	0.52	0.50
V = 8, D = 8	1.00	0.98	0.94	0.91	0.78	0.57	0.37	0.42	0.36

Land	**land**	**landsm**	**landse**	**landmd**	**Landsb**	**Touch**	**Whitebd**	**house**	**sea**
V = 8, D = 4	1.00	1.00	0.98	0.95	0.70	0.52	0.48	0.44	0.29
V = 8, D = 8	0.99	0.98	0.94	0.91	0.60	0.39	0.36	0.37	0.20

Sea	**sea**	**seasm**	**seamd**	**sease**	**Seasb**	**House**	**Whitebd**	**touch**	**land**
V = 8, D = 4	0.97	0.94	0.92	0.90	0.83	0.59	0.47	0.47	0.24
V = 8, D = 8	0.99	0.94	0.93	0.87	0.80	0.41	0.31	0.37	0.18

Table 7. A comparison of similarity analysis results using coefficients of determination and mean absolute errors with feature combination (MAE: Mean Absolute Errors)

House	**house**	**housesm**	**housemd**	**housese**	**Housesb**	**Touch**	**whitebd**	**Land**	**sea**
R^2 with V=8, D=4	0.99	0.99	0.97	0.97	0.87	0.69	0.53	0.52	0.50
MAE with V=8, D=4	160.8	90.4	160.5	158.1	191.4	533.8	344.1	1397.7	449.2
R^2 with V=8, D=8	1.00	0.98	0.94	0.91	0.78	0.57	0.37	0.42	0.36
MAE with V=8, D=8	26.4	25.5	53.2	59.6	83.7	192.5	144.8	560.0	134.3

land	**land**	**landsm**	**landse**	**landmd**	**Landsb**	**touch**	**whitebd**	**House**	**sea**
R^2 with V=8, D=4	1.00	1.00	0.98	0.95	0.70	0.52	0.48	0.44	0.29
MAE with V=8, D=4	246.7	139.2	335.1	312.8	410.6	662.0	395.2	597.7	447.0
R^2 with V=8, D=8	0.99	0.98	0.94	0.91	0.60	0.39	0.36	0.37	0.20
MAE with V=8, D=8	102.1	66.7	157.4	135.0	141.3	246.1	120.6	210.7	150.7

sea	**sea**	**seasm**	**seamd**	**sease**	**Seasb**	**house**	**whitebd**	**Touch**	**land**
R^2 with V=8, D=4	0.97	0.94	0.92	0.90	0.83	0.59	0.47	0.47	0.24
MAE with V=8, D=4	117.9	81.5	135.9	144.7	148.6	446.1	357.5	664.9	1767.9
R^2 with V=8, D=8	0.99	0.94	0.93	0.87	0.80	0.41	0.31	0.37	0.18
MAE with V=8, D=8	17.8	24.7	28.6	47.4	42.2	199.0	136.4	246.2	683.9

CHAPTER 5

KEY FRAME BASED SIMILARITY ANALYSIS

Not all video shots can be represented by their mosaic images. Zoom (zoom-in or zoom-out) video shots are example of such image types. In order to examine the similarity for such video shots, a key frame based similarity analysis method is proposed in this study. Key frames are still images that best represent the contents of the video shot in an abstracted manner. Key frames are usually extracted from the original video data. In this study, the key frames we used are extracted from the original video shot. The extraction of key frames is based on the uniform scheme, described as follows: first, the first and the last frames are always considered as key frames, and remaining key frames are extracted between the first and the last frames based on video content. The key frames extracted are ordered based on their temporal order within the video shot.

The key frame based similarity analysis is based on the assumption that the similarity or dissimilarity between video shots will also be reflected in the similarity or dissimilarity between their key frames. Therefore, the similarity analysis between video shots will amount to examining the similarity between their key frames in the temporal order. The basic procedure for key frame based similarity analysis is as follows:

- extract key frames from the target and the query video shots, and order the key frames of each video shot in the order in which they appear in the video shot
- extract features from each key frame

- perform similarity matching with an ANN
- rank the query video shots based on the coefficients of determination and/or the mean absolute errors.

Data preprocessing

In addition to the video shots used in mosaic based similarity analysis, some more video shots were collected. There are three data sets prepared for the key frame based similarity analysis. Appendix B lists their key frames of these three data sets with three key frames for each video shot. The first data set consists of three video shots of flooding scenes. These three video shots are from different locations. Each flooding video shot is relatively static, that is, focusing on one location. Amongst the three flooding video shots, the scenes from two video shots share some relatively similar structure. Both of them have some sky area on the top, houses and other objects in the middle, and flood water at the bottom of the scene. However, the relative proportions of sky, houses and other objects, and flood water in the scenes are different. These two flooding video shots are separately selected as the target video shots. In addition, two new video shots were generated from each of the two flooding video shots, by dividing the flooding video shot into the first half and the last half (Figure 5). The third flooding video shot consists of just the flood water. The new video shots, plus the third flooding video shot and other unrelated video shots, consist of the two groups of data for the similarity analysis.

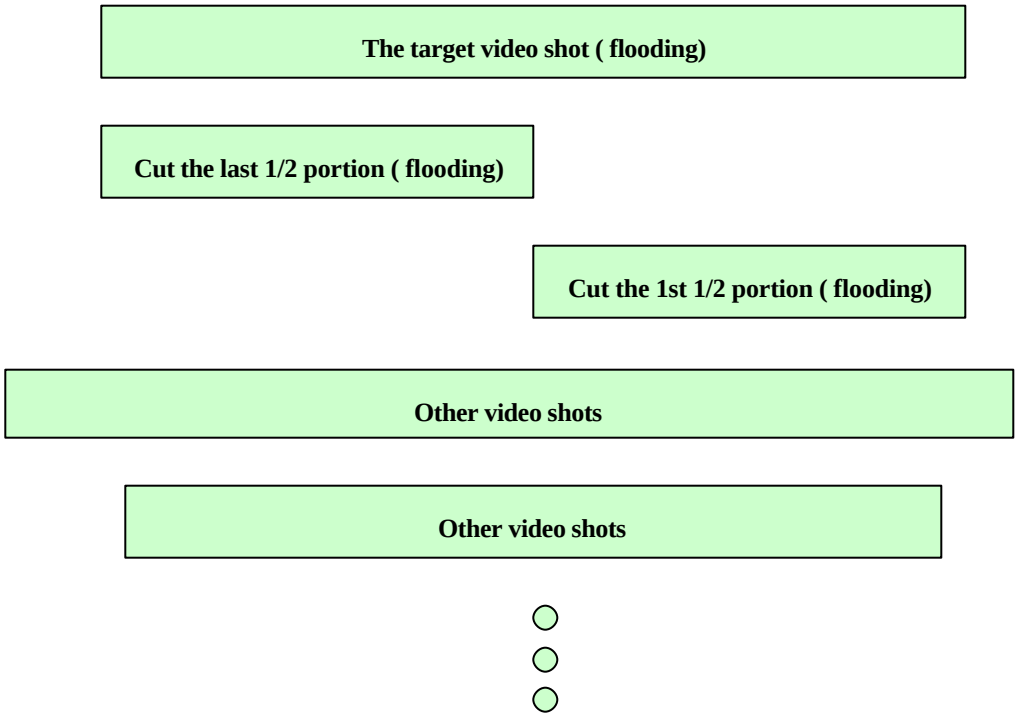


Figure 5. The organization of the flooding video data

The second data set, consisting of the pan video shots, is the same as the data sets used in mosaic based similarity analysis. Four video shots generated from the target video shot, in addition to the other video shots consist of a group of video data (Figure 3).

