

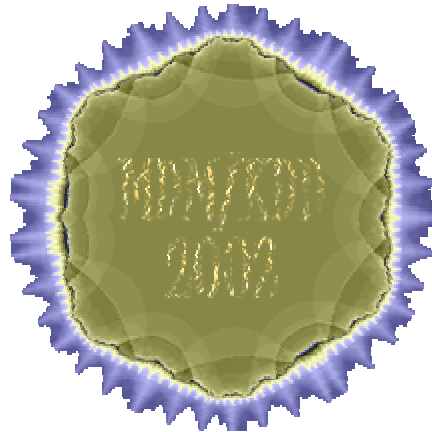
Proceedings

**Third International Workshop on
Multimedia Data Mining**

MDM/KDD'2002

July 23rd 2002

Edmonton, Alberta, Canada



In conjunction with

ACM SIGKDD

**Eighth International Conference on
Knowledge Discovery and Data Mining**

© The copyright of these papers belongs to the paper's authors. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage.

Proceedings of the Third International Workshop on Multimedia Data Mining (MDM/KDD'2002) in conjunction with ACM SIGKDD conference, Edmonton, Alberta, Canada, July 23rd 2002 (Simeon, J. Simoff, Chabane Djeraba and Osmar R. Zaïane, eds.)

Cover art production by Osmar R. Zaïane based on the conference poster by James W. Gary (Bucket Arts)

Proceedings printed in Canada by Quality Color Press Inc. Edmonton.

Foreword

Since the beginning of the century there have been two successful international workshops on multimedia data mining at the KDD forums: MDM/KDD2000 and MDM/KDD2001, in conjunction with KDD2000 (in Boston) and KDD2001 (in San Francisco), respectively. These workshops brought together numerous experts in spatial data analysis, digital media, multimedia information retrieval, state-of-art data mining and knowledge discovery in multimedia database systems, analysis of data in collaborative virtual environments. For more information about the workshops see the reports on the workshops in SIGKDD Explorations (**2** (2), pp. 103-105 and **3** (2), pp. 65-67, respectively). Participants in both workshops were pleased with the event and there was consensus about the necessity of turning it into an annual meeting, where researchers, both from the academia and industry can exchange and compare both relatively mature and green house theories, methodologies, algorithms and frameworks for multimedia data mining. This workshop is organized in response to this interest.

Being a third edition, the workshop this year is aiming to create a stimulating atmosphere for discussing the theoretical foundations of multimedia data mining, frameworks, methods and algorithms for integrated pattern extraction from multimedia data, multimedia data preprocessing, novel architectures for multimedia data mining, and applications of multimedia data mining in different areas. Consequently, the papers selected for presentation at the Third International Workshop on Multimedia Data Mining (MDM/KDD'2002) held in conjunction with the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining in Edmonton, Alberta, Canada, on July 23, 2002, are grouped in the following sessions: Frameworks for Multimedia Data Mining; Multimedia Data Mining Methods and Algorithms; and Applications of Multimedia Data Mining (with two subgroups of applications: in medical image analysis and in content-based multimedia processing). This grouping bears some similarity with the last year workshop, where there was similar emphasis on the research in the area of frameworks and methodologies, and on the research in the application area. The works selected for presentation at this workshop form more cohesive body of work, which indicates that the field has made a step forward towards achieving some level of maturity.

As part of the SIGKDD conference series the workshop follows a rigid peer-review and paper selection process. Once again, we would like to thank all those, who supported this year's efforts on all stages – from the development and submission of the workshop proposal to the preparation of the final program and proceedings. We would like to thank all those who submitted their work to the workshop. In a good data mining tradition, a pattern is emerging – as in the previous workshop there were submissions from 10 different countries. The difference is in the list of countries – this year it includes Australia, Brazil, Canada, France, Germany, Japan, Switzerland, Tunisia, United Kingdom, and United States of America. All papers were extensively reviewed by at least three referees drawn from the program committee. Special thanks go to them for the final quality of selected papers depends on their efforts.

Simeon, J. Simoff, Chabane Djeraba and Osmar R. Zaïane

June 2002

Table of Contents

Chairs and Program Committee	v
Workshop Program	vi
Multimedia Data Mining Framework For Raw Video Sequences Junghwan Oh and Babitha Bandi	1
An Innovative Concept For Image Information Mining Mihai Datcu and Klaus Seidel	11
Multimedia Data Mining Using P-Trees William Perrizo, William Jockheck, Amal Perera, Dongmei Ren, Weihua Wu, Yi Zhang .	19
Scale Space Exploration For Mining Image Information Content Mariana Ciucu, Patrick Heas, Mihai Datcu and James C. Tilton	30
Multimedia Knowledge Integration, Summarization And Evaluation Ana B. Benitez and Shih-Fu Chang	39
Object Boundary Detection For Ontology-Based Image Classification Lei Wang, Latifur Khan And Casey Breen	51
Mammography Cassification By An Association Rule-Based Classifier Osmar R. Zaiane, Maria-Luiza Antonie and Alexandru Coman	62
An Application Of Data Mining In Detection Of Myocardial Ischemia Utilizing Pre- And Post-Stress Echo Images Pramod K. Singh, Simeon J. Simoff and David Feng.....	70
From Data To Insight: The Community Of Multimedia Agents Gang Wei, Valery A. Petrushin and Anatole V. Gershman	76
A Content Based Video Description Scheme And Video Database Navigator Sadiye Guler and Ian Pushee	83
Subjective Interpretation Of Complex Data: Requirements For Supporting Kansei Mining Process Nadia Bianchi-Berthouze and Tomofumi Hayashi	93
User Concept Pattern Discovery Using Relevance Feedback And Multiple Instance Learning For Content-Based Image Retrieval Xin Huang, Shu-Ching Chen, Mei-Ling Shyu and Chengcui Zhang	100
Author Index	109

Workshop Chairs

Simeon J. Simoff
Chabane Djeraba

LocalChair

Osmar R. Zaïane

Program Committee

Marie-Aude Aufaure, INRIA, France
Terry Caelli, University of Alberta, Canada
Chabane Djeraba, University of Nantes, France
Chitra Dorai, IBM Thomas J. Watson Research Center, USA
Alex Duffy, University of Strathclyde, UK
William Grosky, Wayne State University, USA
Howard J. Hamilton, University of Regina, Canada
Jiawei Han, Simon Fraser University, Canada
Mohand-Said Hacid, Claude Bernard University, France
Wynne Hsu, National University of Singapore, Singapore
Odej Kao, University of Paderborn, Germany
Paul Kennedy, University of Technology-Sydney, Australia
Latifur Khan, University of Texas, USA
Inna Kolyshkina, Price Waterhouse Coopers, Australia
Brian Lovell, University of Queensland, Australia
Mark Maybury, MITRE Corporation
Gholamreza Nakhaeizadeh, DaimlerChrysler, Germany
Ole Nielsen, Australian National University, Australia
Monique Noirhomme-Fraiture, Institut d'Informatique, FUNDP, Belgium
Vincent Oria, New Jersey Institute of Technology, USA
Valery A. Petrushin Accenture, USA
Mohamed Quafafou, Institut de Recherche en Informatique de Nantes
Simone Santini, University of California San Diego, USA
Simeon J. Simoff, University of Technology Sydney, Australia
Pramod Singh, University of Technology Sydney, Australia
Duminda Wijesekera, George Mason University, USA

Program for MDM/KDD2002 Workshop

Tuesday, July 23, 2002, Edmonton, Alberta, Canada

9:00 - 9:10 Opening and Welcome

9:10 - 10:00 Session 1 - Frameworks for Multimedia Data Mining

- 09:10 - 09:35 MULTIMEDIA DATA MINING FRAMEWORK FOR RAW VIDEO SEQUENCES
JungHwan Oh and Babitha Bandi
- 09:35 - 10:00 AN INNOVATIVE CONCEPT FOR IMAGE INFORMATION MINING
Mihai Datcu and Klaus Seidel

10:00 - 10:30 Coffee break

10:30 - 12:10 Session 2 - Multimedia Data Mining Methods and Algorithms

- 10:30 - 10:55 MULTIMEDIA DATA MINING USING P-TREES
William Perrizo, William Jockheck, Amal Perera, Dongmei Ren, Weihua Wu, Yi Zhang
- 10:55 - 11:20 SCALE SPACE EXPLORATION FOR MINING IMAGE INFORMATION CONTENT
Mariana Ciucu, Patrick Heas, Mihai Datcu and James C. Tilton
- 11:20 - 11:45 MULTIMEDIA KNOWLEDGE INTEGRATION, SUMMARIZATION AND EVALUATION
Ana B. Benitez and Shih-Fu Chang
- 11:45 - 12:10 OBJECT BOUNDARY DETECTION FOR ONTOLOGY-BASED IMAGE CLASSIFICATION
Lei Wang, Latifur Khan and Casey Breen

12:10 - 13:30 Lunch

13:30 - 15:45 Session 3 - Applications of Multimedia Data Mining

Applications in Medical Image Analysis

- 13:30 - 13:55 MAMMOGRAPHY CASSIFICATION BY AN ASSOCIATION RULE-BASED CLASSIFIER
Osmar R. Zaiane, Maria-Luiza Antonie and Alexandru Coman
- 13:55 - 14:20 AN APPLICATION OF DATA MINING IN DETECTION OF MYOCARDIAL ISCHEMIA
UTILIZING PRE- AND POST-STRESS ECHO IMAGES
Pramod K. Singh, Simeon J. Simoff and David Feng

Applications in Content-Based Multimedia Processing

- 14:20 - 14:45 FROM DATA TO INSIGHT: THE COMMUNITY OF MULTIMEDIA AGENTS
Gang Wei, Valery A. Petrushin and Anatole V. Gershman
- 14:45 - 15:10 A CONTENT BASED VIDEO DESCRIPTION SCHEME AND VIDEO DATABASE
NAVIGATOR
Sadiye Guler and Ian Pushee
- 15:10 - 15:35 SUBJECTIVE INTERPRETATION OF COMPLEX DATA: REQUIREMENTS FOR
SUPPORTING KANSEI MINING PROCESS
Nadia Bianchi-Berthouze and Tomofumi Hayashi
- 15:35 - 16:00 USER CONCEPT PATTERN DISCOVERY USING RELEVANCE FEEDBACK AND
MULTIPLE INSTANCE LEARNING FOR CONTENT-BASED IMAGE RETRIEVAL
Xin Huang, Shu-Ching Chen, Mei-Ling Shyu and Chengcui Zhang

16:00 - 16:15 Discussion and Closure

16:15 - 17:00 Coffee break

17:00 Opening of SIGKDD 2002 Conference

MULTIMEDIA DATA MINING FRAMEWORK FOR RAW VIDEO SEQUENCES

JungHwan Oh, Babitha Bandi

Department of Computer Science and Engineering
University of Texas at Arlington
Arlington, TX 76019-0015 U. S. A.
e-mail: {oh, bandi}@cse.uta.edu

ABSTRACT

In this paper, we propose a general framework for real time video data mining to be applied to the raw videos (traffic videos, surveillance videos, etc.). We investigate whether the existing techniques would be applicable to this type of videos. Then, we introduce new techniques which are essential to process them in real time. The first step of our framework for mining raw video data is grouping input frames to a set of basic units which are relevant to the structure of the video. We call this unit as *segment*. This is one of the most important tasks since it is the step to construct the building blocks for video database and video data mining. The second step is characterizing each segment to cluster into similar groups, to discover unknown knowledge, and to detect interesting patterns. To do this, we extract some features (motion, object, colors, etc.) from each segment. In our framework, we focus on *motion* as a feature, and study how to compute and represent it for further processes. The third step of our framework is to cluster the decomposed segments into similar groups. In our clustering, we employ a multi-level hierarchical clustering approach to group segments using category and motion. Our preliminary experimental studies indicate that the proposed framework is promising.

KEYWORDS: Multimedia Data Mining, Video Segmentation, Motion Extraction, Video Data Clustering

1. INTRODUCTION

Data mining, which is defined as the process of extracting previously unknown knowledge, and detecting interesting patterns from a massive set of data, has been a very active research. As results, several commercial products and research prototypes are even available nowadays. However, most of these have focused on corporate data typically in alpha-numeric database. Even though relatively less research has been performed, very interesting and important studies have been published, and systems have been developed in the areas of multimedia data mining.

Multimedia data mining has been performed for different types of multimedia data; image, audio and video.

An example of image data mining is *CONQUEST* [1] system that combines satellite data with geophysical data to discover patterns in global climate change. The *SKICAT* system [2] integrates techniques for image processing and data classification in order to identify 'sky objects' captured in a very large satellite picture set. The *MultiMediaMiner* [3] project has constructed many image understanding, indexing and mining techniques in digital media.

An example of video and audio data mining can be found in *Mining Cinematic Knowledge* project [4] which creates a movie mining system by examining the suitability of existing concepts in data mining to multimedia, where the semantic content is time sensitive and constructed by fusing data obtained from component streams. A project [5, 6] analyzing the broadcast news programs has been reported. They have developed the techniques and tools to provide news video annotation, indexing and relevant information retrieval along with domain knowledge in the news programs. A data mining framework in audio-visual interaction has been presented [7] to learn the synchronous pattern between two channels, and apply it to speech driven lip motion facial animation system. The other example is a system [8] focusing on the echocardiogram video data management to exploit semantic querying through object state transition data modeling and indexing scheme. We can find some multimedia data mining frameworks [9, 10, 11] for traffic monitoring system. EasyLiving [12, 13] and HAL [14] projects are developing smart spaces that can monitor, predict and assist the activities of its occupants by using ubiquitous tools that facilitate everyday activities.

As mentioned above, there have been some efforts about video data mining for movies, medical videos, and traffic videos. Generally, there are three types of videos; the produced, the raw, and the medical video. The examples of produced video are movies, news videos, dramas, etc. And, those of raw video are traffic videos, surveillance videos, etc. Ultra sound videos including echocardiogram can be an example of the medical videos. In fact, the developments of complex video surveillance systems [15] and traffic monitoring systems [10, 11, 16, 17, 18] have recently captured the interest of both research and industrial worlds due to the growing interest availability of

cheap sensors and processors at reasonable costs, and the increasing safety and security concerns. As mentioned in the literature [9], the common approach in these works is that the objects (i.e., person, car, airplane, etc.) are extracted from video sequences, and modeled by the specific domain knowledge, then, the behavior of those objects are monitored (tracked) to find any abnormal situations. What are missing in these efforts are first, how to index and cluster these unstructured and enormous video data for real-time processing, and second, how to mine them, in other words, how to extract previously unknown knowledge and detect interesting patterns.

These different types of videos need to be treated differently to achieve these missing parts due to their different characteristics. In this paper, we propose a general framework for video data mining to be applied to the *raw videos* in *real time*. We investigate whether the existing multimedia data mining techniques would be applicable to this type of videos. Then, we introduce new techniques which are essential to process them in real time. Figure 1 shows the proposed framework which can be summarized as follows.

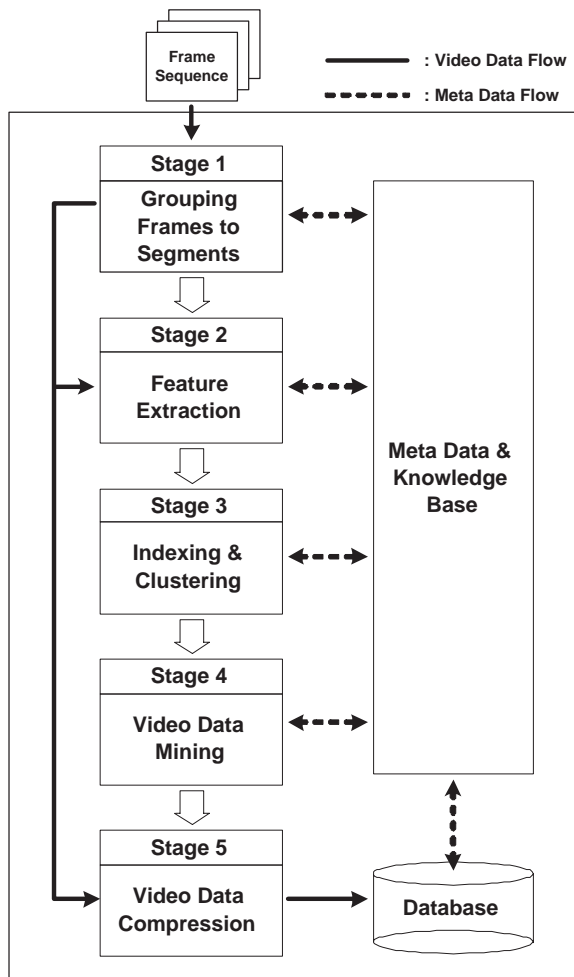


Fig. 1: Proposed Framework for Video Data Mining

- The first stage (Stage 1 in Figure 1) of our framework for mining raw video data is grouping input frames to a set of basic units which are relevant to the structure of the video. This is one of the most important tasks since it is the first step to construct the building blocks of the video database, and to convert videos from raw materials to *data* with semantic information. In general, the most widely used basic unit in produced videos (i.e., movies, news videos) is a *shot* which is defined as collections of frames recorded from a single camera operation. Raw videos are usually recorded from a single fixed camera or multiple cameras with very limited camera motion without any camera on-off. Therefore, the concept of the shot is not relevant since whole video would be a shot by the above definition. In this paper, we investigate how to group the incoming frames into meaningful pieces in real time processing in which the traditional concept of shot is not applicable. This piece is called as *segment* to distinguish it from shot. In addition to this linear decomposition, we build a hierarchical structure of segments. Therefore, we call our segmentation as *hierarchical segmentation*, and each segment is classified into a different category. Another advantage of this hierarchical segmentation is that it can give us various lengths of summaries for incoming videos automatically. More details will be discussed in the next section.

- The second stage (Stage 2 in Figure 1) characterizes each segment to cluster into similar groups, to discover unknown knowledge, and to detect interesting patterns. We need to extract the features such as motions, objects, colors, etc., to characterize these segments. It is not only the features that are important, but also the ways to represent them as we need to compare the decomposed segments to characterize them as mentioned above. For our framework, we consider three features (motions, objects, colors) extracted from each segment. Among these features, *motion* is investigated at this time, and the other features will be studied in near future. To extract motions, we use an accumulation of quantized pixel differences among all frames in a segment [19]. As a result, accumulated motions of segment are represented as a *two dimensional matrix*. The technique to compute motions is very cost-effective because an expensive computation (i.e., optical flow) is not necessary. Because the motions are represented as a matrix, comparison among segments is very efficient and scalable.