The third data set consists of number of zoom video shots with six zoom-in shots and one zoom-out shot. The zoom-in video shots begin with the image of the whole earth, then zooms into a detailed specific location. One of these zoom-in video shots is selected as the target video shot, where as the remaining zoom-in video shots, the video shot with the zoom-out and other non-zoom video shots consist of the query video shots.

Results and discussion

The results and discussion of similarity matching for frame based on similarity analysis will be organized into the following sections. The first section will present and discuss the results with the flooding video data sets. The adjustment of weights is also discussed in this section. We then present and discuss the results on a data set similar to the one used in the mosaic based similarity analysis. Next, the results of the zoom-in data set is presented and discussed. The effect of key frame selection is discussed. The difference between the similarity criteria based on coefficients of determination and mean absolute errors is also discussed. Finally, a brief summary is presented for the three different types of data sets.

1 Similarity analysis of video shots with static structure and similar scene objects and scene structure

Table 8 shows the similarity matching results with individual features and with a feature combination for video shots with similar scene objects and static scene structure. The video shots flood1p1 and flood1p2 are generated from the video shot flood1. Similarly, the video shots flood2p1 and flood2p2 are generated from the video shot flood2. Both flood1 and flood2 are video shots of relatively static scenes (the camera focuses on one location without zooming) with similar object composition and scene structure. The results show that the query video shots that are generated from the target video shot exhibit higher similarity than the others. The video shots flood1 and flood2 also exhibit higher similarity than other video shots since they share a similar object

composition and scene structure. Flood3 generally exhibits the highest similarity to the target video shot amongst the remaining video shots due to the similarity of the scenes containing flood water. Both, individual features and feature combinations yield a similarity rank order. However, these is a difference in the results using a feature combination and using individual features in isolation. In the case of the feature combination, the query video shots generated from the target video shot exhibit higher similarity than other query video shots. Among the remaining video shots, the similarity between flood1 and flood2 is higher than other video shots, and the similarity between flood1 (or flood2) and flood3 is the highest amongst the rest of video shots. In the case of individual features, the query video shots generated from the target video shot also exhibit higher similarity than other video shots. However, the similarity between flood1 and flood2 is not always higher than other video shots among the remaining video shots, and the similarity between flood1 (or flood2) and flood3 is also not the highest amongst the rest of the video shots. The above observations are true both for 3 and 5 key frames (Table 10).

The initial weight assignment scheme is also examined for one group of the video data set (Table 9). The same weight assignment scheme as in mosaic based similarity analysis is used, and similar observations are made as discussed in the mosaic based similarity analysis.

Table 10 shows the results for different number of key frames used to characterize the video shot, either with individual features or with a feature combination. The number of key frames used is 3 and 5, respectively. The difference in number of key frames does not result in a significant difference in the similarity rank order of the query video shots

(Table 10). The main reason for this may come from the relative static scene and the uniform key frame selection.

Table 11 shows the results of similarity analysis with coefficients of determination and mean absolute errors. The mean absolute errors (MAE) can also be used as the criteria for similarity ranking. However, the variation in the MAE is larger than the variation in the coefficients of determination. This is more significant when a single feature is used to examine similarity. This study shows that the coefficient of determination is a better indicator than the mean absolute error for the purpose of ranking video based on similarity.

2. Key frame based similarity analysis for pan video shots

We have discussed before that the similarity analysis of pan video can be performed by first creating their mosaic images and then measuring the similarity between the mosaic images. In this section, we will discuss the similarity analysis of pan video shots using key frames. Table 12 shows the results of key frame-based similarity analysis of pan video shots with either individual features or with a feature combination. In general, if the query video shots are generated from the target video shots, they will exhibit higher similarity to the target video shot than other video shots. This was observed in the case of the mosaic based similarity analysis. Table 13 shows the results of similarity analysis using the mean absolute error. As the before, the mean absolute error can also be used as the criterion to rank order of the query video shots, but is not as good as the coefficient of determination.

Table 8. Similarity analysis with individual features and with a feature combination for static video shots with overall similar scene objects and scene structure (V=8, D=4)

Flood1	**Flood1**	**flood1p2**	**flood1p1**	**flood2**	**Flood3**	**touch**	**house**	**land**	**sea**
Feature combination	0.99	0.97	0.97	0.68	0.44	0.39	0.34	0.33	0.26
Image histogram	1.00	0.96	0.96	0.38	0.18	0.07	0.00	0.00	0.03
Edge direction	0.99	0.79	0.85	0.02	0.14	0.13	0.01	0.14	0.16
Slope feature	0.99	0.97	0.97	0.88	0.65	0.65	0.68	0.67	0.55
wavelets	1.00	0.93	0.93	0.56	0.32	0.41	0.50	0.39	0.43

Flood2	**Flood2**	**flood2p2**	**flood2p1**	**flood1**	**Flood3**	**touch**	**land**	**house**	**sea**
Feature combination	1.00	0.95	0.94	0.67	0.49	0.48	0.44	0.42	0.34
Image histogram	1.00	0.90	0.86	0.33	0.15	0.17	0.00	0.00	0.09
Edge direction	0.98	0.78	0.86	0.02	0.33	0.02	0.09	0.00	0.10
Slope feature	1.00	0.97	0.98	0.84	0.80	0.71	0.78	0.73	0.61
wavelets	1.00	0.90	0.92	0.52	0.67	0.40	0.54	0.51	0.33

Table 9. Initial weight assignment for key frame based similarity analysis (feature combination)

flood1	**Flood1**	**flood1p2**	**flood1p1**	**flood2**	**flood3**	**Touch**	**House**	**land**	**sea**
V=8, D=4	0.99	0.97	0.97	0.68	0.44	0.39	0.34	0.33	0.26
V=8, D=8	1.00	0.96	0.94	0.54	0.36	0.32	0.25	0.25	0.21