- The third stage (Stage 3 in Figure 1) of our framework is to cluster the decomposed segments into similar groups. In our clustering, we employ a multi-

level hierarchical clustering approach to group segments with similar categories in the top level, and similar motions in the bottom level. We use K-Mean algorithm and cluster validity method [20] due to its simplicity and efficiency. This clustering is a fundamental step for future knowledge discovery and pattern detection.

- The next stages (Stage 4 and 5 in Figure 1) are actual mining of raw video sequences processed in the above three stages, and video data compression for storage of these raw videos. The Meta Data & Knowledge Base in the figure is a module to store the results from each stage and provide the necessary information to the stages. The example of knowledge and patterns that we can discover and detect are object identification, object movement pattern recognition, spatio-temporal relations of objects, modeling and detection of normal and abnormal (interesting) events and event pattern recognition. We plan to develop techniques to perform the above mining tasks in near future. Also, a suitability and availability of various video compression techniques including MPEG will be investigated to store these video data in database physically.

The remainder of this paper is organized as follows. In Section 2, we describe a technique to group incoming frames into segments. A motion feature extraction technique is discussed in Section 3. In section 4, we propose a multi-level hierarchical clustering approach to group segments based on the categories, and the motions. The experimental results are discussed in Section 5. Finally, we give our concluding remarks in Section 6.

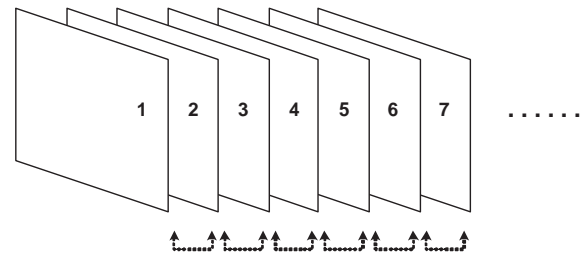
2. VIDEO SEGMENTATION

In this section, we discuss the details of the technique to group the incoming frames into semantically homogeneous pieces by real time processing (we called these pieces as ‘segments’ in the previous section). First, we look at the existing video partitioning techniques based on the concept of ‘shot’ to figure out what the limitations and the problems they have when they are applied to raw videos in which the definition of shot cannot be applied. Then we introduce a novel technique to decompose this type of videos.

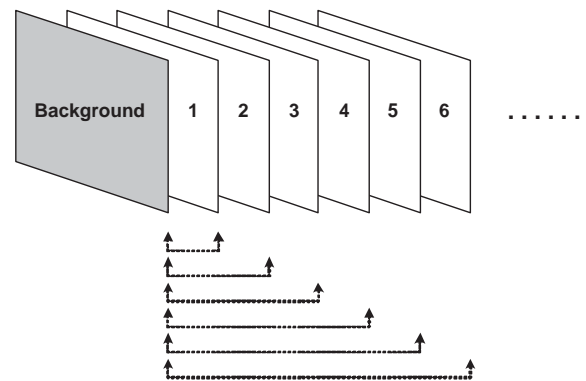
2.1. Existing Techniques for Video Segmentation

In many number of literature, the process for video segmentation is referred to as *shot boundary detection* (SBD) in general since they are dealing with shot as a unit for segmentation. This SBD has been an area of active research. Many techniques have been developed to automatically detect transitions from one shot to the next. These

schemes differ mainly in the way that the inter-frame difference is computed. The main idea of these techniques is that if the difference between the two consecutive frames (see Figure 2(a)) is larger than a certain threshold value, then a shot boundary is considered between two corresponding frames. The difference can be determined by comparing the corresponding pixels of two images [21]. Color or grayscale histograms can also be used [22]. Alternatively, a technique based on changes in edges has also been developed [23]. Other schemes use domain knowledge [24] such as predefined models, objects, regions, etc. Hybrids of the above techniques have also been investigated [25, 26, 27, 28, 29].



(a) Inter Frame Difference between Two Consecutive Frames



(b) Inter Frame Differences with Background Frame

Fig. 2: Frame Comparison Strategies

However, this technique is not effectively working for the raw videos in which there is little camera motions in most sequences. The dotted curve in the bottom of Figure 3 shows the color histogram differences between two consecutive frames in a raw video sequence. Note that this sequence was taken from a crowded hallway in a building, and digitized as 5 frames per second. As shown by this curve, there is not much difference between two consecutive frames. In fact, most of them are less than 10%. In other words, if we use the differences between consecutive frames, most of the frames are to be considered very similar. Therefore, it is very difficult to find clear boundaries for segments. To address this drawback, we propose a new technique for raw video segmentation in the following subsection.

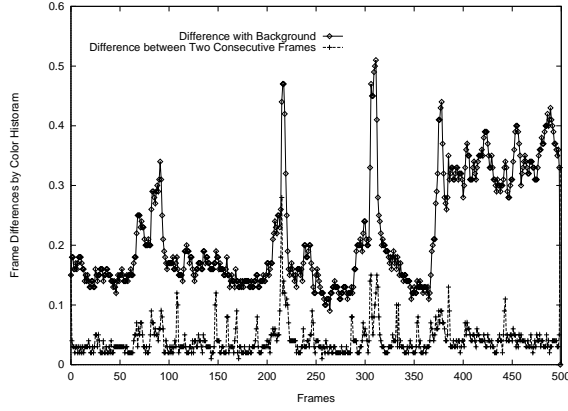


Fig. 3: Two Frame Comparison Strategies

2.2. New Technique for Raw Video Segmentation

The idea of new technique is very simple. Instead of comparing two consecutive frames, we compare each frame with a *background* frame as shown in Figure 2(b). A background frame is defined as a frame with only non-moving components. Since we can assume that the camera remains stationary for our application, a background frame can be a frame of the stationary components in the image. In this work, we manually select a background frame using similar approach in [9]. The solid curve in the top of Figure 3 shows the color histogram difference of background with each frame in the sequence. The differences are magnified so that segment boundaries can be found more clearly. The algorithm to decompose a raw video sequence into meaningful pieces (segments) is summarized as follows. The Step.1 is a preprocessing by off-line processing, and the Step.2 through 6 are performed by on-line real time processing. Note that since this segmentation algorithm is generic, the frame comparison can be done by any technique using color histogram, pixel-matching or edge change ratio. We chose a simple color histogram matching technique for illustration purpose.

- Step.1: A background frame is extracted from a given sequence as preprocessing, and its color histogram is computed. In other words, this frame is represented as a *bin* with a certain number (bin size) of quantized colors from the original. Usually the bin size is 128, 64 or 32 if the RGB value of a pixel in the original frame is 256. As a result, a background frame (F^B) is represented as follows using a *bin* with the size n . Note that P_T is representing the total number of pixels in a background or any frame.

$$F^B = bin^B = (v_1^B, v_2^B, v_3^B, \dots, v_n^B) \quad (1)$$

$$where \sum_{i=1}^n v_i^B = P_T.$$

- Step.2: Each frame (F^k) arrived to system is represented in the same way used to represent the background in the previous step as follows.

$$F^k = bin^k = (v_1^k, v_2^k, v_3^k, \dots, v_n^k) \quad (2)$$

$$where \sum_{i=1}^n v_i^k = P_T.$$

- Step.3: Compute the difference (D^k) between the background (F^B) and each frame (F^k) as follows. Note that the value of D^k is always between zero and one.

$$D^k = \frac{F^B - F^k}{P_T} = \frac{bin^B - bin^k}{P_T}$$

$$= \frac{\sum_{i=1}^n (v_i^B, -v_i^k)}{P_T} \quad (3)$$

- Step.4: Classify D^k into 10 different categories as follows based on its value. Assign a corresponding category number (C_k) to the frame k .

- Category 0 : $D^k < 0.1$
- Category 1 : $0.1 \leq D^k < 0.2$
- Category 2 : $0.2 \leq D^k < 0.3$
- Category 3 : $0.3 \leq D^k < 0.4$
- Category 4 : $0.4 \leq D^k < 0.5$
- Category 5 : $0.5 \leq D^k < 0.6$
- Category 6 : $0.6 \leq D^k < 0.7$
- Category 7 : $0.7 \leq D^k < 0.8$
- Category 8 : $0.8 \leq D^k < 0.9$
- Category 9 : $D^k \geq 0.9$

- Step.5: For real time on-line processing, a temporary table such as Table 1 is maintained. To do this and build a hierarchical structure from a sequence as mentioned in section 1, compare C_k with C_{k-1} . In other words, compare the category number of current frame with the previous frame. We can build a hierarchical structure from a sequence based on these categories which are not independent from each other. We consider that the lower categories contain the higher categories as shown in Figure 4.

Segment No.	Starting Frame No.	Ending Frame No.	Segment Length	Cat. (C_k)	Total Motion (TM)	Avg. Motion (AM)

Table 1: Segmentation Table

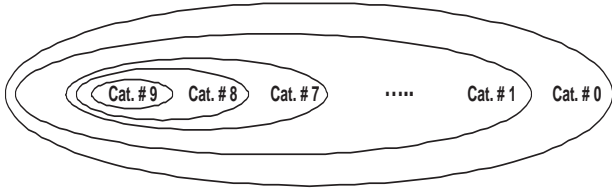


Fig. 4: Relationships (Containments) among Categories

For example, one segment A of Cat. # 1 starts with Frame # a and ends with Frame # b , and the other segment B of Cat. # 2 starts with Frame # c and ends with Frame # d , then it is possible that $a < c < d < b$. In our hierarchical segmentation, therefore, finding segment boundaries become finding category boundaries in which we find a starting frame (S_i) and an ending frame (E_i) for each category i . The following algorithm shows how to find these boundaries.

- If $C_{k-1} = C_k$, then no segment boundary occurs, so continue with the next frame.
 - Else if $C_{k-1} < C_k$, then $S_{C_k} = k$, $S_{C_{k-1}} = k$, ... $S_{C_{k-1}+1} = k$. The starting frames of category C_k through $C_{k-1} + 1$ are k .
 - Else, in other words, if $C_{k-1} > C_k$, then $E_{C_{k-1}} = k - 1$, $E_{C_{k-1}-1} = k - 1$, ..., $E_{C_k+1} = k - 1$. The ending frames of category C_{k-1} through $C_k + 1$ are $k - 1$.
 - If the length of a segment is less than a certain threshold value (β), we ignore this segment since it is too short to carry any semantic content. In general, this value β is one second. In other words, we assume that the minimum length of a segment is one second.
- **Step.6:** As mentioned in the previous section, without any extra computation, we can have several different versions of summaries for the incoming video which have different lengths, in other words, different levels of abstraction. The simple method is to pick all frames whose category value is greater than or equal to C , where $1 \leq C \leq 9$. As results, we can have up to 9 different versions of summaries.

3. MOTION FEATURE EXTRACTION

In this section, we describe how to extract and represent motions from each segment decomposed from a raw video sequence as discussed in the previous section. We developed a technique for automatic measurement of the overall motion in not only two consecutive frames but also whole shot which is a collection of frames in our previous works [30, 19]. We extend this technique to extract the motion from a segment, and represent it in a comparable form in this section. We compute *Total Motion Matrix (TMM)*

which is considered as the overall motion of a segment, and represented as a *two dimensional matrix*. For comparison purpose among segments with different lengths (in terms of number of frames), we also compute an *Average Motion Matrix (AMM)*, and its corresponding *Total Motion (TM)* and *Average Motion (AM)*.

The *TMM*, *AMM*, *TM* and *AM* for a segment with n frames is computed using the following algorithm (Step 1 through 5). We assume that the frame size is $c \times r$ pixels.

- **Step.1:** The color space of each frame is quantized (i.e., from 256 to 64 or 32 colors) to reduce unexpected noises (false detection of motion which is not actually motion but detected as motion).
- **Step.2:** An empty two dimensional matrix *TMM* (its size ($c \times r$) is same as that of frame) for a segment S is created as follows. All its items are initialized with zeros.

$$TMM_S = \begin{pmatrix} t_{11} & t_{12} & t_{13} & \dots & t_{1c} \\ t_{21} & t_{22} & t_{23} & \dots & t_{2c} \\ \dots & \dots & \dots & \dots & \dots \\ t_{r1} & t_{r2} & t_{r3} & \dots & t_{rc} \end{pmatrix} \quad (4)$$

And *AMM_S* which is a matrix whose items are averages computed as follows.

$$AMM_S = \begin{pmatrix} \frac{t_{11}}{n-1} & \frac{t_{12}}{n-1} & \frac{t_{13}}{n-1} & \dots & \frac{t_{1c}}{n-1} \\ \frac{t_{21}}{n-1} & \frac{t_{22}}{n-1} & \frac{t_{23}}{n-1} & \dots & \frac{t_{2c}}{n-1} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{t_{r1}}{n-1} & \frac{t_{r2}}{n-1} & \frac{t_{r3}}{n-1} & \dots & \frac{t_{rc}}{n-1} \end{pmatrix} \quad (5)$$

- **Step.3:** Compare all the corresponding quantized pixels in the same position of two consecutive frames. If they have different colors, increase the matrix value (t_{ij}) in the corresponding position by one (this value may be larger according to the other conditions). Otherwise, it remains the same.
- **Step.4:** Step.3 is repeated until all consecutive pairs of frames are compared.
- **Step.5:** Using the above *TMM_S* and *AMM_S*, we compute a motion feature, *TM_S*, *AM_S* as follows.

$$TM_S = \sum_{i=0}^r \sum_{j=0}^c t_{ij}, \quad AM_S = \sum_{i=0}^r \sum_{j=0}^c \frac{t_{ij}}{n-1} \quad (6)$$

As seen in these formulae, *TM* is the sum of all items in *TMM* and we consider this as total motion in a segment. In other words, *TM* can indicate an amount of motion in a segment. However, *TM* is dependent on not only the amount of motions but also the length of a segment. A *TM* of long segment

with little motions can be equivalent to a TM of short segment with a lot of motions. To distinguish these, simply we use AM which is an average of TM .

To visualize the computed TMM (or AMM), we can convert this TMM (or AMM) to an image which is called *Total Motion Matrix Image (TMMI)* for TMM (*Average Motion Matrix Image (AMMI)* for AMM). Let us convert a TMM with the maximum value, m into a 256 gray scale image as an example. We can convert an AMM using the same way. If m is greater than 256, m and other values are scaled down to fit into 256, otherwise, they are scaled up. But the value zero remains unchanged. An empty image with same size of TMM is created as $TMMI$, and the corresponding value of TMM is assigned as a pixel value. For example, assign white pixel for the matrix value zero which means no motion, and black pixels for the matrix value 256 which means maximum motion in a given shot. Each pixel value for a $TMMI$ can be computed as follows after it is scaled up or down if we assume that $TMMI$ is a 256 gray scale image.

$$\begin{aligned} \text{Each Pixel Value} = \\ 256 - \text{Corresponding Matrix Value} \end{aligned} \quad (7)$$

4. CLUSTERING OF SEGMENTS

In our clustering, we employ a multi-level hierarchical clustering approach to group segments in terms of category, and motion of segments. The algorithm is implemented in a top-down fashion, where the feature, category is utilized at the top level, in other words, we group segments into k_1 clusters according to the categories. For convenience, we called this feature as *Top Feature*. Each cluster is clustered again into k_2 groups based on the motion (AM) extracted in the previous section accordingly, which are called as *Bottom Feature*.

For this multi-level clustering, we adopted K-Mean algorithm and cluster validity method studied by Ngo et. al. [20] since the algorithm is the most frequently used clustering algorithm due to its simplicity and efficiency. It is employed to cluster segments at each level of hierarchy independently. The K-Mean algorithm is implemented as follows.

- **Step.1:** The initial centroids are selected in the following way:
 1. Given v d -dimensional feature vectors, divide the d dimensions to $\rho = \frac{d}{k}$. These subspaces are indexed by $[1, 2, 3, \dots, \rho], [\rho, \rho+1, \rho+2, \dots, 2\rho], \dots, [(k-1)\rho+1, (k-1)\rho+2, (k-1)\rho+3, \dots, k\rho]$.
 2. In each subspace j of $[(j-1)\rho+1, \dots, j\rho]$ associate a value f_i^j for each feature vector \mathcal{F}_i

by

$$f_i^j = \sum_{m=(j-1)\rho}^{j\rho} \rho \mathcal{F}_i(m)$$

3. Choose the initial cluster centroids $\mu_1, \mu_2, \dots, \mu_k$, by

$$\mu_j = \arg \max_{1 < i < v} \mathcal{F}_i^j$$

- **Step.2:** Classify each feature F to the cluster p_s with the smallest distance.

$$p_s = \arg \min_{1 \leq j \leq k} D(\mathcal{F}, \mu_j)$$

This D is a function to measure the distance between two feature vectors and defined as

$$D(\mathcal{F}, \mathcal{F}') = \frac{1}{\mathcal{Z}(\mathcal{F}, \mathcal{F}')} \left(\sum_{i=1}^v |\mathcal{F}(i) - \mathcal{F}'(i)|^k \right)^{\frac{1}{k}}$$

where

$$\mathcal{Z}(\mathcal{F}, \mathcal{F}') = \sum_{i=1}^v \mathcal{F}(i) + \sum_{i=1}^v \mathcal{F}'(i)$$

which is a normalizing function. In this function, $k = 1$ for L_1 norm and $k = 2$ for L_2 norm. The L_1 and L_2 norms are two of the most frequently used distance metrics for comparing two feature vectors. In practice, however, L_1 norm performs better than L_2 norm since it is more robust to outliers [31]. Furthermore, L_1 norm is more computationally efficient and robust. We use L_1 norm for our experiments.

- **Step.3:** Based on the classification, update cluster centroids as

$$\mu_j = \frac{1}{v_j} \sum_{i=1}^{v_j} \mathcal{F}_i^{(j)}$$

where v_j is the number of shots in cluster j , and $\mathcal{F}_i^{(j)}$ is the i^{th} feature vector in cluster j .

- **Step.4:** If any cluster centroid changes the value by Step.3, go to Step.2, otherwise stop.

The above K-Mean algorithm can be used when the number of clusters k is explicitly specified. To find optimal number (k) clusters, we have employed the cluster validity analysis [32]. The idea is to find clusters that minimize intra-cluster distance while maximize inter-cluster distance. The cluster separation measure $\varphi(k)$ is defined as

$$\varphi(k) = \frac{1}{k} \sum_{i=1}^k \max_{1 \leq v \leq k} \frac{\eta_i + \eta_j}{\xi_{ij}}$$

where $\eta_j = \frac{1}{v_j} \sum_{i=1}^{v_j} D(\mathcal{F}_i^j, \mu_i)$, $\xi_{ij} = D(\mu_i, \mu_j)$. ξ_{ij} is the inter-cluster distance of cluster i and j , while η_j is

the intra-cluster distance of cluster j . The optimal number of cluster k' is selected as $k' = \min_{1 \leq k \leq q} \varphi(k)$. In other words, the K-Mean algorithm is tested for $k = 1, 2, \dots, q$, and the one which gives the lowest value of $\varphi(k)$ is chosen.

In our multi-level clustering structure, a centroid at the top level represents the category of segments in a cluster, and a centroid at the bottom level represents the general motion characteristics of a sub-cluster.

5. EXPERIMENTAL RESULTS

Our experiments in this paper were designed to assess the following performance issues:

- How does the proposed segmentation algorithm work to group incoming frames?
- How do TM , AM and the proposed algorithm work for clustering of segments?

Our test video clips were originally digitized in AVI format at 30 frames/second. Their resolution is 160×120 pixels. We used the rates of 5 and 2 frames/second as the incoming frame rates. Our test set has 111 minutes and 51 seconds of raw video taken from a hallway in a building which consist of total 17,635 frames.

5.1. Performance of Video Segmentation

A simple segmentation example can be found in Figure 5 and Table 2. The fourth and fifth columns of the table show the length (number of frames) of each segment and its category. The next two columns (Total Motion and Average Motion) will be discussed in the following subsection. The proposed segmentation algorithm discussed in section 2 was applied to our test video sequence mentioned above. As results, four different hierarchical segments are partitioned in Figure 5. The most common content of this type of video is that the objects (i.e., people, vehicles, etc.) are appearing and disappearing with various directions. The segment # 4 (Category # 2) represents this type of content in which a person is appearing and disappearing in this case.

Segment No.	Starting Frame No.	Ending Frame No.	Segment Length	Cat. (C_k)	Total Motion (TM)	Avg. Motion (AM)
1	206	219	14	2	63	4.5
2	206	214	9	3	28	3.1
3	206	211	6	4	15	2.5
4	207	209	3	5	3	1.0

Table 2: Segmentation Result for Figure 5

Table 3 shows the overall segmentation results for our test set. The second and the third columns of the table represent the number of frames per each category, and the accumulated number of frames up to the corresponding category. For example, the number, 3,871 in the row of cat. #3 indicates the sum of the number of frames (the second column) from the category # 9 to the category # 3. As seen in this table, the higher category segments can be hierarchical summaries for the lower category segments.

Category	No. of Frames	No. of Frames Accumulated	No. of Segments	Avg. No. of Frames / Segment
Cat. # 0	2877	17,635	-	-
Cat. # 1	6533	14,758	309	47.8
Cat. # 2	4354	8,225	216	38.1
Cat. # 3	3580	3,871	183	21.2
Cat. # 4	244	291	36	8.1
Cat. # 5	32	47	10	4.7
Cat. # 6	12	15	4	3.8
Cat. # 7	3	3	1	3
Cat. # 8	0	0	0	0
Cat. # 9	0	0	0	0

Table 3: Overall Segmentation Results for Test Set

5.2. Performance of TM, AM and Clustering

Before we discuss the performance of the proposed algorithm for clustering, we show some examples of TM , and AM in Table 2. Figure 7 shows $TMMI$ and $AMMI$ for the segments (#1, #2, #3 and #4) in Figure 5. Throughout this figure, we can see that the TMs and the AMs represented by $TMMIs$ and $AMMIs$ are able to measure the exact amounts(degrees) of the motions in each segment accurately.

As mentioned in the previous section, first, the segments are clustered by the categories assigned to segments. In the next level, each cluster is partitioned into smaller sub-clusters using AM . Figure 6 shows a very simple example of clustering segments. As seen in this figure, the segments are clustered by category, and further partitioned using a motion feature, AM . The different sizes of object(s) are distinguished by the category, in other words, the segments in the higher categories have relatively larger or more objects. On the other hand, the average motions, represented by AM can distinguish the amount(degree) of motions in different segments.

6. CONCLUDING REMARKS

The example of knowledge and patterns that we can discover and detect from the raw video sequences are object identification, object movement pattern recognition,

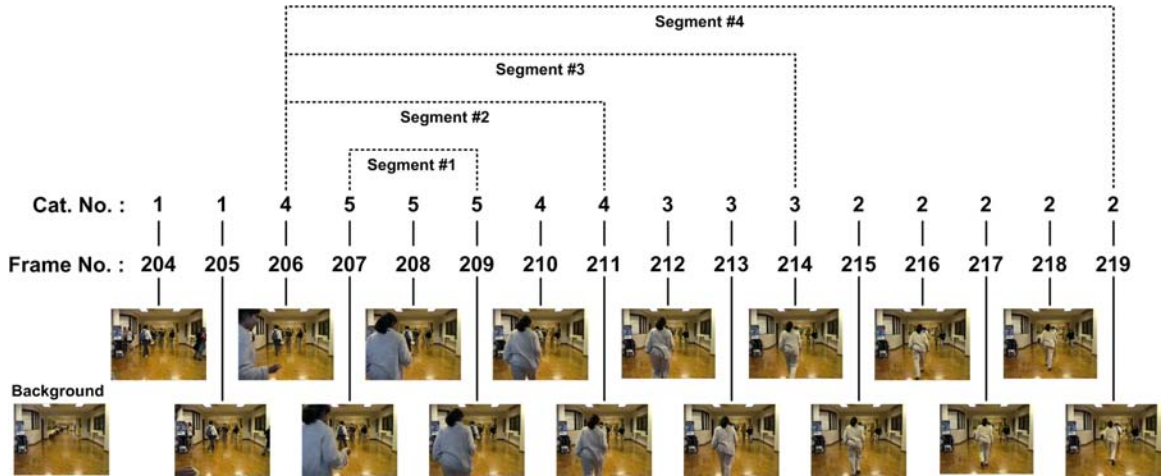


Fig. 5: Segmentation example

Category	Segments	AM
2		2.3
		1.7
3		1.9
		1.2
4		1.5
		2.0
5		1.5
		2.5

Fig. 6: Sample Clustering Results

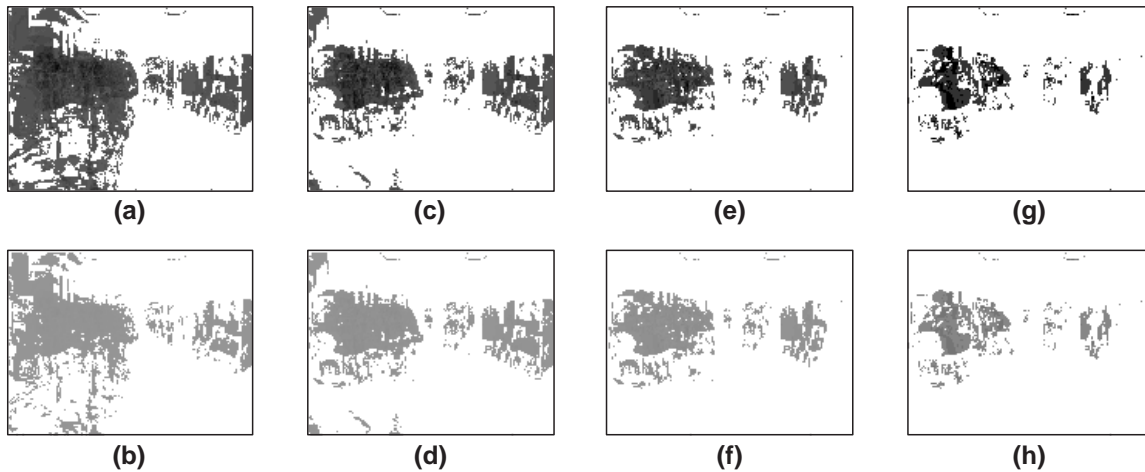


Fig. 7: (a) and (b) : TMMI and AMMI of Segment #1, (c) and (d) : TMMI and AMMI of Segment #2, (e) and (f) : TMMI and AMMI of Segment #3, and (g) and (h) : TMMI and AMMI of Segment #4

spatio-temporal relations of objects, modeling and detection of normal and abnormal (interesting) events and event pattern recognition. In this paper, we propose a general framework for this raw video data mining to perform the fundamental tasks which are temporal segmentation of video sequences, feature (motion in our case) extraction, and clustering of segments. Although our experimental data set are limited, the results are showing that the proposed framework is performing the fundamental tasks effectively and efficiently. In the future study, we will consider the other features (objects, colors) extracted from segments for more sophisticated clustering and indexing. Also, a suitability and availability of various video compression techniques including MPEG will be investigated to store these video data in database physically.

7. REFERENCES

- [1] P. Stolorz, H. Nakamura, E. Mesrobian, R. Muntz, E. Shek, J. Santos, J Yi, K Ng, S. Chien, C. Mechoso, and J. Farrara. Fast spatio-temporal data mining of large geophysical datasets. In *Proc. of Int'l Conf. on KDD*, pages 300–305, 1995.
- [2] U. Fayyad, S. Djorgovski, and N. Weir. Automating the analysis and cataloging of sky surveys. *Advances in Knowledge Discovery with Data Mining*, pages 471–493, 1996.
- [3] Z.-N Li, O.R. Zaiane, and Z. Tauber. Illumination invariance and object model in content-based image and video retrieval. *Journal of Visual Communication and Image Representation*, 1998.
- [4] D. Wijesekera and D. Barbara. Mining cinematic knowledge: Work in progress. In *Proc. of International Workshop on Multimedia Data Mining (MDM/KDD'2000)*, pages 98–103, Boston, MA, August 2000.
- [5] K. Shearer, C. Dorai, and S. Venkatesh. Incorporating domain knowledge with video and voice data analysis in news broadcasts. In *Proc. of International Workshop on Multimedia Data Mining (MDM/KDD'2000)*, pages 46–53, Boston, MA, August 2000.
- [6] V. Kulesh, V. Petrushin, and I. Sethi. The perseus project: Creating personalized multimedia news portal. In *Proc. of International Workshop on Multimedia Data Mining (MDM/KDD'2001)*, pages 31–37, San Francisco, CA, August 2001.
- [7] Y. Chen, W. Gao, Z. Wang, J. Miao, and D. Jiang. Mining audio/visual database for speech driven face animation. In *Proc. of International Conference on Systems, Man and Cybernetics*, pages 2638–2643, 2001.
- [8] P.K. Singh and A.K. Majumdar. Semantic content-based retrieval in a video database. In *Proc. of International Workshop on Multimedia Data Mining (MDM/KDD'2001)*, pages 50–57, San Francisco, CA, August 2001.
- [9] S. Chen, M. Shyu, C. Zhang, and J. Strickrott. Multimedia data mining for traffic video sequences. In *Proc. of International Workshop on Multimedia Data Mining (MDM/KDD'2001)*, pages 78–86, San Francisco, CA, August 2001.
- [10] R. Cucchiara, M. Piccardi, and P. Mello. Image analysis and rule-based reasoning for a traffic monitoring system. *IEEE Transactions on Intelligent Transportation Systems*, 1(2):119–130, June 2000.

- [11] D. Dailey, F. Cathey, and S. Pumrin. An algorithm to estimate mean traffic speed using uncalibrated cameras. *IEEE Transactions on Intelligent Transportation Systems*, 1(2):98–107, June 2000.
- [12] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easy living. In *Proc. of 3rd IEEE International Workshop on Visual Surveillance*, pages 3–10, 2000.
- [13] S. Shafer, J. Krumm, B. Meyers, B. Brumitt, M. Czerwinski, and D. Robbins. The new easy living project at microsoft research. In *Proc. of DARPA/NIST Workshop on Smart Spaces*, pages 127–130, 1998.
- [14] M. Coen. The future of human-computer interaction or how i learned to stop worrying and love my intelligent room. *IEEE Intelligent Systems*, 14(2):8–10, March 1999.
- [15] I. Pavlidis, V. Morellas, P. Tsiamyrtzis, and S. Harp. Urban surveillance systems: From the laboratory to the commercial world. *Proceedings of The IEEE*, 89(10):1478–1497, Oct. 2001.
- [16] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic monitoring and accident detection at intersections. In *IEEE International Conference on Intelligent Transportation Systems*, pages 703–708, Tokyo, Japan, 1999.
- [17] T. Huang, D. Koller, J. Malik, and G. Ogasawara. Automatic symbolic traffic scene analysis using belief networks. In *Proc. of AAAI, 12th National Conference on Artificial Intelligence (AAAI'94)*, pages 966–972, Seattle, WA, 1994.
- [18] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *Proc. of European Conference on Computer Vision*, pages 189–196, Stockholm, Sweden, 1994.
- [19] JungHwan Oh and Praveen Sankuratri. Automatic distinction of camera and objects motions in video sequences. In *To appear in Proc. of IEEE International Conference on Multimedia and Expo (ICME 2002)*, Lausanne, Switzerland, Aug. 2002.
- [20] C.W. Ngo, T.C. Pong, and H.J. Zhang. On clustering and retrieval of video shots. In *Proc. of ACM Multimedia 2001*, pages 51–60, Ottawa, Canada, Oct. 2001.
- [21] E. Ardizzone and M. Cascia. Automatic video database indexing and retrieval. *Multimedia Tools and Applications*, 4:29–56, 1997.
- [22] H. Yu and W. Wolf. A visual search system for video and image databases. In *Proc. IEEE Int'l Conf. on Multimedia Computing and Systems*, pages 517–524, Ottawa, Canada, June 1997.
- [23] R. Zabih, J. Miller, and K. Mai. A feature-based algorithm for detecting and classifying scene breaks. In *Proc. of ACM Multimedia '95*, pages 189–200, San Francisco, CA, 1995.
- [24] R. Lienhart and S. Pfeiffer. Video abstracting. *Communications of the ACM*, 40(12):55–62, December 1997.
- [25] L. Zhao, W. Qi, Y. Wang, S. Yang, and H. Zhang. Video shot grouping using best-first model merging. In *Proc. of SPIE conf. on Storage and Retrieval for Media Databases 2001*, pages 262–269, San Jose, CA, Jan. 2001.
- [26] S. Han and I. Kweon. Shot detection combining bayesian and structural information. In *Proc. of SPIE conf. on Storage and Retrieval for Media Databases 2001*, pages 509–516, San Jose, CA, Jan. 2001.
- [27] JungHwan Oh, Kien A. Hua, and Ning Liang. A content-based scene change detection and classification technique using background tracking. In *SPIE Conf. on Multimedia Computing and Networking 2000*, pages 254–265, San Jose, CA, Jan. 2000.
- [28] JungHwan Oh and Kien A. Hua. An efficient and cost-effective technique for browsing and indexing large video databases. In *Proc. of 2000 ACM SIGMOD Intl. Conf. on Management of Data*, pages 415–426, Dallas, TX, May 2000.
- [29] Kien A. Hua and JungHwan Oh. Detecting video shot boundaries up to 16 times faster. In *The 8th ACM International Multimedia Conference (ACM Multimedia 2000)*, pages 385–387, LA, CA, Oct. 2000.
- [30] JungHwan Oh and Tummala Chowdary. An efficient technique for measuring of various motions in video sequences. In *To appear in Proc. of The 2002 International Conference on Imaging Science, System, and technology (CISST'02)*, Las Vegas, NV, June 2002.
- [31] P.J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, 1987.
- [32] A. K. Jain. *Algorithm for Clustering Data*. Prentice Hall, 1988.

AN INNOVATIVE CONCEPT FOR IMAGE INFORMATION MINING

MIHAI DATCU

German Aerospace Center - DLR
Remote Sensing Technology Institute - IMF
Oberpfaffenhofen, D-82234 Wessling

Tel: +49 8153 28 1388

Fax: +49 8153 28 1444

Email: mihai.datcu@dlr.de

KLAUS SEIDEL

Remote Sensing Group
Computer Vision Lab ETH
CH 8092 Zürich - SWITZERLAND

Tel: +41 1 632 5284

Fax: +41 1 632 1251

Email: seidel@vision.ee.ethz.ch

Abstract

Information mining opens new perspectives and a huge potential for information extraction from large volumes of heterogeneous images and the correlation of this information with the goals of applications.

We present a new concept and system for image information mining, based on modelling the causalities which link the image-signal contents to the objects and structures within interest for the users. The basic idea is to split the information representation into four steps:

1. image feature extraction using a library of algorithms such to obtain a *quasi-complete* signal description
2. unsupervised grouping in a large number of clusters to be suitable for a large set of tasks
3. data reduction by parametric modelling the clusters
4. supervised learning of user semantics, that is the level where, instead of being programmed, the systems is trained by a set of examples; thus the links from image contents to the users are created.

The record of the sequence of links is a knowledge acquisition process, the system memorizes the user hypotheses. Step 4. is a man-machine dialogue, the information exchange is done using advanced visualization tools. The system learns what the users need.

The system is presently prototyped for inclusion in a new generation of intelligent satellite ground segment systems, value adding tools in the area of geoinformation, and several applications in medicine and biometrics are also foreseen.

Key words

information mining, data mining, CBIR

Preliminaries

The image archives are heterogeneous, huge data repositories, they are high complexity sources of valuable information, e.g. the Earth Observation data archives contain millions of optical, radar and other types of images and data. The exploration of their content is not an easy task. Among the promising methods proposed in the last years are the methods of data and information mining. However, accessing the image information content, in comparison with other data types, is rising higher complexity problems, residing mainly in the huge volume of data, the rich information content, and the subjectivity of the user interpretation. The present article makes an analysis of the Image Information Mining methods seen as an information transmission problem: the source of information is an image archive, the receiver is the community of users. Data and information mining are exploratory processes focusing on the techniques for analyzing and combining raw data and detecting patterns and regularities within the data set. The success of the exploratory information search depend on the capacity to capture and describe the full complexity of the data. Thus we use a concept integrating multiple methods: information theory, stochastic modelling, Bayesian inference, machine learning. Information theory deals with encoding data in order to transmit it correctly and efficiently. The theory of stochastic processes and machine learning deal with estimating models of data and predicting future observations. There is a relationships between these fields: the most compact encoding of the data is by the probabilistic model that describes it best, thus there is a fundamental link between information and probabilistic models. This link is the basic to implement optimal algorithms for information extraction, detecting causalities, and for the design of information systems implementing image information mining functions. The

article presents and analysis several methods for mining the information content of large image repositories, and exemplifies image mining functions, like, search by example, search by data model, exploration in the scale space and image complexity, knowledge acquisition, and adapting to the user conjecture.