Flood2	**Flood2**	**Flood2p2**	**Flood2p1**	**Flood1**	**flood3**	**Touch**	**Land**	**house**	**sea**
V=8, D=4	1.00	0.95	0.94	0.67	0.49	0.48	0.44	0.42	0.34
V=8, D=8	1.00	0.90	0.90	0.53	0.38	0.37	0.30	0.28	0.27

Table 10 (a) Results of similarity matching with different number of key frames (V=8, D=4, target video flood1.avi).

flood1	**flood1**	**flood1p2**	**flood1p1**	**flood2**	**Flood3**	**touch**	**house**	**land**	**sea**
Feature combination, 3 frames	0.99	0.97	0.97	0.68	0.44	0.39	0.34	0.33	0.26
Image histogram, 3 frames	1.00	0.96	0.96	0.38	0.18	0.07	0.00	0.00	0.03
Edge direction, 3 frames	0.99	0.79	0.85	0.02	0.14	0.13	0.01	0.14	0.16
Slope feature, 3 frames	0.99	0.97	0.97	0.88	0.65	0.65	0.68	0.67	0.55
Wavelets, 3 frames	1.00	0.93	0.93	0.56	0.32	0.41	0.50	0.39	0.43
Feature combination, 5 frames	0.99	0.96	0.96	0.68	0.44	0.43	0.35	0.37	0.28
Image histogram, 5 frames	1.00	0.95	0.96	0.39	0.17	0.12	0.00	0.00	0.04
Edge direction, 5 frames	0.99	0.81	0.85	0.08	0.13	0.09	0.00	0.20	0.19
Slope feature, 5 frames	1.00	0.97	0.96	0.88	0.65	0.67	0.64	0.70	0.58
Wavelets, 5 frames	1.00	0.91	0.93	0.56	0.32	0.40	0.43	0.43	0.47

Table 10 (b) Results of similarity matching with different number of key frames (V=8, D=4, target video flood2.avi).

flood2	**flood2**	**flood2p2**	**flood2p1**	**flood1**	**flood3**	**touch**	**land**	**house**	**sea**
Feature combination, 3 frames	1.00	0.95	0.94	0.67	0.49	0.48	0.44	0.42	0.34
Image histogram, 3 frames	1.00	0.90	0.86	0.33	0.15	0.17	0.00	0.00	0.09
Edge direction, 3 frames	0.98	0.78	0.86	0.02	0.33	0.02	0.09	0.00	0.10
Slope feature, 3 frames	1.00	0.97	0.98	0.84	0.80	0.71	0.78	0.73	0.61
Wavelets, 3 frames	1.00	0.90	0.92	0.52	0.67	0.40	0.54	0.51	0.33
Feature combination, 5 frames	1.00	0.95	0.93	0.66	0.48	0.47	0.44	0.39	0.32
Image histogram, 5 frames	1.00	0.89	0.86	0.32	0.16	0.16	0.00	0.01	0.10
Edge direction, 5 frames	0.98	0.78	0.80	0.05	0.35	0.01	0.10	0.00	0.12
Slope feature, 5 frames	1.00	0.97	0.97	0.84	0.79	0.72	0.77	0.67	0.61
Wavelets, 5 frames	0.99	0.89	0.90	0.52	0.63	0.46	0.55	0.49	0.35

Table 11 (a) A comparison of similarity criteria based on coefficients of determination and mean absolute errors (MAE means Mean Absolute error, target video flood1.avi)

flood1	**flood1**	**flood1p2**	**flood1p1**	**Flood2**	**flood3**	**touch**	**house**	**land**	**sea**
R^2, feature combination	0.99	0.97	0.97	0.68	0.44	0.39	0.34	0.33	0.26
R^2, image histogram	1.00	0.96	0.96	0.38	0.18	0.07	0.00	0.00	0.03
R^2, edge direction	0.99	0.79	0.85	0.02	0.14	0.13	0.01	0.14	0.16
R^2, slope feature	0.99	0.97	0.97	0.88	0.65	0.65	0.68	0.67	0.55
R^2, wavelets	1.00	0.93	0.93	0.56	0.32	0.41	0.50	0.39	0.43
MAE, feature combination	22.6	31.7	30.4	87.0	113.6	108.4	100.0	101.2	106.5
MAE, image histogram	11.1	29.3	30.7	150.1	176.8	158.9	171.0	182.0	182.6
MAE, edge direction	3.9	20.5	14.1	39.5	72.4	60.1	62.0	61.9	76.3
MAE, slope feature	28.2	40.2	37.2	64.7	116.0	94.5	94.1	89.4	106.2
MAE, wavelets	0.8	5.2	5.2	14.6	17.4	13.0	17.7	18.7	16.9

Table 11 (b) A comparison of similarity criteria based on coefficients of determination and mean absolute errors (MAE means Mean Absolute error, target video flood2.avi)

Flood2	**flood2**	**flood2p2**	**flood2p1**	**Flood1**	**flood3**	**touch**	**land**	**house**	**sea**
R^2, feature combination	1.00	0.95	0.94	0.67	0.49	0.48	0.44	0.42	0.34
R^2, image histogram	1.00	0.90	0.86	0.33	0.15	0.17	0.00	0.00	0.09
R^2, edge direction	0.98	0.78	0.86	0.02	0.33	0.02	0.09	0.00	0.10
R^2, slope feature	1.00	0.97	0.98	0.84	0.80	0.71	0.78	0.73	0.61
R^2, wavelets	1.00	0.90	0.92	0.52	0.67	0.40	0.54	0.51	0.33
MAE, feature combination	6.8	27.3	23.8	73.7	100.8	86.0	100.0	96.8	95.8
MAE, image histogram	5.8	40.2	41.6	146.2	165.6	145.7	166.6	169.1	170.3
MAE, edge direction	5.9	16.2	13.3	43.8	68.2	67.1	62.1	61.6	82.9
MAE, slope feature	15.5	35.3	30.5	91.7	84.1	85.7	83.3	97.5	96.1
MAE, wavelets	1.4	6.6	5.4	16.6	11.8	13.1	16.0	17.4	18.4

Table 12. Key frame based similarity analysis for pan video shots using R2 (V=8, D=4, number of key frames=3)

House	**House**	**Housesm**	**Housemd**	**Housesb**	**Housese**	**Land**	**flood2**	**Touch**	**flood1**	**flood3**	**Sea**
Feature combination	1.00	0.97	0.80	0.68	0.67	0.42	0.41	0.38	0.34	0.31	0.25
Image histogram	1.00	0.99	0.37	0.25	0.32	0.00	0.02	0.07	0.01	0.01	0.01
Edge direction	0.99	0.96	0.63	0.19	0.20	0.03	0.00	0.05	0.00	0.00	0.05
Slope feature	0.99	0.96	0.90	0.82	0.79	0.67	0.74	0.64	0.64	0.63	0.60
Wavelets	0.99	0.64	0.83	0.60	0.69	0.51	0.52	0.48	0.49	0.42	0.41