1. From content based image retrieval to mining the image information

The continuously expansion of multimedia in all sectors of activity is facing us with a double explosion:

- the number of image data sets
- the data size and information variability of each image

e.g. with a digital camera we can acquire 10 Gb of images during a 3 weeks holiday, a satellite sensor can acquire 100 Gb per day.

Thus, since many years, it is known that classical image file text annotation is prohibitive for large data bases. The last decade is marked by important research efforts to develop Content based Image Retrieval (CBIR) concepts and systems [11]. Images in an archive are searched by their *visual* similarity with respect to color, texture or shape characteristics. While image size and information content is continuously growing CBIR was not any more satisfactory and Region Based Information Retrieval (RBIR) has been developed [11]. Each image is segmented and individual *objects* are indexed by primitive attributes like color, texture and shape. Thus, RBIR is a solution to deal with the variability of image content.

However, both CBIR and RBIR have been *computer centered* approaches, i.e. the concepts could only little or not at all adapt to the user needs. Further, the image retrieval systems have been equipped with *relevance feedback* functions [1]. The systems are designed to search images similar to the user conjecture. The algorithms are based on analyses of the probabilities of an image to be the search target. A feedback which takes this part into account is introduced.

Another interesting approach was developed based on a learning algorithm to select and combine feature grouping and to allow users to give positive and negative examples. The method refines the user interaction and enhances the quality of the queries [8].

Both previously mentioned concepts are first steps to include the user in the search loop, they are *information mining concepts*. Also, these are methods in the trend of designing *human centered systems*.

2. Images and image information

Compared with *Data Mining* the field of *Image Information Mining* reaches much higher complexity resulting from:

- the huge volume of data (Tb to Pb)
- the variability and heterogeneity of the image data (diversity of sensors, time or conditions of acquisition, etc)
- the image content, its meaning is many times *subjective*, depending to the users interest
- the large range of user interest, semantics and contextual (semiotic) understanding.

In general, by *image* we understand *picture* thus relating it to the (human) visual perception and understanding. A picture is characterized by its primitive features such as color, texture, shape at different scales. Its perception and understanding is in form of symbols and semantics in a certain semiotic context [12].

However, the concept of image is beyond the pictorial understanding. Images are multidimensional signals, like computer tomography, hyperspectral images or results of simulations. They are communicated to users via 2-dimensional visual projections. Thus images can contain quantitative, objective information, as acquired by an instrument.

In Fig 1 an example is presented for the visualization of a data set of a Digital Terrain Model (DEM) in comparison with a color rendered satellite image of the same Alpine region. The *visual* information in the DEM image is not easy to read. The information of terrain elevation is contained in the image samples. The color image, however, shows the complexity of pictorial information.

In the perspective of image information mining both the types of images, pictorial and multidimensional signals rise the same problematic. Their understanding depends on the accuracy of:

- information content modelling
- modelling the users understanding.

Thus, image information mining can be seen as a communication task. The source of information is the large heterogeneous image archive. The receiver is the community of users. The accuracy of communication the, i.e. the success of finding the information needed as exploration results, depends on the accuracy of the previously assumed levels of modelling.

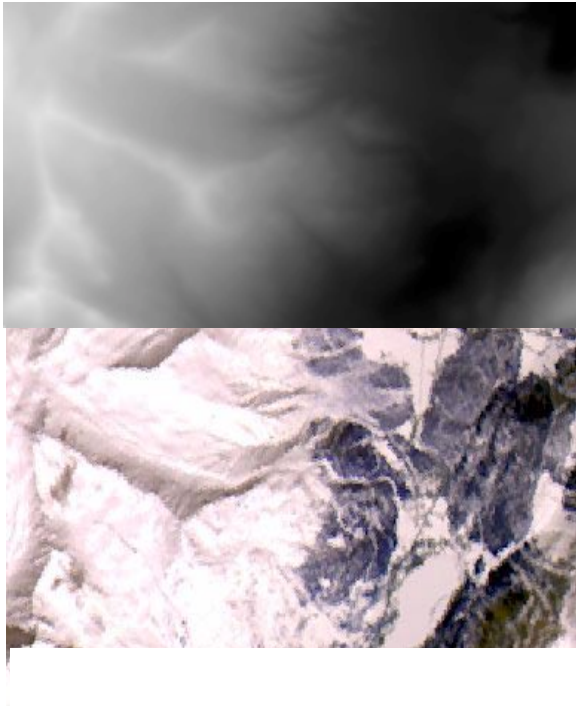


Figure 1. Top: Visualization of a digital Elevation Model DEM data set of Davos, Switzerland. The information on terrain height is contained in the pixel intensity, the information is quantitative and is not rich in visual meaning.

Bottom: Satellite image (Landsat TM) of the same area. The information is pictorial, aggregation of colors, textures and geometrical objects at different scales makes it possible to understand the scenery of an alpine ski resort.

3. Information mining: concept and system

We developed a theoretical concept for image information representation and adaptation for the user conjecture [2,3,4,6,7]. A quasi-complete description of the image content is obtained by utilization of a library of models. The feature extraction is equivalent with splitting the image content in different information channels. An unsupervised clustering is done for each information channel as an information encoding and data reduction operation. Then, during the operation of the system, an interactive learning process allows the user to create links, i.e. to discover conditions between the low-level signal description and the target of the user. .

The image features reflect the physical parameters of the imaged scene, thus, assuming the availability of certain models, the scene parameters can be extracted. For example, color and image texture carries information about the structure of object surfaces. However, in the case of modelling high complexity signals, a large number of sources

coexists within the same system, thus multiple candidates models are needed to describe the information sources in the image. Also, to reduce complexity, to capture the class structure, and discover causalities and to provide computational advantages, the models are likely to be analyzed hierarchically. The hierarchical information representation is further presented and depicted in Fig. 2:

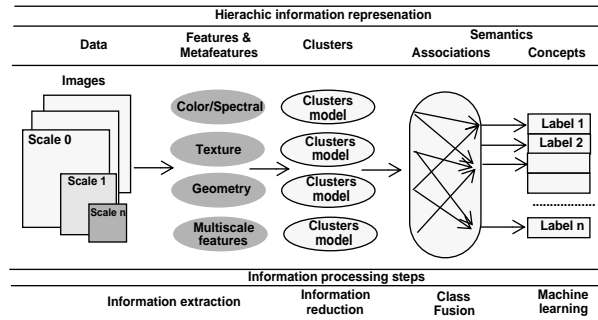


Figure 2. The hierarchical representation of the image information content, and the causalities to correlate the user conjecture to the image content. The key elements are: the quasi-complete image signal description by extraction of the elementary features, the data reduction by clustering, thus inducing also a measure of some similarity over the feature space, the utilization of the cluster models as elements of an abstract vocabulary which in an interactive learning process enables to learn the semantics of the target and the user conjecture.

- **Image data:** the information is contained in the samples of the raw data. It is the lowest level of information representation.
- **Image features:** the performance of information extraction depends critically on the descriptive or predictive accuracy of the probabilistic model employed. Accurate modelling typically requires high-dimensional and multi-scale modelling. For non-stationary sources, accuracy also depends on adaptation to local characteristics. For a quasi-complete characterization the image content, information is extracted in form of parameters characterizing the: color or spectral properties, texture as interactions among spatially distributed samples, the geometrical attributes of image objects.
- **Meta features:** estimation of the image features, requires the assumption of some data models. The type of model used, its evidence and complexity, plays the role of meta information, i.e. describing the quality of the extracted parameters. From a data aggregation perspective, a meta feature is an indicator of information commensurability, e.g. estimated texture features using cooccurrence matrix are not comparable with parameters

of Markov random fields. The meta features have semantic value.

- **Cluster model:** the signal features have n-dimensional representations. Due to observation noise or model approximations the feature space is not occupied homogeneously. Thus, another level of information abstraction is the type of feature grouping, i.e. the cluster models, and the associated parameters. The obtained clusters represent information only for each category of the features.
- **Semantic representation:** it is known that the distinction between the perception of information as signals and symbols is generally not dependent on the form in which the information is presented but rather on the conjecture in which it is perceived, i.e. upon the hypothesis and expectations of the user. Augmentation of data with meaning requires a higher level of abstraction. The extracted information, represented in form of classes is fused in a supervised learning process. Prior information in form of training data sets or expert knowledge is used to create semantic categories by associations to different information classes. Thus, the observations are labelled and the contextual meaning is defined.

In order to implement the hierarchical representation of the image information content, the data are pre-processed. The image features are extracted for different image scales. In the next processing step the image features are clustered, and further a signal content index is created using the cluster description, the scale information, and the type of stochastic model assumed for the image parameters. A Bayesian learning algorithm allows a user to visualize and to encapsulate interactively his prior knowledge of certain image structures and to generate a supervised classification in the joint space of clusters, scales, and model types. The index of each image pixel is encoded by the spatial correspondence of the class information. The user is enabled to attach his meaning to similar structures occurring in different images, thus adding a label in the archive inventory. This label is further used to specify queries. The hierarchical information, meta-information, associations and semantic labels are stored and managed by a Data Base Management System. The system is implemented in a server-client architecture as presented in Figure 3.

This concept was implemented and successfully demonstrated with an on-line experimental system, see <http://isis.dlr.de/mining>. The novel *mining* functions presently provided by the system are further presented.

3.1. Semantic Content Based Image Retrieval

Following an automatic processing at data ingestion or in a semi-automatic manner using an interactive learning process, the system can create links between the concept level and the image data and cluster levels.

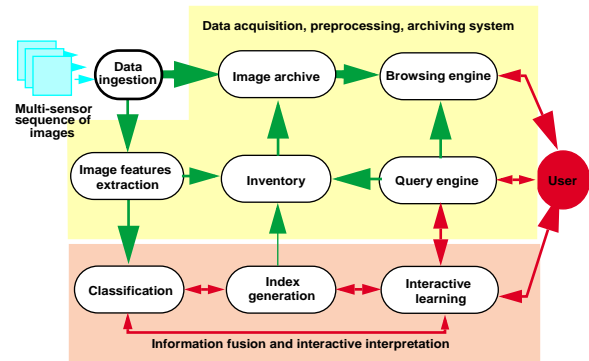


Figure.3: The system architecture. In yellow the server, violet the client.

The user is enabled to specify semantic queries at concept level and the system is returning all images with the specified content and a classification on individual images. An example is given in Fig. 4.

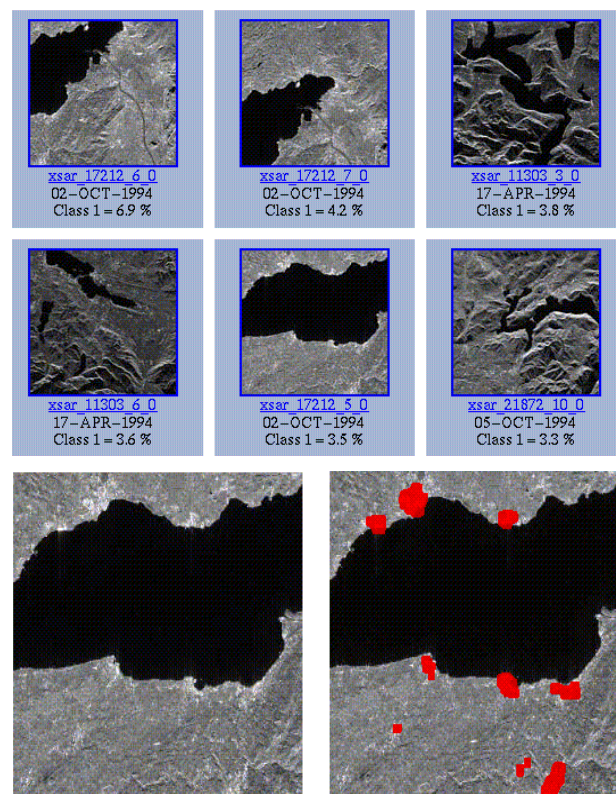


Figure 4. Top: Result of semantic query - discovering settlements. The images have been automatically analyzed at ingestion in the archive, and a catalogue entry was created for all images containing build up areas. Bottom: Each image has attached the result of the classification, the regions marked in red correspond to villages and cities, thus the result of the query is the list of images, augmented with the expected semantic image content. Synthetic Aperture Radar X-SAR SRL images of Switzerland.

In the case of Earth Observation the geographical location is also used as meta-information allowing to find the location of the *intensity* images as indicated in Fig. 5.

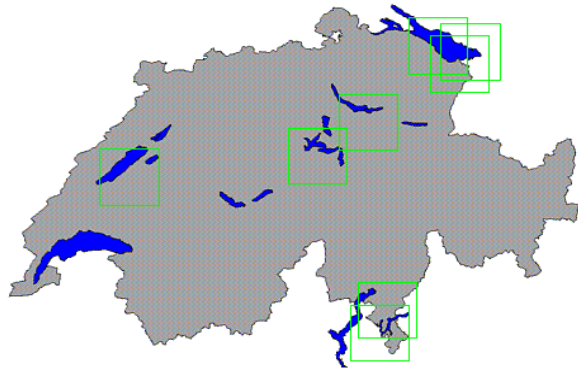


Figure 5. The geographical location of the images obtained as result of a semantic query (Fig. 3).

3.2. Mining driven by primitive signal features

The mining driven by primitive signal features, such as spectral signatures or structural patterns, is enabled by the exploration of the links between the cluster and image data levels. Examples of spectral and textural signature mining is depicted in Fig. 6. The spectral mining is an example of physical, quantitative model exploration. For the Landsat-TM images used for exemplification only 6 spectral bands have been selected.

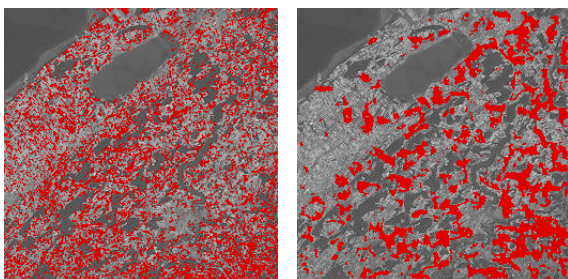


Figure 6. Left: Spectral image content, in red, obtained by the correlation of a specified cluster model with the pixel position in the image. Right: Texture image content obtained in similar manner, however, the textural information characterizes structures, thus the resulting classification has connected areas. The information is indexed enabling to discover all images with similar spectral or textural properties. Landsat TM image of Switzerland.

3.3. Mining information theoretical measures

In the exploration of large image archives with rich information content it is important to group the data according to various objective information measures. That helps the users to *orient* within the search process.

One important characteristic is the scale at which relevant information is concentrated. We used an multiscale stochastic process for automatic scale detection and segmentation [9,10]. An example is shown in Fig. 7. The exploration of image archives by scale is a process which is implicitly using a priori knowledge assumed by the user: the ratio of the image resolution and size of objects he is searching for.

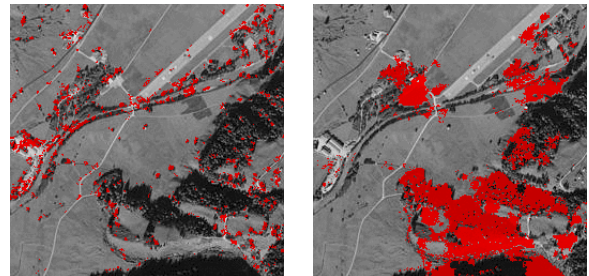


Figure 7. Left: structures correspond to a fine scale. Right: In the same image structures corresponding to a rougher scale. The scale of structures in images is a fundamental descriptor, both in relation with the visual interpreting, and objectively in relation with the resolution of the sensor. The parameters of a multi-scale random field are used to automatically detect the relevant scales. The information is indexed enabling to discover all images with structures at similar scales. Aerial photography.

The *complexity* of the images is another information theoretical measure used to rank images. The complexity is defined as the Kullback-Leiber divergence between the cluster level and the image data level. The complexity depends on the quality and type of model used. In Fig. 8 examples of ranking images are presented according to their spectral and textural complexity.



Figure 8. Top: Example of images of low (left) and high (right) spectral complexity. Bottom: Example of images of low (left) and high (right) structural complexity. The complexity of the images was measured as Kullback-Leiber entropy at the classification and clustering levels in the information hierarchy. The low complexity images are poor in information content, high complexity images show more “activity” thus giving a better chance to discover “interesting” structures, or objects. The complexity values are indexed enabling to discover all images with similar behavior.

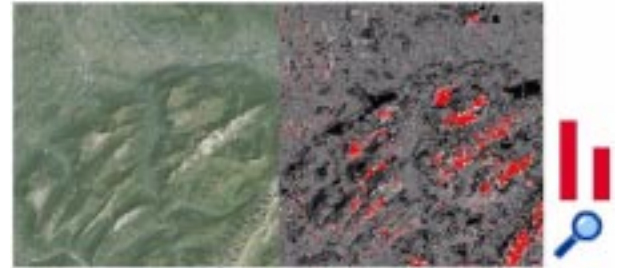
3.4. Mining by interactive learning

The interactive learning is the process to discover the links between the user interest (target), the image content in terms of describing models and the images containing the assumed structure[3,7]. In a first step the interactive learning uses a Bayesian network to create the links between the concept and cluster levels. During the interactive learning the image data (quicklooks) are used to give examples and to index the spatial position of the target structures. In a second step, also using a Bayesian approach, a probabilistic search over the image space is performed. At this stage the links between the concept level, clusters and image data levels are created. The learning process is using positive and negative examples, both from the user and machine site. It is a man-machine dialog.

In Fig. 9 an example is presented for the exploration of different models (texture at various scales and spectral signatures) to discover different semantic objects in the data.

Online training of **new_label** using **two features** (gmrf14.:spectr6.)

Cover-type name:



Online training of **alps** using **two features** (gmrf11.WS03:gmrf11.WS04)

Cover-type name:

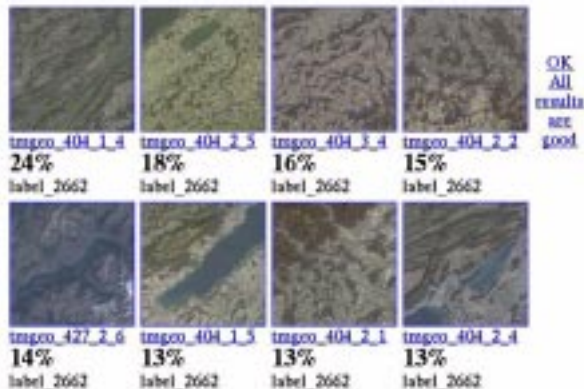


Figure 9. Top: Interactive training using fusion of spectral and textural information at the finest image scale. The target semantics is “meadow”. Bottom: On the same image, interactive training using fusion of texture information estimated for scales 1:2 and 1:3, the target semantics is “mountain”. The interactive learning is an information mining process able to adapt to the user conjecture. It is a pure exploratory function based on learning, fusion, and classification processes, using the pre-extracted image primitive attributes, and allowing an open, very large semantic space. The user defined target is generalized over the entire image archive, thus allowing further exploration.

The results of the probabilistic search are depicted in Fig. 10 for the cases indicated in Fig. 9.

Probabilistic Search Result for **label_2662**

A. Images with **highest coverage** of **label_2662** (Click on an image to continue learning on it):



Probabilistic Search Result for **alps**

A. Images with **highest coverage** of **alps** (Click on an image to continue learning on it):

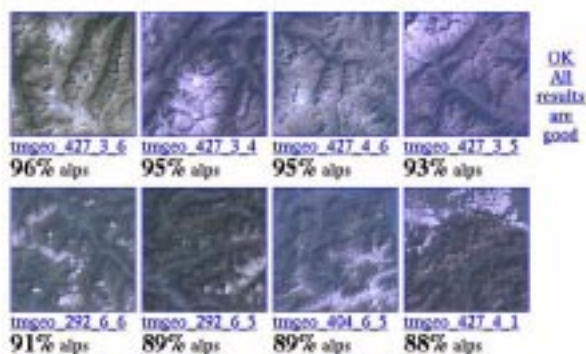


Figure 10. Top: the result of probabilistic search for images containing “meadow”. Bottom: the result of probabilistic search for images containing “mountains”. Both query results correspond to the interactive training as defined in Fig. 9.

3.5. Knowledge driven image information mining and user conjecture

During the interactive learning and probabilistic search the database management system (DBMS) holds a record of:

- the user semantic
- the combination of models able to explain the user’s target
- the classification of the target structure in each individual image
- a set of statistical and information theoretical measures of goodness of the learning process.

This information and associations represent a body of knowledge, either discovered or learned from the various system users. This information is further being used for other mining tasks. This acquired and learned information and knowledge is itself object of mining, e.g. grouping of semantic levels, relevance feedback, joint grouping between the semantic space and the statistical or information theoretical measures of goodness of the learning process.

4. Conclusions

We based and developed a new concept for image information mining. We regard the mining process as a communication task, from a user centered perspective. The hierarchy of information representation, in conjunction with the quasi-complete image content description, enables implementation of a large variety of mining functions. The concept was demonstrated for a variety of Earth Observation data. Further work is done for the development of intelligent satellite ground segment systems, and value adding tools. However its potential is broader, other fields of applications are possible, such as medical imagery, biometrics, etc.

The proposed concept is far away from being fully exploited. Presently ongoing theoretical development is approfondating the problematic of image complexity. In the case of high heterogeneity observations the complexity and the course of dimensionality are two key issues which can hinder the interpretation. Therefore, as an alternative solution to the “interpretation”, we propose an exploratory methodology approached from a information theoretical perspective in a Bayesian frame.

Another direction is the analysis of cluster models from the perspective of an “objective” semantic approach, aiming at the elaboration of methods to understand the nature of the feature space.

A direction of application of the developed methodology is the mining of temporal series of images, considering the integration of spatio-temporal signal analysis.

Even the concept of learning the user conjecture was at some extent demonstrated. Difficult problems are further under research, such developing image grammars and representation of image content in different contextual environments. This is a semantic problem which can arise between different users when they define or describe the same structures differently, requiring the primitive attributes, features, domains, values, or causalities to be translated.

A number of challenges, mainly in the design of multidimensional DBMS, man-machine interfaces, distributed information systems, will probably be approached soon.

ACKNOWLEDGEMENT

The project has been supported by the Swiss Federal Institute of Technology (ETH) Research Foundation *Advanced Query and Retrieval Techniques for Remote Sensing Image Archives* (Grant: RSIA 0-20255-96). The author would like to thank Michael Schröder and Hubert Rehrauer for converting the concept into algorithms and setting up the Multi-Mission Demonstrator (MMDEMO).

REFERENCES

- [1] I.J. Cox, M.L. Miller, S.M. Omohundro and P. N. Yianilos, 1996, "PicHunter: Bayesian Relevance Feedback for Image Retrieval," Proc. Int. Conf. on Pattern Recognition, Vienna, Austria.
- [2] M. Datcu, K. Seidel, M. Walessa, 1998, *Spatial Information Retrieval From Remote Sensing Images: Part I. Information Theoretical Perspective*, IEEE Tr. on Geoscience and Remote Sensing, Vol. 36, pp. 1431-1445.
- [3] M. Datcu, K. Seidel, G. Schwarz, 1999, *Elaboration of advanced tools for information retrieval and the design of a new generation of remote sensing ground segment systems*, in I. Kanellopoulos, editor, Machine Vision in Remote Sensing, Springer, pp. 199-212.
- [4] M. Datcu, K. Seidel, 1999, *Bayesian methods: applications in information aggregation and data mining*. International Archives of Photogrammetry and Remote Sensing, Vol. 32, Part 7-4-3 W6, pp. 68-73.
- [5] M. Datcu, K. Seidel, S. D'Elia, P. G. Marchetti, 2002, *Knowledge-driven Information-Mining in remote sensing image archives*, ESA Bulletin.
- [6] M. Schröder, H. Rehrauer, K. Seidel, M. Datcu, 1998, *Spatial Information Retrieval From Remote Sensing Images: Part II. Gibbs Markov Random Fields*, IEEE Tr. on Geoscience and Remote Sensing, Vol. 36, pp. 1446-1455.
- [7] M. Schröder, H. Rehrauer, K. Seidel, M. Datcu, 2000, *Interactive learning and probabilistic retrieval in remote sensing image archives*, IEEE Trans. on Geoscience and Remote Sensing, Vol. 38, pp. 2288-2298
- [8] T. P. Minka, R. W. Picard, 1997, *Interactive learning with a society of models*. Pattern Recognition, vol. 30, pp.565-581.
- [9] H. Rehrauer, K. Seidel, M. Datcu, 1999, *Multi-scale indices for content-based image retrieval*. in Proc. of 1999 IEEE International Geoscience and Remote Sensing Symposium IGARSS'99, volume V, pp. 2377-2379.
- [10] H. Rehrauer, M. Datcu, 2000, *Selecting scales for texture models*, In Texture analysis in machine vision, ed.: M.K. Pietikäinen, Series in machine perception and artificial intelligence, vol. 40, World Scientific.
- [11] C. R. Veltkamp, H. Burkhardt, H.-P. Kriegel (eds.). 2001, *State-of-the-Art in Content-Based Image and Video Retrieval*. Kluwer.
- [12] Ji Zhang, Wynne Hsu, Mong Li Lee, 2001, *Image Mining: Issues, Frameworks and Techniques*, in Proceedings of the Second International Workshop on Multimedia Data Mining (MDM/KDD'2001), San Francisco, CA, USA, August, 2001.

MULTIMEDIA DATA MINING USING P-TREES^{1, 2}

WILLIAM PERRIZO, WILLIAM JOCKHECK, AMAL PERERA, DONGMEI REN, WEIHUA WU, YI ZHANG

North Dakota State University
Fargo, North Dakota 58105
william.perrizo@ndsu.nodak.edu

ABSTRACT

The DataSURG group at NDSU has a long-standing interest in data mining remotely sensed imagery (RSI) for agricultural, forestry and other prediction and analysis applications. A spatial data structure, the Peano count tree, was developed that provided an efficient, lossless, data mining ready representation of the many types of data involved in these applications. This data structure has made possible the mining of multiple very large data sets, including time-sequence of RSI and multimedia land data. The Peano count tree (P-tree) technology provides an efficient way to store and mine images of any format, together with pertinent land data of still other formats.

With the invention of Gene chips and gene expression microarrays (MA data) for use in medicine, plant science and many other application areas, new multimedia data mining challenges appeared. MA data presents a one-time, gene expression level map of thousands of genes subjected to hundreds of conditions. An important multimedia plant science application of the near future is to integrate macro-scale analysis of RSI with the micro-scale analysis of MA and to do the latter across multiple organisms. Most of the MA research has been done for a particular organism and the results have been archived as text abstracts (e.g., Medline abstracts). It will therefore be necessary to combine text mining with most multimedia RSI and MA mining. This is truly a multimedia data mining setting. The way text is almost always mined today is to extract pertinent features into tables and to then mine the tables (i.e., extract structured records from the unstructured text first). P-trees are a convenient technology to mine all media involved in this research.

In fact, in almost all multimedia data mining applications, feature extraction converts the pertinent data to relational or tabular form, and then the tuples or rows are

data mined. If multi-medias are going to be mined by first converting to a common format or media, a good candidate common data structure for that purpose is the P-tree. The P-tree data structure is designed for just such a data mining setting.

Keywords

Spatial - Temporal Data Mining, Multimedia, P-tree

1 INTRODUCTION

Data mining often involves handling large volumes of data. However, over the years the concept of what was a large volume of data has evolved. Problems that simply were considered intractable are now taken on with optimism. Spatial-temporal data and other multimedia data are examples where data mining is beginning to be effectively applied.

The DataSURG group at NDSU came to data mining from the context of evaluation of remotely sensed images for use in agricultural applications. These projects involved evaluation of remote imagery of agricultural fields combined with other data sets to produce yield projections. A typical data set might be composed of 1.7 million grid points in a field, each with up to 6 values associated with it.

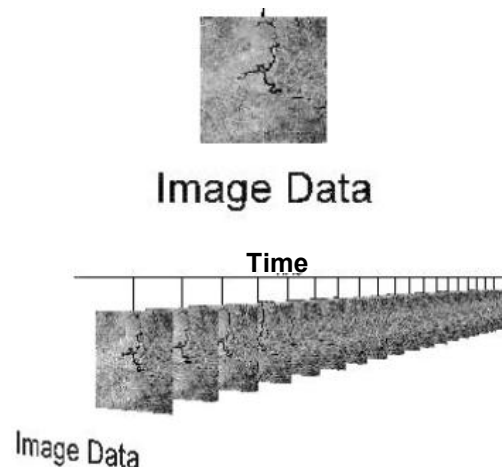


Figure 1: Image data sequenced in the time dimension

¹ Patents are pending on the bSQ and P-tree technology.

² This work is partially supported by GSA Grant ACT# K96130308, NSF Grant OSR-9553368 and DARPA Grant DAAH04-96-1-0329.

Initially these sets were considered large but advances in computer technology and the development of P-tree technology made the sets easily manageable. As more and more data was incorporated the concept of mining sequences of these images developed.

These tools that had been applied to layers of data from different sources are now being viewed as a way to handle sequences of large data sets as they arrive. These data sets do not need to be images but can be stored using the same structures to expedite access.

The purpose of this paper then is to establish that the techniques originally developed for RSI data can provide a major contribution to multimedia data mining. To this end the paper first examines several multimedia data mining approaches to determine their common elements. This element is the production of high dimensional, sparse feature space. This common factor provides the opportunity to use the P-tree technology that is then presented. The use of this technology provides a method to apply multiple data mining techniques to the feature space.

1.1 Multimedia Data Mining

Multimedia data mining is the mining of high-level multimedia information and knowledge from large multimedia databases [10]. It includes the construction of multimedia data cubes which facilitate multiple dimensional analysis of multimedia data and the mining of multiple kinds of knowledge, including summarization, classification and association.

The common characteristic in many data mining applications, including many multimedia data mining applications is that, first, specific features of the data are captured as feature vectors or tuples in a table or relation and then tuple-mined.

There are some examples of multimedia data mining systems. IBM's Query by image content [10] and MIT's Photo book extract image features such as color histograms hues, intensities, shape descriptors, as well as quantities measuring texture. Once these features have been extracted, each image in the database may now be thought of as a point in this multidimensional feature space (one of the coordinates might, for the sake of a simplistic example, correspond to the overall intensity of red pixels, and so on).

Another example is MultiMediaMiner [10]. MultiMediaMiner is a system prototype for multimedia data mining which applies multi-dimension database structures, attribute-oriented induction, multi-level association analysis, statistical data analysis, and machine learning approaches for mining different kinds of rules in relational databases and data warehouses. The system contains 4 major components: image excavator for the extraction of images and videos from multimedia

repositories, a processor for the extraction of image features and storing precomputed data in database, a user interface, and a search kernel for matching queries with image and video feature in the database.

1.1.1 Video-Audio Data Mining

The high dimensionality of the feature spaces and the size of the multimedia datasets make meaningful multimedia data summarization a challenging problem. Video-Audio data mining and other multimedia data mining often involves a preliminary feature extraction step in which the pertinent data is formed into a relation of tuples or possibly time series of tuples, each tuple describing specific selected features of a "frame". P-tree provides a common structure for multi-media data set, which facilitates multimedia data mining.

The process of audio-video multimedia data mining goes as follows:

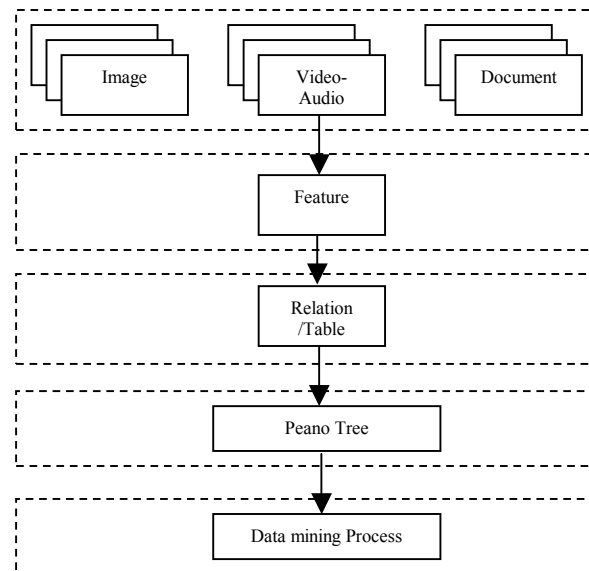


Figure 2 process of video-audio multimedia data mining

For example, performing face recognition from video sequences, involves first extracting specific face geometry attributes (e.g., relative position of nose, eyes, chinbones, chin, etc.) and then forming a tuple of those geometric attributes. Faces are identified by comparing face-geometric features with those stored in a database for known individuals. Partial matches allow recognition even if there are glasses, beards, weight changes, etc. There are many applications of face recognition technology including surveillance, digital library indexing, secure computer logon, and airport and banking security [15].

Another multimedia data mining example is voice biometrics [15]. It relies on human speech, one of the primary modality in human-to-human communication, and provides a non-intrusive method for authentication. By extracting appropriate features from a person's voice and

forming a vector or tuple of these features to represent the voiceprint, the uniqueness of the physiology of the vocal tract and articulator properties can be captured to high degree and used very effectively for recognizing the identity of the person.

1.1.2 Text mining

Text mining can find useful information from unstructured textual information like letters, emails and technical documents. But these kinds of unstructural textural information are not ready for data mining. [8]

Text mining generally involves the following two phases:

1. Preparation phase: document representation
2. Processing phase: clustering or classification

In order to apply data mining algorithms to text data, a weighted feature vector is typically used to describe a document. These feature vectors contain a list of the main themes or keywords or wordstems, along with a numeric weight indicating the relative importance of the theme or term to the document as a whole [9]. The feature vectors are usually highly dimensional, but sparsely populated [8]. P-trees are well suited for representing such feature vector sets. After the mapping of documents to feature vector tables or relations, we can perform document classification in either of two ways: tuple clustering or tuple classification.

1.2 Multimedia Summary

In summary, the key point of this discussion is that a large volume of multimedia data is typically preprocessed into some sort of representation in a high dimension feature space. These feature spaces usually take the form of a table or relation. The data mining of multimedia data then becomes a matter of row or tuple mining (clustering or classification) of the feature tables or relations. While this paper does not propose new techniques for the process of feature extraction, but does propose a new approach to the storage and processing of the feature space, once it is created. Good multimedia representations and formats can help a lot. In the next section of this paper, we describe a technology for storing and mining multimedia feature spaces efficiently and accurately.

2 Peano Count Trees (P-trees)

In this section, we discuss a data structure, called the Peano Count Tree (or P-tree), and its algebra and properties. First, we note again that in most multimedia data mining applications, feature extraction is used to convert the raw multimedia data to relational or tabular form, and then the tuples or rows are data mined. The P-

tree data structure is designed for just such a data mining setting. P-trees provide a lossless, compressed, data mining-ready representation of the relational data set [7].

Given a relational table (with ordered tuples or rows), the data can be organized in different formats. BSQ, BIL and BIP are three typical formats. The Band Sequential (BSQ) format is similar to the relational format. In BSQ format, each attribute is stored as a separate file and each individual band uses the same tuple ordering. Thematic Mapper (TM) satellite images are in BSQ format. For images, the Band Interleaved by Line (BIL) format stores the data in line-major order, i.e., the first row of all bands, followed by the second row of all bands, and so on. SPOT images, which come from French satellite platforms, are in BIL format. Band Interleaved by Pixel (BIP) is a pixel-major format. Standard TIFF images are in BIP format.

We propose a new generalization of BSQ format called bit Sequential (bSQ), to organize any relational data set with numerical values [7]. We split each attribute into separate files, one for each bit position. There are several reasons why we use the bSQ format. First, different bits make different contributions to the values. In some applications, the high-order bits alone provide the necessary information. Second, the bSQ format facilitates the representation of a precision hierarchy. Third, bSQ format facilitates compression. P-trees are basically quadrant-wise, Peano-order-run-length-compressed, representations of each bSQ file. Fast P-tree operations, especially fast AND operation, provide the possibilities for efficient data mining.