Land	**Land**	**landsm**	**Landsb**	**Landmd**	**landse**	**flood2**	**house**	**flood3**	**touch**	**flood1**	**sea**
Feature combination	0.99	0.90	0.72	0.68	0.60	0.42	0.41	0.35	0.35	0.35	0.25
Image histogram	0.98	0.75	0.36	0.14	0.17	0.00	0.00	0.00	0.00	0.00	0.00
Edge direction	0.99	0.97	0.36	0.77	0.45	0.06	0.03	0.26	0.19	0.19	0.26
Slope feature	0.99	0.94	0.86	0.92	0.80	0.82	0.70	0.77	0.67	0.67	0.55
Wavelets	1.00	0.61	0.67	0.83	0.56	0.55	0.52	0.55	0.54	0.37	0.26

Table 13. Key frame-based similarity analysis for pan video shots using MAE (V=8, D=4, number of key frames=3)

house	**house**	**housesm**	**housemd**	**housese**	**Housesb**	**touch**	**land**	**sea**	**flood2**	**flood1**	**flood3**
Feature combination	10.81	15.13	55.22	62.78	73.91	100.58	105.16	111.26	123.65	125.19	148.90
Image histogram	7.52	9.80	86.54	121.30	124.22	165.95	159.06	187.53	203.91	208.20	215.83
Edge direction	7.65	7.95	35.45	50.91	55.98	64.19	63.42	83.79	39.80	45.58	86.79
Slope feature	27.86	23.97	53.33	71.27	73.32	111.72	100.60	111.08	102.93	116.86	137.07
Wavelets	2.16	5.62	8.72	12.57	14.99	12.86	16.78	16.49	15.15	16.91	16.36

land	**land**	**landsm**	**landmd**	**landsb**	**Landse**	**touch**	**house**	**sea**	**flood3**	**flood1**	**flood2**
Feature combination	16.39	20.57	61.88	70.06	76.10	101.32	105.32	111.53	116.33	126.44	134.75
Image histogram	35.00	42.61	153.22	136.64	147.86	172.41	170.64	178.19	222.49	218.52	217.14
Edge direction	8.01	7.91	28.59	48.93	53.03	58.49	63.44	74.25	65.39	41.20	39.27
Slope feature	23.60	26.62	43.88	63.26	72.15	83.96	101.57	102.33	98.57	109.24	81.66
Wavelets	1.07	5.62	9.45	13.50	14.83	12.04	17.45	18.37	13.47	18.44	14.54

3. Key frame based similarity analysis for zoom video shots

In this group of video shots, zoom and non-zoom video shots are included. Among these seven zoom video shots, six are zoom-in video shots and one is a zoom-out video shot. The beginning of all the zoom-in video shots is the whole earth. From the whole earth, the zoom-in video zooms onto a detailed location on the earth's surface. The zoom-out video shot proceeds in the reverse order. A brief description of the seven zoom video shots is given below:

- Atlanta.avi – zoom into the downtown area of Atlanta, GA
- DC.avi – zoom into the Washington DC area
- Orlando.avi – zoom into the Orlando area, FL
- NY.avi – zoom into the New York city area, NY
- Longbh.avi – zoom into the Long Beach area, CA
- Hollwd.avi – zoom into the Hollywood area, CA
- Atl_out.avi – zoom out from the downtown area of Atlanta, GA

Among the above video shots, Atlanta.avi is selected as the target video shot. A similarity analysis between the Atlanta video shot and other video shots is conducted. From the results (Table 14), we can observe that the similarities of the video shots DC, Orlando, and NY to the Atlanta video shots are higher than those of the video shots Longbh and Hollwd. The main reasons for this are the following: first, Atlanta, Washington DC, Orlando, and New York city are all located on the east coast, where as Hollywood and Long Beach, CA are located on the west coast; second, video shots Atlanta, DC, Orlando, and NY, all zoom into urban areas, where as video shot Hollwd zooms into a mountain area with the word "HOLLYWOOD" and video shot Longbh zooms into an elongated

island area surrounded by water. We may also notice that the Atl_out video shot bears the least similarity to the Atlanta zoom-in video shot because of the reverse order of frames in the video shot.

Another observation is that the combination of different features typically yields a better result for similarity analysis. Table 14 also shows the results based on the mean absolute error. We find that the mean absolute error can also indicate the relative order of similarity among video shots, but it is not as consistent as the coefficient of determination.

4. Summary for Key frame based similarity analysis

In above three sections, we have presented and discussed the results of similarity analysis for three types of data sets. This section will summarize the above discussion. Although the similarity analysis with a single individual feature can yield a reasonable result, the results from different features may yield different similarity orders because different features may reflect different intrinsic properties of the underlying video shot (Tables 8, 12, and 14). A combination of different features may be a better option for similarity analysis since the different video properties are represented as a weighted combination.

From the video shots we examined, the change of initial weights has only a minor effect on the final results. Based on the features input to the ANN, the ANN model will automatically generate the weights for the features combination. Thus, the initial choice of weights has less importance. This is a major advantage of using an ANN for similarity analysis because we do not need to pay a great deal of attention to the choice of the initial

weights if we do not have much information about the video shots that will used for similarity analysis. The initial weights are refined over course of training of the ANN.

In this study, we find that varying the number of key frames does not result in a significant difference in the rank order of the query video shots (Table 10). However, this would not necessarily be true in all cases since many other factors affect the results, such as the contents of video shot, the scheme of key frame selection and the location of each key frame within the video shot. There may exist an optimal number of key frames to be used for the purpose of similarity analysis of a particular group of video shots.

We would also like to mention the similarity criterion used in our study. We find that the coefficient of determination may be a better criterion than the mean absolute error to quantify the similarities among the video shots. The mean absolute error is essentially the feature (or feature combination) distance between video shots and could also be used as an indicator of similarity. However, we found that the mean absolute error is not as consistent as the coefficient of determination for the purpose of similarity analysis.