In Figure 3, we give a very simple illustrative example with only two bands in a scene having only four pixels (two rows and two columns). Both decimal and binary reflectance values are given. We can see the difference of BSQ, BIL, BIP and bSQ formats.

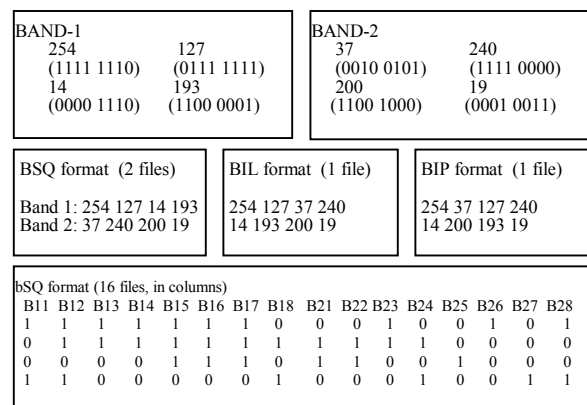


Figure 3 BSQ, BIP, BIL and bSQ formats for a two-band 2x2 image

2.1 Basic P-trees

In this subsection we assume the relation is the pixel relation of an image so that there is a natural notion of rows and columns. However, for an arbitrary relations or table, we can consider the row order to be Peano order (in 1-D, 2-D, 3-D or higher dimensions) and achieve the very same result. Using an X-Y image is the simplest setting in which to introduce the idea of P-trees.

Given a Relation that has been decomposed into bSQ format, we reorganize each bit file of the bSQ format into a tree structure, called a Peano Count Tree (P-tree). The idea is to recursively divide the entire image into quadrants and record the count of 1-bits for each quadrant, thus forming a quadrant count tree [7]. P-trees are somewhat similar in construction to other data structures in the literature (e.g., Quadrees [3, 4, 5] and HHcodes [6]).

For example, given a 8x8 bSQ file (one-bit-one-band file), its P-tree is as shown in Figure 4.

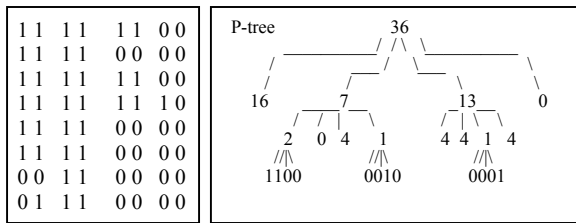


Figure 4 P-tree for a 8x8 bSQ file

In this example, 36 is the number of 1's in the entire image, called root count. This root level is labeled level 0. The numbers 16, 7, 13, and 0 at the next level (level 1) are the 1-bit counts for the four major quadrants in raster order. Since the first and last level-1 quadrants are composed entirely of 1-bits (called pure-1 quadrants) and 0-bits (called pure-0 quadrants) respectively, sub-trees are not needed and these branches terminate. This pattern is continued recursively using the Peano or Z-ordering (recursive raster ordering) of the four sub-quadrants at each new level. Eventually, every branch terminates (since, at the "leaf" level all quadrants are pure). If we were to expand all sub-trees, including those for pure quadrants, then the leaf sequence would be the Peano-ordering of the image. The Peano-ordering of the original image is called Peano Sequence. Thus, we use the name Peano Count Tree for the tree structure above.

The fan-out of a P-tree need not be fixed at four. It can be any power of 4 (effectively skipping levels in the tree). Also, the fan-out at any one level need not coincide with the fan-out at another level. The fan-out pattern can be chosen to produce maximum compression for each bSQ file. We use P-Tree-r-i-l to indicate the fan-out pattern, where r is the fan out of the root node, i is the fan out of all internal nodes at level 1 to L-1 (where root has level L, and

leaf has level 0), and l is the fan out of all nodes at level 1. We have implemented P-Tree-4-4-4, P-Tree-4-4-16, and P-Tree-4-4-64.

Definition 1: A basic P-tree $P_{i,j}$ is a P-tree for the j^{th} bit of the i^{th} band i . The complement of basic P-tree $P_{i,j}$ is denoted as $P_{i,j}^c$ (the complement operation is explained below). For each band (assuming 8-bit data values, though the model applies to data of any number bits), there are eight basic P-trees, one for each bit position. We will call these P-trees the basic P-trees of the spatial dataset. We will use the notation, $P_{b,i}$ to denote the basic P-tree for band, b and bit position, i . There are always $8n$ basic P-trees for a dataset with n bands. P-trees have the following features:

- P-trees contain 1-counts for every quadrant.
- The P-tree for any sub-quadrant at any level is simply the sub-tree rooted at that sub-quadrant.
- A P-tree leaf sequence (depth-first) is a partial run-length compressed version of the original bit-band.
- Basic P-trees can be combined to reproduce the original data (P-trees are lossless representations).
- P-trees can be partially combined to produce upper and lower bounds on all quadrant counts.

P-trees can be used to smooth data by bottom-up quadrant purification (bottom-up replacement of mixed counts with their closest pure counts).

P-trees can be generated quite quickly and can be viewed as a "data mining ready" and lossless format for storing spatial or any relational data.

2.2 P-tree variations

A variation of the P-tree data structure, the Peano Mask Tree (PM-tree, or PMT), is a similar structure in which masks rather than counts are used. In a PM-tree, we use a 3-value logic to represent pure-1, pure-0 and mixed quadrants (1 denotes pure-1, 0 denotes pure-0 and m denotes mixed). The PM-tree for the previous example is also given below. PMT requires less storage compared to PCT. PCT has the advantage of being able to provide the 1 bit count without traversing the tree. Since a PM-tree is just an alternative implementation for a Peano Count tree (PCT), we will use the term "P-tree" to cover both Peano Count tree (PCT) and Peano Mask tree (PMT).

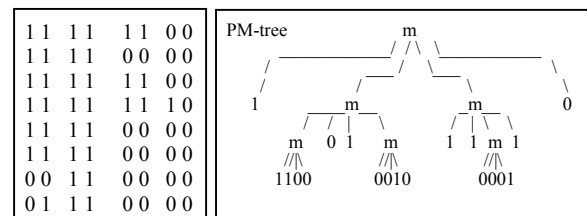


Figure 5. PM-tree

Other useful variations include P1-tree and P0-Tree. These are examples of a class of P-trees called **Predicate Trees**. Given a any quadrant predicate (a condition that is either true or false with respect to each quadrant), we use 1 to indicate true and 0 to indicate false for each quadrant at each level. The P1-tree (predicate is *pure-1*) and P0-tree of the example are.

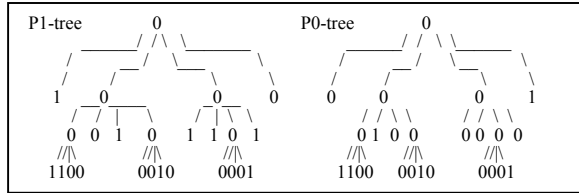


Figure 6 P1-tree and P0-tree

The predicate can be *not-pure-0* (NP0-tree), *not-pure-1-tree* (NP1-tree), etc.

A logical P-tree algebra including complement, AND and OR. The complement of a basic P-tree can be constructed directly from the P-tree by simply complementing the counts at each level (subtracting from the pure-1 count at that level), as shown in the example below. Note that the complement of a P-tree provides the 0-bit counts for each quadrant. P-tree AND/OR operations are also illustrated also.

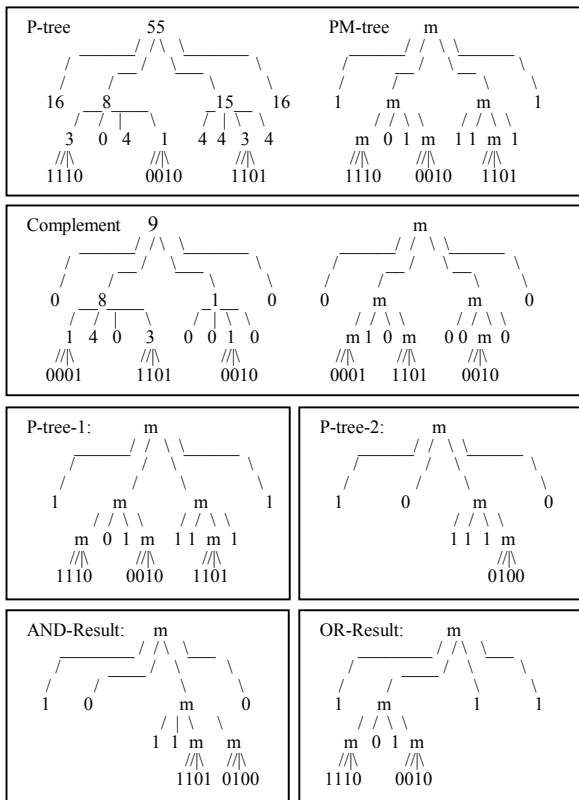


Figure 7. P-tree Algebra (Complement, AND, OR)

AND is the most important operation. The OR operation can be implemented in the very similar way. Below we will discuss various options to implement P-tree ANDing.

2.3 Level-wise P-tree ANDing

ANDing is a very important and frequently used operation for P-trees. There are several ways to perform P-tree ANDing. First let's look at a simple way. We can perform ANDing level-by-level starting from the root level. Table 1 gives the rules for performing P-tree ANDing. Operand 1 and Operand 2 are two P-trees (or sub-trees) with root X_1 and X_2 respectively. Using PM-trees, X_1 and X_2 could be any value among 1, 0 and m (3-value logic representing pure-1, pure-0 and mixed quadrant). Rules for P-tree ANDing are given in Table 1. For example, to AND a pure-1 P-tree with any P-tree will result in the second operand; to AND a pure-0 P-tree with any P-tree will result in the pure-0 P-tree. It is possible to ANDing two m's results in a pure-0 quadrant if their four sub-quadrants result in pure-0 quadrants.

Operand 1	Operand 2	Result
1	X_2	Sub-tree with root X_2
0	X_2	0
X_1	1	Sub-tree with root X_1
X_1	0	0
m	m	0 if four sub-quadrants result in 0; Otherwise m

Table 1 P-tree AND rules

2.4 P-tree AND using Pure-1 paths

In the following algorithm, we will assume P-trees are coded in a compact, depth-first ordering of the paths to each pure-1 quadrant. We use a hierarchical quadrant id (Qid) scheme below to identify quadrants. At each level, we append a sub-quadrant id number (0 means upper left, 1 upper right, 2 lower left, 3 lower right).

0	100	101	11
	102	103	
		12	13
2	3		

Figure 8 Quadrant id (Qid)

For a spatial data set with 2^n -row and 2^n -column, there is a mapping from raster coordinates (x, y) to Peano coordinates (called quadrant ids or Qids). If x and y are expressed as n -bit strings, $x_1x_2\dots x_n$ and $y_1y_2\dots y_n$, then the mapping is $(x, y)=(x_1x_2\dots x_n, y_1y_2\dots y_n) \rightarrow (x_1y_1 \cdot x_2y_2 \dots \cdot x_ny_n)$. Thus, in an 8 by 8 image, the pixel at $(3,6) = (011,110)$ has quadrant id $01.11.10 = 1.3.2$. For simplicity, we wrote the Qid as 132 instead of 1.3.2.

An example is given in below. Each path is represented by the sequence of quadrants in Peano order, beginning just below the root. Since a quadrant will be pure-1 in the result only if it is pure-1 in both/all operands, the AND is done as follows: scan the operands; output matching pure-1 paths.

The AND operation is effectively the pixel-wise AND of bits from bSQ files or their complement files. However, since such files can contain hundreds of millions of bits, shortcut methods are needed. Implementations of these methods have been done which allow the performance of an n -way AND of Tiff-image P-trees (1320 by 1320 pixels) in a few milliseconds. We discuss such methods later in the paper. The process of converting data to P-trees is also time consuming unless special methods are used. For example, our methods can convert even a large TM satellite image (approximately 60 million pixels) to its basic P-trees in just a few seconds using a high performance PC computer. This is a one-time process.

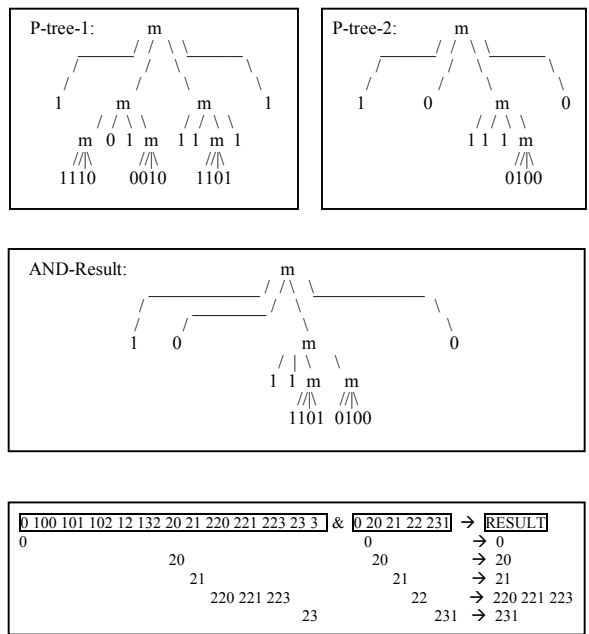


Figure 9 P-tree AND using pure-1 path

2.5 Value and Tuple P-trees

By performing the AND operation on the appropriate subset of the basic P-trees and their complements, we can construct P-trees for values with more than one bit.

Definition: A **value P-tree** $P_i(v)$, is the predicate P-tree for value equality with v at band i (v can be in 1-bit to 8-bit precision).

Value P-trees can be constructed by ANDing basic P-trees or their complements. For example, value P-tree $P_i(110)$ gives the count of pixels with band- i bit 1 equal to 1, bit 2 equal to 1 and bit 3 equal to 0, i.e., with band- i value in the range of $[192, 224)$. It can be constructed from the basic P-trees as:

$$P_i(110) = P_{i,1} \text{ AND } P_{i,2} \text{ AND } P_{i,3}'$$

P-trees can also represent data for any value combination from any band, even the entire tuple. In the very same way, we can construct **tuple P-trees**.

Definition: A **tuple P-tree** $P(v_1, v_2, \dots, v_n)$, is the predicate P-tree for equality with (v_1, v_2, \dots, v_n) for $i=1..n$. We have,

$$P(v_1, v_2, \dots, v_n) = P_1(v_1) \text{ AND } P_2(v_2) \text{ AND } \dots \text{ AND } P_n(v_n)$$

If value v_j is not given, it means it could be any value in Band j . For example, $P(110, ,101,001, , , ,)$ stands for a tuple P-tree of value 110 in band 1, 101 in band 3 and 001 in band 4 and any value in any other band.

Definition: An **interval P-tree** $P_i(v_1, v_2)$, is the predicate P-tree for band- i membership in the interval of $[v_1, v_2]$. We have,

$$P_i(v_1, v_2) = \text{OR } P_i(v), \text{ for all } v \text{ in } [v_1, v_2].$$

Definition: A **box P-tree** $P(l_1, h_1, \dots, l_n, h_n)$, is the predicate P-tree for membership in the box, $[l_1, h_1] \times \dots \times [l_n, h_n]$. We have,

$$P(l_1, h_1, \dots, l_n, h_n) = \text{AND } P_i[l_i, h_i], \text{ for } i=1..n.$$

Any predicate P-tree can be constructed by performing one multi-way AND of the appropriate basic P-trees and their complements (and possible an OR operation).

3 PROPERTIES OF P-TREES

In this section, we will discuss the good properties of P-trees. We will use the following notations:

$p_{x,y}$ is the pixel with coordinate (x, y) , $V_{x,y,i}$ is the value for the band i of the pixel $p_{x,y}$, $b_{x,y,i,j}$ is the j^{th} bit of $V_{x,y,i}$ (bits are numbered from left to right, $b_{x,y,i,0}$ is the leftmost bit). Indices: x : column (x -coordinate), y : row (y -coordinate), i : band, j : bit.

For any P-trees P, P_1 and P_2 , $P_1 \& P_2$ denotes P_1 AND P_2 , $P_1 | P_2$ denotes P_1 OR P_2 , $P_1 \oplus P_2$ denotes P_1 XOR P_2 , P' denotes COMPLEMENT of P .

$P_{i,j}$ is the basic P-tree for bit j of band i , $P_i(v)$ is the value P-tree for the value v of band i , $P_i(v_1, v_2)$ is the interval P-tree for the interval $[v_1, v_2]$ of band i , $rc(P)$ is the root count of P-tree P . P^0 is pure-0 tree, P^1 is pure-1 tree. N is the number of pixels in the image or space under consideration.

Lemma 1: For any two P-trees P_1 and P_2 , $rc(P_1 | P_2) = 0 \Rightarrow rc(P_1) = 0$ and $rc(P_2) = 0$. More strictly, $rc(P_1 | P_2) = 0$, if and only if $rc(P_1) = 0$ and $rc(P_2) = 0$.

Proof: (Proof by contradiction) Let, $rc(P_1) \neq 0$. Then, for some pixels there are 1s in P_1 and for those pixels there must be 1s in $P_1 | P_2$ i.e. $rc(P_1 | P_2) \neq 0$. But we assumed $rc(P_1 | P_2) = 0$. Therefore $rc(P_1) = 0$. Similarly we can prove that $rc(P_2) = 0$.

The proof for the inverse, $rc(P_1) = 0$ and $rc(P_2) = 0 \Rightarrow rc(P_1 | P_2) = 0$ is trivial. This immediately follows the definitions.

Lemma 2:

- a) $rc(P_1) = 0$ or $rc(P_2) = 0 \Rightarrow rc(P_1 \& P_2) = 0$
- b) $rc(P_1) = 0$ and $rc(P_2) = 0 \Rightarrow rc(P_1 \& P_2) = 0$.
- c) $rc(P^0) = 0$
- d) $rc(P^1) = N$
- e) $P \& P^0 = P^0$
- f) $P \& P^1 = P$
- g) $P | P^0 = P$
- h) $P | P^1 = P^1$
- i) $P \& P' = P^0$
- j) $P | P' = P^1$

Proofs are immediate.

Lemma 3: $v_1 \neq v_2 \Rightarrow rc\{P_i(v_1) \& P_i(v_2)\} = 0$, for any band i .

Proof: $P_i(v)$ represents all the pixels having value v for the band i . If $v_1 \neq v_2$, no pixel can have the values of both v_1 and v_2 for the same band. Therefore, if there is a 1

in $P_i(v_1)$ for any pixel, there must be 0 in $P_i(v_2)$ for that pixel and vice versa. Hence $rc\{P_i(v_1) \& P_i(v_2)\} = 0$.

Lemma 4: $rc(P_1 | P_2) = rc(P_1) + rc(P_2) - rc(P_1 \& P_2)$.

Proof: Let the number of pixels for which there are 1s in P_1 and 0s in P_2 is n_1 , the number of pixels for which there are 0s in P_1 and 1s in P_2 is n_2 and the number of pixels for which there are 1s in both P_1 and P_2 is n_3 .

Now, $rc(P_1) = n_1 + n_3$, $rc(P_2) = n_2 + n_3$, $rc(P_1 \& P_2) = n_3$

and $rc(P_1 | P_2) = n_1 + n_2 + n_3 = (n_1 + n_3) + (n_2 + n_3) - n_3$

$= rc(P_1) + rc(P_2) - rc(P_1 \& P_2)$

Theorem: $rc\{P_i(v_1) | P_i(v_2)\} = rc\{P_i(v_1)\} + rc\{P_i(v_2)\}$, where $v_1 \neq v_2$.

Proof: $rc\{P_i(v_1) | P_i(v_2)\} = rc\{P_i(v_1)\} + rc\{P_i(v_2)\} - rc\{P_i(v_1) \& P_i(v_2)\}$ (Lemma 4)

If $v_1 \neq v_2$, $rc\{P_i(v_1) \& P_i(v_2)\} = 0$. (Lemma 3)

Therefore, $rc\{P_i(v_1) | P_i(v_2)\} = rc\{P_i(v_1)\} + rc\{P_i(v_2)\}$.

4 DATA MINING TECHNIQUES USING P-TREES

The P-tree technology has been extended to work with a large number of data mining techniques. These include the following.

4.1 P-tree-based DTI Classifiers

This technique was used on large quantities of spatial data collected in various application areas, including remote sensing, geographical information systems (GIS), astronomy, computer cartography, environmental assessment and planning, etc. These data collections effectively arrive as streams of data since new data is constantly being collected. The problem with previous classifiers was that this presented a serious problem. Using P-tree technology, fast calculation of measurements, such as information gain, was achieved. The P-tree based decision tree induction classification and a classical decision tree induction method was experimental shown to be significantly faster than existing classification methods, making well suited for mining on streams and multimedia. [28]

4.2 Bayesian Classifiers

A Bayesian classifier is a statistical classifier, which uses Bayes' theorem to predict class membership as a

conditional probability that a given data sample falls into a particular class. The complexity of computing the conditional probability values can become prohibitive for most of the multimedia applications with a large attribute space. Bayesian Belief Networks relax many constraints and uses the information about the domain to build a conditional probability table. Naïve Bayesian Classification is a lazy classifier. Computational cost is reduced with the use of the Naïve assumption of class conditional independence, to calculate the conditional probabilities when required. Bayesian Belief Networks require build time and domain knowledge where as the Naïve approach loses accuracy if the assumption is not valid. The P-tree data structure allows us to compute the Bayesian probability values efficiently, without the Naïve assumption by building P-trees for the training data. Calculation of probability values require a set of P-tree AND operations that will yield the respective counts for a given pattern. Bayesian classification with P-trees has been used successfully on remotely sensed image data to predict yield in precision agriculture [30].

4.3 ARM

Association Rule Mining, originally proposed for market basket data, has potential applications in many areas. Extracting interesting patterns and rules from datasets composed of images and associated data can be of importance. However, in most cases the data sizes are too large to be mined in a reasonable amount of time using existing algorithms. Experimental results showed that using P-tree techniques in an efficient association rule mining algorithm P-ARM has significant improvement compared with FP-growth and Apriori algorithms. [28]

4.4 KNN and Closed KNN Classifiers

KNN classifiers typically have a very high cost associated with building a new classifier each time new data arrives. In this situation, k-nearest neighbor (KNN) classification is a very good choice, since no residual classifier needs to be built ahead of time. KNN is extremely simple to implement and lends itself to a wide variety of variations. The construction of the neighborhood is the high cost operation. By using P-tree technology and finding a closed-KNN set which does not have to be reconstructed. Experimental results show closed-KNN yields higher classification accuracy as well as significantly higher speed. [31]

4.5 P-tree Data Mining Performance

Based on the experimental work discussed above incorporation of P-tree technology into data mining applications has consistently improved performance. The data mining ready structure has demonstrated its potential for improving performance in multimedia data.

Many types of data show continuity in dimensions that are not themselves used as data mining attributes. Spatial data that is mined independently of location will consist of large areas of similar attribute values. Data streams and many types of multimedia data, such as videos show a similar continuity in their temporal dimension. The P-tree data structure uses these continuities to compress data efficiently while allowing it to be used in computations. Individual bits of the mining-relevant attributes are represented in separate P-trees. Counts of attribute values or attribute ranges can efficiently be calculated by an "AND" operation that all relevant P-trees. These "AND"-operations can be efficiently implemented based on the regular structure that compresses entire quadrants, while making use of pre-computed counts that are kept at intermediate levels of the tree structure.

5 IMPLEMENTATION ISSUES AND PERFORMANCE

The performance of the P-tree data structure is discussed with respect to P-tree storage and the execution time for AND operations. The amount of internal memory required for each P-tree structure is related to the respective size of the P-tree file stored in secondary storage. The creation and storing of P-trees is a one-time process. To make a generalized P-tree structure, the following file structure is proposed (table 2) for storing basic P-trees. .

1 byte	2 bytes	1 byte	4 bytes	2 bytes	
Format Code	Fan-out	# of levels	Root count	Length of the body	Body of the P-tree

Table 2 P-tree file structure

Format code: Format code identifies the format of the P-tree, whether it is a *PCT* or *PMT* or in any other format.

Fan-out: This field contains the fan-out information of the P-tree. Fan-out information is required to traverse the P-tree in performing various P-tree operations. The fan-out is decided at creation time. In the case of using different fan-outs at different levels, it will be used as an identifier.

of levels: Number of levels in the P-tree. This will indicate the number of levels in the P-tree for the given fan-out.

Root count: *Root count* i.e. the number of 1s in the P-tree. Though we can calculate the *root count* of a P-tree on the fly from the P-tree data, these 4 bytes of space can save computation time when we only need the root count of a P-tree to take advantage of the properties described in section 2.5. The root count of a P-tree can be computed at the time of construction with very little extra cost.

Length of the body: Length of the body is the size of the P-tree file in bytes excluding the header. The size of the

P-tree varies due to the level of compression in the data. To allocate memory dynamically for the P-trees, it is better to know the size of the required memory size before reading the data from disk. This will also be an indicator of the distribution of the data, which can be used to estimate the required AND time in advance for the given search space.

Body of the P-tree : This will contain a long stream of bytes representing the P-tree in the respective format.

We only store the basic P-trees for each dataset. All other P-trees (value P-trees and tuple P-trees) are created on the fly when required. This results in a considerable saving of space. Figure 10, 11 and 11 gives the storage requirements for various formats of data (TIFF, SPOT and TM scene) using various formats of P-trees (PCT or PMT) with different fan-out patterns. Fan-out pattern f1-f2-f3 will indicate a fan-out of f1 for the root level, f3 for the leaf level and f2 for all the other levels. The variation in the size is due to the different levels of compression for each bit in the image. It is important to note that P-tree is a lossless representation of the original data. Different representations have an effect on the computation of the Ptree operators. The performance of the processor against memory access should be taken into consideration when selecting a representation.

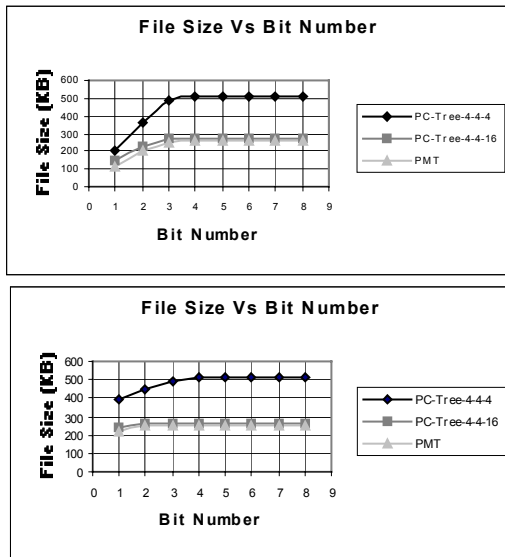


Figure 10 Comparison of file size for different bits of Band 1 & 2 of a TIFF image

The efficiency of data mining with the P-tree data structure relies on the time required for basic P-tree operators. The AND operation on 8 basic P-trees can be done in 12 milliseconds for an image file with 2 million pixels. Experimental results also show that the AND operation is scalable with respect to data size and the number of attribute bits. Figure 13 and 14 show the time required to perform the P-tree AND operation.

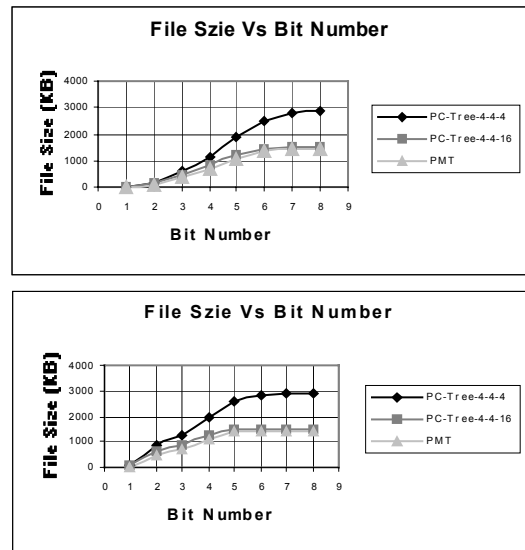


Figure 11 Comparison of file size for different bits of Band 3 & 4 of a SPOT image

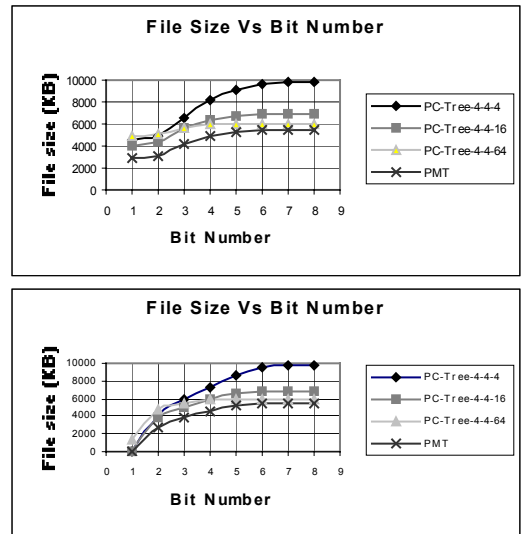


Figure 12 Comparison of file size for different bits of Band 5 & 6 of a TM image

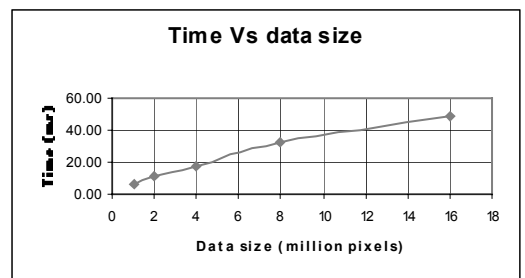


Figure 13 Comparison of time required to perform AND operation with different data sizes

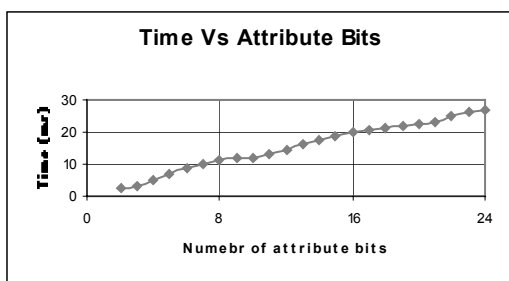


Figure 14 Time to perform AND operation for different number of attribute bits

The P-tree data structure provides an opportunity to use high performance parallel and distributed computing, independent of the data mining technique. The most common approach is to use a quadrant based partition, i.e. a horizontal partition. In this approach the AND operations on each partition can be accumulated to produce the global count. A vertical partition can also be used with a slight increase in communication cost. In this approach the AND operation on partially created value P-trees from each partition will produce the global count. Both these approaches can be used to mine distributed multi media data by converting the data into P-trees and storing it at the data source. The particular data mining algorithm will be able to pull the required counts through a high speed dedicated network or the Internet. If the latency delay is considerably high this approach may put a restriction on the type of algorithms to suit batched count requests from the P-trees.

6 RELATED WORK

Concepts related to the P-tree data structure, include Quadtrees [1, 2, 3, 4, 5] and its variants (such as point quadtrees [3] and region quadtrees [4]), and HH-codes [6].

Quadtrees decompose the universe by means of iso-oriented hyperplanes. These partitions do not have to be of equal size, although that is often the case. The decomposition into subspaces is usually continued until the number of objects in each partition is below a given threshold. Quadtrees have many variants, such as point quadtrees and region quadtrees.

HH-codes, or Helical Hyperspatial Codes, are binary representations of the Riemannian diagonal. The binary division of the diagonal forms the node point from which eight sub-cubes are formed. Each sub-cube has its own diagonal, generating new sub-cubes. These cubes are formed by interlacing one-dimensional values encoded as HH bit codes. When sorted, they cluster in groups along the diagonal. The clusters are order in a helical pattern, thus the name "Helical Hyperspatial".

The similarities among P-tree, quadtree and HHCode are that they are quadrant based. The difference is that P-trees focus on the count. P-trees are not index, rather they

are representations of the datasets themselves. P-trees are particularly useful for data mining because they contain the aggregate information needed for data mining.

7 CONCLUSION

This paper reviewed some of the issues of multimedia data mining and concludes that one of the major issues of multimedia data mining is the sheer size of the resulting feature space extracted from the raw data. Deciding how to efficiently store and process this high volume, high dimensional data will play a major role in the success of a multimedia data mining project. This paper proposes the use of a data mining ready data structure to solve the problem. To that end the Peano Count Tree (or P-tree), and its algebra and properties were presented. The P-tree structure can be viewed as a data-mining-ready structure that facilitates efficient data mining [7]. Previous work has demonstrated that using the P-tree algebra can perform standard data mining techniques efficiently while operating directly from a compress data storage.

8 REFERENCES

- [1] Volker Gaede and Oliver Gunther, "Multidimensional Access Methods", *Computing Surveys*, 30(2), 1998.
- [2] H. Samet, "The quadtree and related hierarchical data structure". *ACM Computing Survey*, 16, 2, 1984.
- [3] H. Samet, "Applications of Spatial Data Structures", Addison-Wesley, Reading, Mass., 1990.
- [4] H. Samet, "The Design and Analysis of Spatial Data Structures", Addison-Wesley, Reading, Mass., 1990.
- [5] R. A. Finkel and J. L. Bentley, "Quad trees: A data structure for retrieval of composite keys", *Acta Informatica*, 4, 1, 1974.
- [6] HH-codes. Available at <http://www.statkart.no/nlhdb/iveher/hhtext.html>
- [7] William Perrizo, Qin Ding, Qiang Ding and Amalendu Roy, "Deriving High Confidence Rules from Spatial Data using Peano Count Trees", Springer-Verlag, LNCS 2118, July 2001
- [8] Jochen Doerre, Peter Gerstl, Roland Seiffert "Text Mining: Finding Nuggets in Mountains of Textural Data"
- [9] Dan Sullivan "The Need for Text Mining in Business Intelligence"
- [10] Osmar R.Zaiane, Jiawei Han, Ze-Nian Li, Sonny H.Chee, Jenny Y.Chiang, "MultiMediaMiner: A System Prototype for MultiMedia Data mining", In pro.1998 ACM-SIGMOD Conf.on Management of Data, June 1998
- [11] Wei-Hao Lin, Rong Jin, Alexander Hauptmann, "Meta-classification of Multimedia Classifiers", First International Workshop on Knowledge Discovery in Multimedia and Complex Data

- [12] P. Indyk, R. Motwani, P. Raghavan “locality-preserving hashing in multidimensional spaces”,
- [13] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The KDD process for extracting useful knowledge from volumes of data. *Communication of ACM*, 39(11):27-34, November 1996.
- [14] Wei-hao lin, Rong Jin, Alexander Hauptmann, Meta-classification of Multimedia classifiers, First international workshop on knowledge discovery in multimedia and complex data, Taipei, Taiwan, May 6, 2002
- [15] William Baker, Arthur Evans, Lisa Jordan, Saurabh Pethe, “User Verification System” The Mid-Atlantic Student Workshop on Programming Languages and Systems Pace University, April 19, 2002
- [16] C. Aggarwal, “Re-designing Distance Functions and Distance-Based Applications for High Dimensional Data”, SIGMOD 2001.
- [17] M. Gavrilov, D. Anguelov, P. Indyk, R. Motwani, “Mining The Stock Market: Which Measure Is Best?”, KDD 2000
- [18] J. Caraca-Valente, I. Lopez-Chavarrias, “Discovering Similar Patterns in Time Series”, KDD 2000
- [19] J. Yoon, T. Kim, and H. Lee, “The Information of Trading Volume in the Prediction of Stock Index returns: A Nonparametric Investigation”, INFORMS & KORMS, 2000.
- [20] A. Hinneburg, C. Aggarwal, and D. Keim, “What Is the Nearest Neighbor in High Dimensional Spaces?”, Proc. of the 26th VLDB Conference 2000.
- [21] C. Aggarwal, A. Hinneburg, and D. Keim, “On the Surprising Behavior of Distance Metrics in High Dimensional Space”, ICDT 2001.
- [22] Chabane Djeraba, “Image Access and Data Mining: An Approach”, PKDD 2000.
- [23] Chabane Djeraba, Henri Briand, “Temporal and Interactive Relations in a Multimedia Database System”, ECMAST 1997.
- [24] Osmar R. Zaiane, Simeon J. Simoff, “Multimedia Data Mining for the Second Time”, SIGKDD Explorations, Vol 3, N 2, January 2002.
- [25] Osmar R. Zaiane, Jiawei Han, Hua Zhu, “Mining Recurrent Items in Multimedia with Progressive Resolution Refinement”, ICDE 2000.
- [26] Simeon J. Simoff, Osmar R. Zaiane, “Multimedia data mining”, KDD 2000.
- [27] Osmar R. Zaiane, Jiawei Han, Ze-Nian Li, Jean Hou, “Mining Multimedia Data”, *CASCON'98: Meeting of Minds*, 1998.
- [28] “Decision Tree Classification of Spatial Data Streams Using Peano Count Trees”, Qiang Ding, Qin Ding and William Perrizo, Proceedings of ACM Symposium on Applied Computing (SAC'02), Madrid, Spain, March 2002, pp. 413-417.
- [29] "Association Rule Mining on Remotely Sensed Images Using P-trees", Qin Ding, Qiang Ding and William Perrizo, Proceedings of PAKDD 2002, Springer-Verlag, LNAI 2336, May 2002, pp. 66-79.
- [30] Mohamed Hossain, ‘Bayesian Classification using P-Tree’, Master of Science Thesis, North Dakota State University, December 2001.
- [31] "K-nearest Neighbor Classification on Spatial Data Stream Using P-trees", Maleq Khan, Qin Ding and William Perrizo, Proceedings of PAKDD 2002, Springer-Verlag, LNAI 2336, May 2002, pp. 517-528.
- [32] "Biological Systems and Data Mining for Phylogenomic Expression Profiling " Willy Valdivia-Granda*, Edward Deckard, William Perrizo, Qin Ding, Maleq Khan, Qiang Ding, Anne Denton

Scale Space Exploration for Mining Image Information Content

Mariana Ciucu, Patrick Heas, Mihai Datcu
IMF Remote Sensing Technology Institute
DLR German Aerospace Center,
D-82230 Wessling, Germany
mihai.datcu@dlr.de

James C. Tilton
NASA's Goddard Space Flight Center
Applied Information Sciences Branch
Greenbelt, MD 20771, USA
James.C.Tilton.1@gsc.nasa.gov

ABSTRACT

Images are highly complex multidimensional signals, with rich and complicated information content. For this reason they are difficult to analyze through a unique automated approach. However, a hierarchical representation is helpful for the understanding of image content.