Mosaic based and key frame based similarity analysis

So far we have discussed two methods for similarity analysis, namely mosaic based analysis and key frame based analysis. In this section, we will compare these two methods, and then discuss and compare their advantages and disadvantages. As previously mentioned, a mosaic image represents the entire scene for a pan video shot, where the redundancy between frames is removed. The mosaic image is essentially the compressed representation of a video shot without loss of spatial information content for

the video shots. The features that are extracted from the mosaic image are based on the entire spatial contents of the video shot. The similarity analysis of video shots based on their mosaic images may produce a more reliable result since there is no loss of spatial information (Table 15). Key frames are a small subset extracted from the set of all frames for the entire video shot. Depending on the different methods used for key frame extraction, the extracted key frames might lose some degree of spatial information content. The features extracted from these key frames may not represent all the spatial information within the video shot thus causing the results to be less reliable (Table 15). One way to alleviate this is to increase the number of key frames. However, this increase in the number of key frames would significantly increase the work load. In addition, increasing the number of key frames may introduce errors in that the combination of key frames may not be the right one. Therefore, instead of improving the similarity measurement it may reduce the power of the method. Our study shows that the rank orders in both mosaic and key frame based similarity measurement methods are roughly similar although they vary in terms of the computed similarity measure. Because of loss of video spatial information with the key frame based method, the resulting coefficient of determination (similarity measure) with the key frame based method is lower than that computed with the mosaic based method if the query video shot is highly similar to the target video shot. For the query video shots have a low similarity to the target video shot, the reverse may be observed. This tends to reduce the reliability of key frame based similarity analysis.

For mosaic based similarity analysis, there are also some limitations. Although the mosaic image retains the spatial information in the video shot, the temporal

information is lost. If there are two videos with same scene but in the reverse frame order, the mosaic based method will not detect any difference. In contrast, the key frame based method would detect the difference because of the difference in the temporal sequence of the key frames. Another limitation of the mosaic based method is that this method can only be used effectively for pan video shots. Other types of video shots (such as zoom video shots) cannot be represented effectively with mosaic images. Key frame based similarity analysis would not have this limitation. Therefore, the key frame based similarity method would potentially have a wider range of application. This would lead us to think about the similarity analysis in future study. For panoramic video shots we may choose mosaic based analysis. For other video shots we may choose key frame based analysis.

Table 14 a. Key frame based similarity analysis for zoom video shots (V=8, D=4, number of key frames=5)

Atlanta	**Atlanta**	**DC**	**Orlando**	**NY**	**hollwd**	**touch**	**Longbh**
R^2, feature combination	1.00	0.76	0.68	0.48	0.43	0.30	0.28
R^2, image histogram	0.99	0.72	0.60	0.29	0.20	0.12	0.17
R^2, edge direction	1.00	0.36	0.53	0.30	0.11	0.06	0.14
R^2, slope feature	1.00	0.81	0.73	0.63	0.60	0.43	0.40
R^2, wavelets	1.00	0.69	0.58	0.53	0.50	0.15	0.24
Atlanta	**Atlanta**	**DC**	**Orlando**	**NY**	**touch**	**house**	**land**
MAE, feature combination	18.5	75.8	79.5	96.2	115.6	116.3	119.6
MAE, image histogram	10.0	135.6	157.4	169.0	160.0	147.6	145.1
MAE, edge direction	6.4	55.1	57.2	64.4	60.4	62.5	65.6
MAE, slope feature	17.5	72.8	82.6	114.5	126.8	118.7	130.0
MAE, wavelets	1.1	15.1	20.1	19.0	16.4	21.8	20.9

Table 14 b. Key frame based similarity analysis for zooming video shots(MAE means Mean Absolute Error, V=8, D=4, number of frames=5)

Atlanta	**Atlanta**	**house**	**land**	**flood3**	**flood2**	**sea**	**flood1**	**Atl_out**
R^2, feature combination	1.00	0.25	0.23	0.22	0.18	0.18	0.15	0.13
R^2, image histogram	0.99	0.04	0.00	0.03	0.04	0.00	0.03	0.06
R^2, edge direction	1.00	0.00	0.07	0.00	0.00	0.02	0.00	0.12
R^2, slope feature	1.00	0.38	0.50	0.43	0.34	0.44	0.27	0.16
R^2, wavelets	1.00	0.27	0.25	0.20	0.20	0.05	0.14	0.15
Atlanta	**Atlanta**	**sea**	**hollwd**	**Longbh**	**Atl_out**	**flood1**	**flood2**	**flood3**
MAE, feature combination	18.5	123.9	127.8	129.4	133.9	148.9	153.6	160.2
MAE, image histogram	10.0	189.5	272.5	187.9	154.6	219.8	216.7	206.6
MAE, edge direction	6.4	89.9	55.5	74.7	110.8	44.7	40.8	78.7
MAE, slope feature	17.5	111.5	95.2	134.3	161.6	174.7	176.3	149.2
MAE, wavelets	1.1	20.9	19.9	29.1	28.7	22.1	19.3	19.0

Table 15 (a) Comparison of mosaic based similarity analysis and key frame based similarity analysis (mosaic: mosaic based similarity analysis, frame: key frame based similarity analysis, target video shot: house.avi)

House	**House**	**Housesm**	**Housemd**	**housese**	**housesb**	**touch**	**land**	**sea**
Feature combination, mosaic	0.99	0.99	0.97	0.97	0.87	0.69	0.52	0.50
Image histogram, mosaic	0.98	0.97	0.93	0.93	0.83	0.58	0.38	0.25
Edge direction, mosaic	1.00	0.99	0.92	0.85	0.41	0.24	0.00	0.01
Slope feature, mosaic	1.00	0.99	0.98	0.98	0.88	0.76	0.57	0.65
Wavelets, mosaic	1.00	0.96	0.97	0.94	0.79	0.78	0.66	0.67
Feature combination, frame	1.00	0.97	0.80	0.67	0.68	0.38	0.42	0.25
Image histogram, frame	1.00	0.99	0.37	0.32	0.25	0.07	0.00	0.01
Edge direction, frame	0.99	0.96	0.63	0.20	0.19	0.05	0.03	0.05
Slope feature, frame	0.99	0.96	0.90	0.79	0.82	0.64	0.67	0.60
Wavelets, frame	0.99	0.64	0.83	0.69	0.60	0.48	0.51	0.28

Table 15 (b) Comparison of mosaic based similarity analysis and key frame based similarity analysis (mosaic: mosaic based similarity analysis, frame: key frame based similarity analysis, target video shot: land.avi)

Land	**Land**	**Landsm**	**Landse**	**landmd**	**landsb**	**touch**	**house**	**sea**
Feature combination, mosaic	1.00	1.00	0.98	0.95	0.70	0.52	0.44	0.29
Image histogram, mosaic	0.99	0.98	0.95	0.89	0.53	0.38	0.19	0.04
Edge direction, mosaic	0.97	0.97	0.94	0.87	0.59	0.18	0.00	0.20
Slope feature, mosaic	1.00	1.00	0.99	0.96	0.77	0.56	0.53	0.39
Wavelets, mosaic	1.00	0.95	0.91	0.92	0.75	0.63	0.59	0.60
Feature combination, frame	0.99	0.90	0.60	0.68	0.72	0.35	0.41	0.25
Image histogram, frame	0.98	0.75	0.17	0.14	0.36	0.00	0.00	0.00
Edge direction, frame	0.99	0.97	0.45	0.77	0.36	0.19	0.03	0.26
Slope feature, frame	0.99	0.94	0.80	0.92	0.86	0.67	0.70	0.55
Wavelets, frame	1.00	0.61	0.56	0.83	0.67	0.54	0.52	0.26

CHAPTER 6

CONCLUSION

In this study, similarity analysis of video shots is performed. Two methods are proposed. In the first method, the mosaic scene images are generated from the video shots. The feature extraction and similarity matching are conducted using the mosaic images. Since not all video shots can be effectively represented with mosaic images, the key frame based similarity analysis is also investigated. Instead of generating mosaic images, key frames are extracted from the original video shot and are arranged in the same temporal order as they appear in the video shot. The features are extracted from each key frame of the video shot. The similarity matching is then performed based on the paired frame features between the target video shot and the query video shots.