In this paper, we describe an application of a scale-space clustering algorithm (melting) for exploration of image information content. Clustering by melting considers the feature space as a thermodynamical ensemble and groups the data by minimizing the free energy, having the temperature as a scale parameter. We develop clustering by melting for multidimensional data, and propose and demonstrate a solution for the initialization of the algorithm.

Due to computational reasons due to the curse of dimensionality, for initialization of clusters we choose the initial clusters centers with another algorithm, which performs a fast cluster estimation with low computation cost. We further analyze the information extracted by melting and propose an information representation structure that enables exploration of image content. This structure is a tree in the scale space showing how the clusters merge.

Implementation of the algorithm is through a multi-tree structure. With this structure, we can explore the image content as an information mining function, we obtain a more compact data structure, we have maximum of information in scale space because we memorize the bifurcation points and the trajectories of the centers points in the scale space.

The information encoded in the tree structure enables the fast reconstruction and exploration of the data cluster structure and the investigation of hierarchical sequences of image classifications.

We demonstrated examples using satellite multispectral image (SPOT 4) and Synthetic Aperture Radar – SAR and Digital Elevation Models – DEM derived from SAR interferometry (SRTM).

Keywords

Data mining, melting algorithm, fast cluster estimation

1. INTRODUCTION

Data mining and knowledge discovery are the processes of analyzing data from different perspectives and summarizing it into useful information. Technically, data mining is the process of finding correlations or patterns of fields in large relational databases [3].

1.1. Clustering

Clustering is one of the most important tasks performed in Data Mining applications. Clustering of data is a method by which large sets of data are grouped into clusters having similar behaviour. Thus, we see clustering means grouping of data or dividing a large data set into smaller data sets of some similarity.

A clustering algorithm attempts to find natural groups of components (or data) based on some similarity. The clustering algorithm also finds *the centroid e.g. center of mass or center of gravity*) of a group of data sets. To determine cluster membership, most algorithms evaluate a distance between a point and the cluster *centroids*. The output from a clustering algorithm is a statistical description of the clusters, *centroids* and the number of components in each cluster.

There is more than one way to measure a distance. There are distances that are Euclidean if the attributes are continuous, and there are other distances based on similarity. Generally, the distance between two points is taken as a common metric to assess the similarity among the components of a population.

The Euclidian distance measure between two points $p = (p_1, p_2, \dots)$ and $q = (q_1, q_2, \dots)$ is:

$$d = \sqrt{\sum_{i=1}^k (p_i - q_i)^2} \quad (1)$$

The various clustering concepts available can be grouped into two classifications, that are by the type of structure imposed on the data [1]:

1. Hierarchical clustering
2. Nonhierarchical clustering.

1. Hierarchical clustering

A hierarchical clustering is a sequence of partitions in which each partition is needed to form the subsequent partition in the sequence. These methods include those techniques where the input data are not partitioned into the desired number of classes in a single step. Instead, a series of successive fusions of data are performed until the final number of clusters is obtained. An important objective of hierarchical clustering is to provide a picture of the data that can be easily interpreted, such as a dendrogram. An example of hierarchical clustering is the melting algorithm.

2. Nonhierarchical clustering (partitional clustering)

These methods include those techniques in which a desired number of clusters is assumed at the start, and a single partition is found. Points are allocated among clusters so that a particular clustering criterion is optimized. A possible criterion is the minimization of the variability within clusters, as measured by the sum of the variance of each parameter that characterizes a point. Examples of nonhierarchical clustering are K-means, and Expectation-Maximization (EM)

K-means has as an input a predefined number of clusters, and is a simple, iterative procedure. This algorithm assigns each data point to the cluster center closest to it, forming in this way k exclusive clusters of the data.

Expectation Maximization (EM) algorithm is a mixture based algorithm that assumes the data set can be modelled as a linear combination of multivariate normal distributions. The algorithm finds the distribution parameters that maximize a model quality measure, called likelihood, producing the maximum likelihood (ML) solution.

2. CLUSTERING BY MELTING AND OUR IMPLEMENTATION

Melting algorithm is a clustering algorithm based on information theory and statistical mechanics and is the only algorithm that incorporates scale and cluster independence. Using information theory and statistical mechanics, Wong [7] showed that cluster centers correspond to the local minima of a thermodynamical free energy F that depends on the data points and the scale parameter β . The algorithm is scale-space based and provides more effective clustering than other methods.

The basic idea is that clusters depend on the scale one uses to examine the data.

At a very coarse scale, the whole dataset is a cluster; while at a very fine scale, every datum is itself a cluster. In scale space, one should see all the clusters and the meaningful clusters tend to stay unchanged over a long range of scales.

It is easy to see from the relevant equation that the number of minima depends on the distribution of the data points and the scale parameter β , which is the "inverse temperature." If we start with a large β (low temperature) so that every data point is a cluster, then as we gradually decrease β (increase the temperature), the clusters merge; and finally, at a very small β (very high temperature), all data points merge to one cluster.

If clusters of several points indeed exist, the information should be present in the data itself. Data points closer to the cluster center should give more information about the clusters while those far away should give less. These different degrees of contribution can be modeled probabilistically by defining $p(x|y)$ as a contribution of data point x to a cluster center y .

The problem is to find the set of cluster center y that best suit the data points x with respect to some constraints. The best solution is obtained by maximizing the entropy:

$$H = \sum_{x \in D} p(x|y) \log p(x|y) ,$$

where D is data space.

Suppose the cost function is $e(x) = (x - y)^2$, where x is a data point and y is a cluster center. This is the squared distance. Maximizing the entropy with the constraint:

$$\sum_{x \in D} p(x|y) e(x) = C$$

$$\text{we obtain } p(x|y) = \frac{\exp[-\beta(x - y)^2]}{Z}$$

$$\text{where } Z = \sum_{x \in D} \exp[-\beta(x - y)^2]$$

To make the connection with thermodynamics, the free energy is $F = -\frac{1}{\beta} \log Z$. At equilibrium, a thermodynamic system settles into equilibrium if it has minimum free energy.

Minimum free energy is obtained if $\frac{\partial F}{\partial y} = 0$, or equivalently

$$y = \frac{\sum_{x \in D} (x - y) * \exp[(-\beta) * (x - y)^2]}{\sum_x \exp[(-\beta) * (x - y)^2]} \quad (2)$$

This equation is very different from that obtained by the maximum likelihood of a Gaussian mixture.

For a given β , the problem of clustering is mapped to the problem of finding solution for y of Eq. (3). However, for a general β , the solution cannot be found analytically. The solutions are identical to the fixed points of the following map:

$$y \xrightarrow{f} y + \sum_{x \in D} \frac{(x - y) * \exp[(-\beta) * (x - y)^2]}{\sum_x \exp[(-\beta) * (x - y)^2]} \quad (3)$$

The solutions can be computed by an iterative equation (11) [2].

Thus, the structure of the melting algorithm is:

1. An initial high β is chosen and every data point is set as a cluster.
2. β is decreased a little bit
3. the mapping (3) is repeated N times or until the cluster converges
4. If two or more clusters, which previously were distinct, share the same center, the set of data associated with the new cluster is the union of those with the original clusters.
5. If more than one clusters exist, go to 2. Otherwise, stop.

The information obtained by melting algorithm is:

- The set of clusters as functions of temperature
- Trajectories of cluster centers as functions of temperature
- Bifurcation points
- Free energy schedule dependency of temperature
- The sequences of hierarchical image classification

This information can be used to explore the image content as an information mining function.

However, due the computational complexity, an optimal data representation is needed for:

- more compact data structure
- fast and easy access to the information

We propose a tree structure, that has a two node structure:

Node1

- pointer to the same node structure (to *Node1*)
- pointer to the following node structure (to *Node2*)

Node2

- vector for features (in our case we have four features for four bands)
- scalare for beta

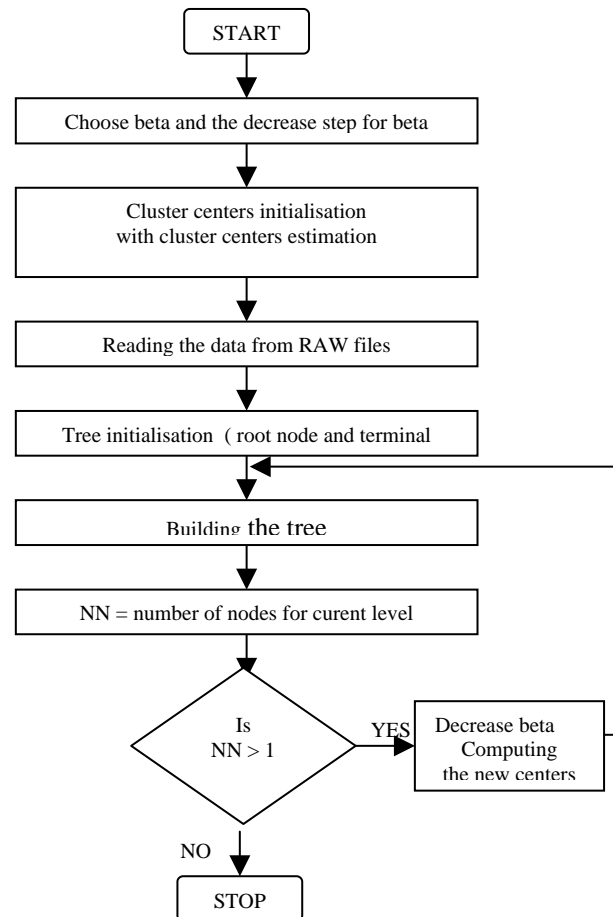
- scalare for index, which is for image map
- pointer to *Node1*

The index is necessary for this structure because if two clusters centers have the same value we put in the next level of the tree the same index. With this index, we can obtain the sequences of images classification, as we can see in Section 4, in figure 4, 11. With this structure we can make fast and easy the plot of clusters centers versus temperature, as we can see in figures 5-8, 12, 13. Thus, is only necessary to cross the tree from the terminal nodes to the root node, for each terminal node, with a recursive function. In our algorithm, which is implemented in C, each level of tree corresponds to each temperature, and for this consideration, we can reconstruct the information of image from one temperature to another.

The tree contains the maximum information about the image in scale space, because we don't record only the bifurcation points, but also the trace of all the center points in the scale space.

The tree structure is a multi tree, which has a multi - tree to the left and a multi - tree to the right. The tree is built from the terminal nodes to the root, because we wish that all the computations be done during the building of tree. The heap memory is only necessary for recording the tree structure.

The flowchart of this algorithm, which contains the melting algorithm and the tree structure, follows:



3.1. Computational problem and dimensionality aspects

The generalization of the algorithm for the multidimensional case raises two problems:

- *the computational complexity*

The computational complexity is :

$$O((n \times d \times n_i \times n_\beta)) + \log 2(n_i),$$

where

n is number of points

d is the dimensions for the features

n_i is number of iterations

n_β is number of temperature steps

$\log 2(n_i)$ is the tree complexity, where n_i is number of nodes from tree, $n_i = 2^{(n_\beta+1)} - 1$

The solution for this is to split the computation into two steps:

1. off-line – generating the tree information structure
2. on-line – analyzing and exploring of image content

- *the curse of dimensionality at algorithm initialization*

We can deal with this in many ways. For example:

1. choosing the initial clusters centers randomly. However, in this case we can lose much information about data;
2. choosing the initial cluster centers with another algorithm, such as the "Fast cluster centers estimation," which will be discussed in the next section.

The second way is better than first, because we don't lose information and with this we have a low computational cost, because we begin only with few data points as a cluster and not with all data points.

3.1.1. Fast cluster centers estimation

Numerical gradient estimation methods may be used in order to reduce the computational demands of a class of multidimensional clustering algorithms, or may be used in a direct way to make an initial exploration of large data sets by evaluating the number of existing clusters.

3.1.1.1. Description of the Merging Gradient Estimation algorithm

This algorithm is presented in Fox [5].

Assuming that clusters are regions of relatively high point density within the data space, which is to say that

the rate of change of points occurrence with respect to distance travelled in all directions of the space is relatively high – i.e. higher than the rate occurrence which would be encountered if all the points were uniformly distributed over all the space since this represents the maximum entropy case in which any cluster exists. Furthermore clusters centers may then be considered as local maxima of such gradients. However this local maxima of the gradient i.e. marginal density, has to exhibit a value greater than the marginal density that would occur if all the points were evenly distributed. As an example the upper right graph of figure 1 shows the density of points repartition in a two dimensional space and the marginal densities on the two axes of synthetic Gaussian data.

The computational procedure is as follows:

First, of the N dimensional Gaussian data X of n elements is read.

$$X^i = (x_1^i, x_2^i, \dots, x_n^i), i = 1, \dots, n \quad (4)$$

The next step is to sort the data for each of the N dimensions into ascending numerical order since travelling sequentially through sorted vectors corresponds to travelling along the different dimension axes.

$$S^m = (s_1^m, s_2^m, \dots, s_n^m), m = 1, \dots, n \quad (5)$$

$$S^m = \text{sort}(s_1^i, s_2^i, \dots, s_n^i), i = 1, \dots, n \quad (6)$$

Define the vector C representing the cumulative sum of points encountered as one move along any of the sorted vectors s_j .

$$C_i = i; i = 1, \dots, N - 1 \quad (7)$$

The marginal density estimates in each direction may be then interpreted as the gradient of the N graphs generated by plotting C versus s_j the figure 1 (upper left and lower right graphs). This exhibits the repartition of a Gaussian synthetic data for two dimensions of the feature space the marginal densities on two axes of this space and also the step functions C versus s_j . However, to compute the gradients presented as well in these graphs a numerical differentiation from discretely sampled data is required. A simple but fast technique is applied here. It begins by filtering the sorted vectors s_j in order to smooth out the raw data C versus s_j curves. Hence, we obtain:

$$f_j^m = \frac{1}{2h+1} \sum_{r=m-h}^{r=m+h} s_j^r \quad (8)$$

The smoothing window used here is a parameter that determinates the scale of Gaussian structures we will detect. The next step is the computation of the gradient estimates g_j . It may then be obtained from the smoothed C versus f_j curves according to the constructions

$$g_j^m = \frac{2h}{f_j^{m+h} - f_j^{m-h}} \quad (9)$$

$$g_{mean,j} = \frac{n-1}{f_j^n - f_j^1} \quad (10)$$

The second equation computes the average point density that would exist if the data was uniformly distributed in all the space. The edges may be computed for the filtering and for the gradient estimates by the use of descending spans. Then all local maxima of the gradient estimates, which are above the average marginal density value, have to be extracted. The final step is to select only the maxima that correspond to an existing data value in the n different dimensions. Of course, the correspondence to the original data has to be saved. These maxima correspond to the approximated centers of the clusters.

3.1.1.2. Application of an optimised algorithm

In order to reduce the computational time of a "classical" sorting procedure, a sorting routine of complexity $N*n$ (number of dimension by number of data points) has been developed. The idea is to scan the data only once and to sort, each data point for each dimension, in his associated dynamic collections itemized by his value. For an 8 bits, 4 dimensional data set, the number of collection will be then lower or equal to $4*256$. Then, for each dimension, the collections are concatenated by order of crescent value to constitute the N different sorted vectors.

A last change is applied here in order to avoid centers of similar value. This can happen when irregularity remain after smoothing the data. The extra centers are simply removed.

Finally, this algorithm has complexity $N*n$, what in time computation, constitute an advantage on for example the K-Means algorithm which has complexity $N*n*K$, where K is the number of cluster. Furthermore, the algorithm doesn't need to have a fixed number of clusters as an input.

Discussion of the results

Taking into account the main quality of the algorithm, which is the low computational cost, the results shows a good efficiency versus time consumed.

We tested this algorithm initially on 4 dimensional synthetic data composed of uniform distributed noise, and 3 Gaussian structures of different mean only in two dimensions in order to simplify the interpretation of the results. One of them has a larger variance and another has a lower density.

The algorithm performance in finding the correct number of Gaussian structures in a reasonable amount of time consumption depends on the smoothing parameter discussed previously. This parameter influences the regularity of the gradient function and consequently the number of maxima detected. Since we are smoothing, we are losing precision on the centers value. Moreover, if we use a large smoothing window to detect only the relevant Gaussian structures, the lost of precision on the centers value will make it impossible to find the correspondence of maxima between the different dimensions. On one hand, we will obtain, by a small smoothing window, a good detection of all the clusters