Different features are extracted from both mosaic images and key frames for each video shot. The features in feature space include image histograms, slope magnitudes and slope directions, edge directions, and wavelets. Similarity analysis with a single feature shows that the results from the edge feature are very sensitive, where as the results from the wavelet and slope features are less sensitive. A combination of features is allowed in our study with an adjustable weight for each feature. The weights are adjusted either automatically with ANN training or with initial weight setting. This allows us to adjust the weight for each feature to optimize the results in different types of similarity analysis.

In this study, we performed a non-linear feature combination of the weights using an ANN. The Ward ANN is used in our study. The Ward network model has the capability of grouping different data patterns with different activity functions. It can also determine the optimal number of hidden neurons. Unlike conventional approaches, the ANN architecture we developed has many outputs with each output representing the coefficient of determination, mean absolute errors, or any other measure of similarity between the target video shot and the query video shot. Another characteristic of the ANN architecture is that we do not provide evaluation data (i.e. production data set) for the network, that is, we only divided the feature data patterns into a training data set and test data set, with percentages of 60 and 40 respectively. The reason for this is that the system we have developed will not used to predict other new problem(s). We only need to examine the current feature data and find the similarity between them.

With the test results for the video data we collected in this study, we find that the weighted feature combination may yield better results than single feature for similarity analysis although exceptions do exist. We have also compared the two measures, i.e. coefficient of determination and mean absolute error, for similarity analysis. We find that using the coefficient of determination for similarity analysis may be better than using the mean absolute error. The mean absolute error is essentially a measurement of distance between features. Our study shows that the mean absolute error has a relatively larger variance than the coefficient of determination. Using the correlation coefficient would be preferred on account of its greater consistency.

The developed system needs to be further tested with other video data sets, especially with very dynamic video shots. Although we used different types of video

shots in our prototype system test, the scene structure in our test video shots is generally uniform and relatively slowly varying. In addition, there are other features, such as edge magnitudes and motion features, that could also be added into the system. In our study, we have performed the similarity analysis on video shots rather than an entire video sequences. Based on similarity analysis of video shots, we could also further study the similarity of entire video sequences.

REFERENCES

D. A. Adjeroh, M. C. Lee, and I. King. 1998. A distance measure for video sequence similarity mathching. *IEEE Workshop on Multimedia Database management System*. p.72-79.

R. Alferez, and Y. –F. Wang. 1999. Image indexing and retrieval using image-derived, geometrically and illumination invariant features. *IEEE International Conference on Multimedia Computing Systems.* v. 1, p. 177-182.

D. Androutsos, K. N. Plataniotis, and A. N. Venetsanopoulos. 1999. A novel vector-based approach to color image retrieval using a vector angular-based distance metric. *Computer Vision Image Understanding*. v. 75, n. 1, p. 46-58.

S. Antani, R. Kasturi, R. Jain. 2002. A survey on use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*. v.35, p. 945-965.

F. G. Ashby, and N. A. Perrin. 1988. Toward a unified theory of similarity and recognition. *Psychological Review*. V 95, n 1, p. 124-150.

S. Berretti, A. Del Bimbo, and E. Vicario. 2002. Spatial arrangement of color in retrieval by visual similarity. *Pattern Recognition*. v. 35, p. 1661-1674.

J. Canny. 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. v. 8, n. 6, p. 679-697.

S. H. Cha, and S. N. Srihari. 2002. On measuring the distance between histograms. *Pattern Recognition*. v. 35, p. 1355-1370.

Y. K. Chan, and C. C. Chang. 2001. Image matching using run-length feature. *Pattern Recognition Letters*. v. 22, p. 447-455.

G. Ciocca, and R. Schettini. 2002. Content-based similarity retrieval of trademarks using relevance feedback. *Pattern Recognition*. v. 34, p.1639-1655.

C. Colombo, and A. Del bimbo. 1999. Color induced image representation and retrieval. *Pattern Recognition*. v. 32, n. 10, p. 1685-1696.

A. Del bimbo, M. Mugnaini, P. Pala, F. Turco, 1998. Visual querying by color perceptive regions. *Pattern Recognition*. v. 31, n. 9, p. 1241-1253.

A. Del Bimbo, and P. Pala. 1996. Effective image retrieval using deformable templates. *Proceedings of the International Conference on Pattern Recognition*. p. 120-124.

I. Fudos, and L. Palios. 2002, An efficient shape-based approach to image retrieval. *Pattern Recognition Letters*. v. 23, p. 731-741.

K. Hirata, and T. Kato. 1992. Query by visual example – content based image retrieval. In Pirotte, Delobel, and Gottlob, editors, *Advances in Database Technology (EDBT'92)*. p. 56-71. Springer-Verlag, Berlin.

M. Inoue, Y. Mitsukura, M. Fukumi, and N. Akamatsu. 2000. Neural net based retrieval by using color and location information. *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*. v. 4, p. 2575-2579.

C. E. Jacobs, A. Finkelstein, and D. H. Salesin. 1995. Fast multi-resolution image querying. *Proceedings of SIGGRAPH '95*. p. 277-286. ACM, New York, NY.

A. K. Jain, and A. Vailaya. 1998. Shape-based retrieval: a case study with trademark image databases. *Pattern Recognition*. v. 31, n. 9, p. 1369-1390.

A. K. Jain, A. Vailaya, and X. Wei. 1999. Query by video clip. *Multimedia Systems*. v7, p. 369-384.

M. S.. Kankanhalli, B. M Mehtre, and H. Y. Huang. 1999. Color and spatial feature for content-based image retrieval. *Pattern Recognition Letters*. v. 20, p. 109-118.

M. S.. Kankanhalli, B. M. Mehtre, and J. K. Wu. 1996. Cluster based color matching for image retrieval. *Pattern Recognition*. v. 29, n.4, p.701-708.

C. L. Krumhansl. 1985. Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density. *Psychological Review*. V 85, n 5, p. 445-463.

H. K. Lee, and S. I. Yoo. 2000. Neural network-based image retrieval using nonlinear combination of heterogeneous features. *Proceedings of IEEE Conference on Evolutionary Computation*. v. 1, p. 667-674.

J. H. Lim, J. K. Wu, and D. Narasimhalu. 2001. Learning similarity matching in multimedia content-based retrieval. *IEEE Transactions on Knowledge and Data Engineering*. v. 13, n. 5, p. 846-850.

W. Y. Ma , and B. S. Manjunath. 1998. A texture Thesaurus for Browsing Large Aerial Photographs. *Journal of the American Society for Information Science*. v. 49, n. 7, p. 633-648.

M. K. Mandal, T. Aboulnasr, S. Panchanathan. 1999. Fast wavelet histogram techniques for image indexing. *Computer Vision Image Understanding*. v. 75, n. 2, p. 186-196.

B. S. Manjunath, and W. Y. Ma. 1996. Texture features for browsing and retrieval of image data. *IEEE Transaction of Pattern Analysis and Machine Intelligence*. v. 18, n. 8, p.837-842.

A. Mojsilovic, J. Kovacevic, J. Hu, R. J. Safranek, and S. K. Ganpathy. 1999. Matching and retrieval based on vocabulary and grammar of color patterns. *Proceedings of IEEE International Conference on Multimedia Computing Systems*. v. 1, p. 189-194.

F. Mokhtarian, and S. Abbasi. 2002. Shape similarity retrieval under affine transforms. *Pattern Recognition*. v. 35, p. 31-41.

I. D. Moore, A. Lewis, and J. C. Gallant. 1993. Terrain attributes: Estimation methods and scale effects. In *Modeling Change in Environmental System* (Edited by Jakeman et al.). John Wiley & Sons Ltd. p. 190-213.

W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. 1993. The QBIC project: Querying images by content using color, texture, and shape. *Storage and Retrieval for Image and video Databases, Volume 1908 of proceedings of the SPIE*, p 173-187. SPIE, Bellingham, WA, 1993.

P. Pala, and S. Santini. 1999. Image retrieval by shape and texture. *Pattern Recognition*. v. 32, p. 517-527.

A. Pentland, R. W. Picard, and S. Sclaroff. 1994. Photobook: Tools for content-based manipulation of image databases. *Proceedings of SPIE-94*, Bellingham, Washington. P.34-47.

E. G. M. Petrakis, and E. Milios. 1999. Efficient retrieval by shape content. *IEEE International Conference on Multimedia Computing Systems.* v. 2, p. 616-621.

J. Puzicha, T. Hofmann, J. M. Buhmann. 1997. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. *Proceedings of IEEE Conference on Computer Vision and pattern Recognition*. p.267-272.

G. Qiu. 2002. Indexing chromatic and achromatic patterns for content-based colour image retrieval. *Pattern Recognition*. v. 35, p. 1675-1686.

E. Rosh. 1975. Cognitive reference points. *Cognitive Psychology*, v 7, p. 532-547.

Y. Rubner, C. Tomasi, L. Guibas.1998. A metric for distributions with applications to image databases. *IEEE International Conference on Computer Vision*.

S. Santini, and R. Jain. 1999. Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. V 21, n 9, p. 871-883.

S. Scarloff. 1997. Deformable prototypes for encodeing shape categories in image databases. *Pattern Recognition*. v. 30, p. 627-641.

N. Sebe, and M. S. Lew. 2001. Color-based retrieval. *Pattern Recognition Letters*. v. 22, p. 223-240.

M. K. Shan and S. Y. Lee. 1998. Content-based video retrieval based on similarity of frame sequence. *Proceedings of IEEE workshop on multimedia database management systems.*

M. K. Shan and S. Y. Lee. 2001. A framework for temporal similarity measures of content-based scene retrieval. *Pattern Recognition Latter*. v 22, n 5, p. 517-532.

G. Sheikholeslami, S. Chatterjee, and A. Zhang. 1998. NeuroMerge: An approach for merging heterogeneous features in content-based image retrieval systems.

Proceedings of IEEE Workshop on Multimedia Database Management Systems. p. 106-113.

R. N. Shepard. 1964. Attention and the metric structure of the stimulus space. *Journal of Mathematical Psychology*. v 1, p.54-87.

J. R. Smith, and S. –F. Chang. 1994. Quad-tree segmentation for texture-based image query. *ACM International Conference on Multimedia*. P279-286.

J. Smith, and S. Chang. 1999. Integrated spatial and feature image query. *ACM Multimedia System*. v. 7, n. 2, p. 129-140.

E. J. Stollnitz, T. D. Derose, and D. H. Salesin. 1996. *Wavelets for computer graphics – Theory and applications*. Morgan Kaufmann Publishers, Inc. San Francisco, CA

M. Swain, and D. Ballard. 1991. Color indexing. *International Journal of Computer Vision*. v. 7, n1, p. 11-32.

A. Tversky. 1977. Feature of similarity. *Psychological Review*. V 84, n 11, p. 327-351.

A. Tversky and I. Gati. 1982. Similarity, separability, and the triangle inequality. *Psychological Review*. V. 89, p. 123-154

Ward System Group, Inc. 1996. *Neural Shell User's manual*. Frederick, MD

M. M. Yeung, and B. Liu. 1995. Efficient matching and clustering of video shots. *Proceedings of International Conference on Image Processing'95*, Washington, DC, p 338-341.

H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu. 1995. Video parsing, retrieval and browsing: An integrated and content-based solution. *Processings of ACM Multimedia'95*, San Francisco, CA, p. 15-24.

H. J. Zhang, J. H. Wu, D. Zhong, and S. W. Smoliar. 1997. An integrated system for

content-based video retrieval and browsing. *Pattern Recognition*. v 30, n4, p. 643-658.

Y. Zhong, and A. Jain. 2000. Object localization using color, texture and shape. *Pattern Recognition*. v. 33, n. 4, p. 671-684.

X. S. Zhou, and T. S. Huang. 2001. Edge-based structural features for content-based image retrieval. *Pattern Recognition Letters*. v. 22, p. 457-468.

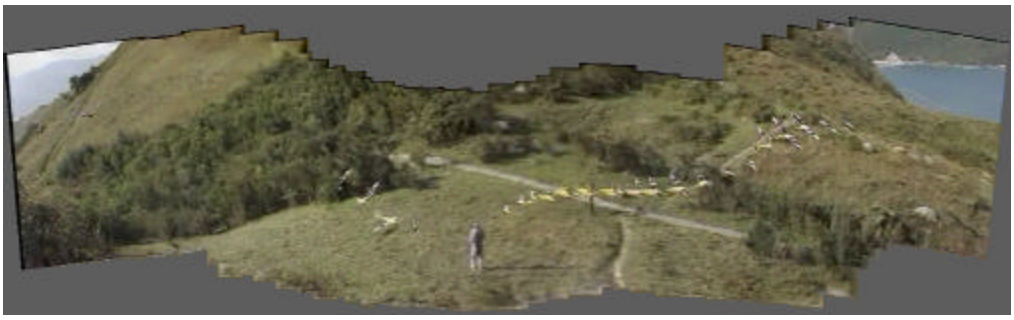
APPENDIX A. MOSAIC IMAGES GENERATED FROM PANORAMIC VIDEO SHOTS

a. Mosaic image from house.avi:

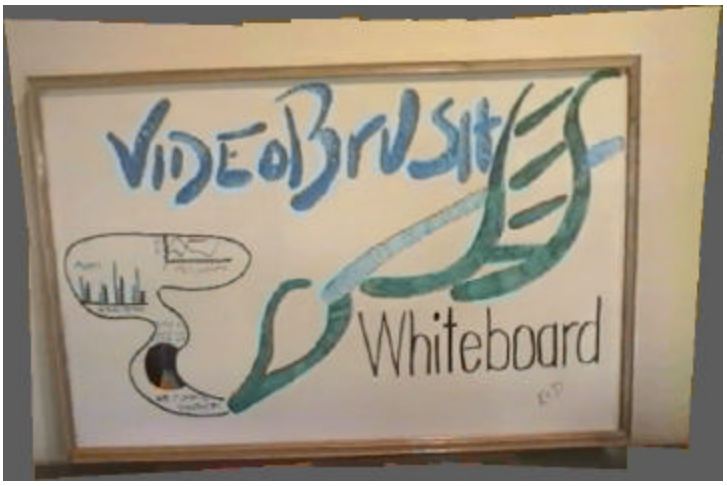
b. Mosaic image from landing.avi:

c. Mosaic image from sea.avi:

4. Mosaic image from touch.avi:



5. Mosaic image from whitebd.avi:



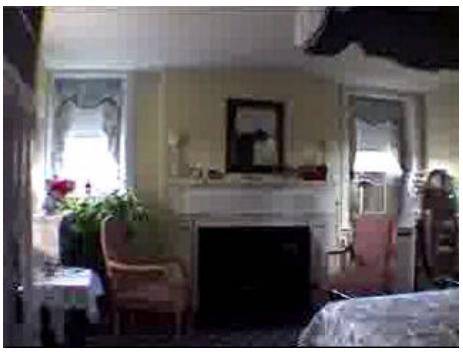
APPENDIX B. KEY FRAMES EXTRACTED FROM VIDEO SHOTS (NUMBER OF KEY FRAMES = 3)

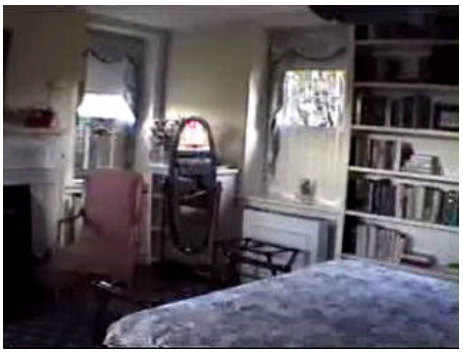
a. Key frames extracted from flood1.avi:

b. Key frames extracted from flood2.avi:

c. Key frames extracted from flood3.avi:

d. Key frames extracted from house.avi:



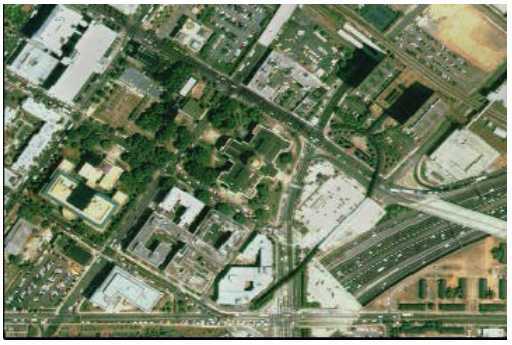


e. Key frames extracted from landing.avi:

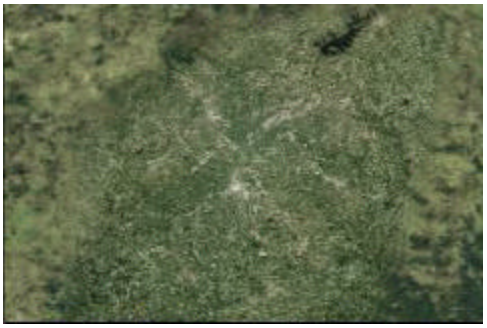
f. Key frames extracted from touch.avi:

i. Key frames extracted from sea.avi:

j. Key frames extracted from Alt_out.avi:

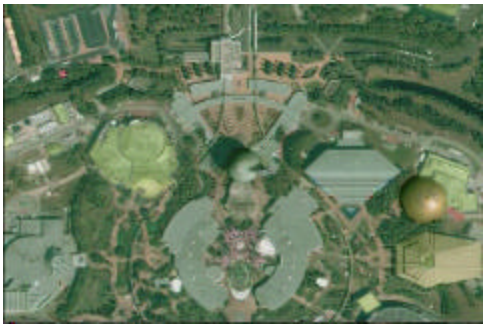


k. Key frames extracted from Atlanta.avi:

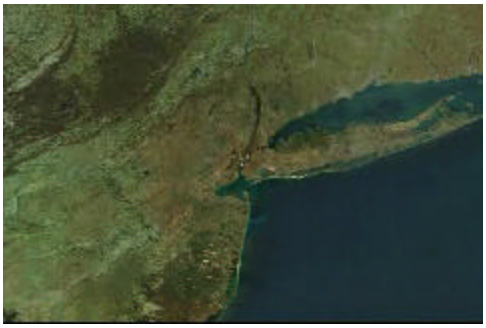


l. Key frames extracted from DC.avi:

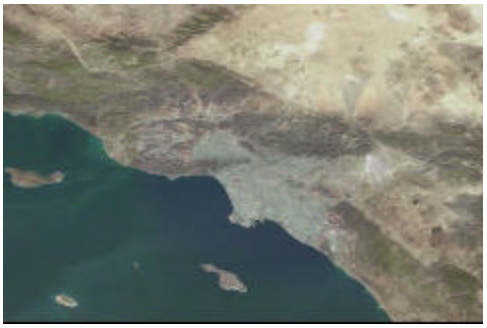
m. Key frames extracted from Orlando.avi:



n. Key frames extracted from NY.avi:



o. Key frames extracted from Hollwd.avi:



p. Key frames extracted from Longbh.avi: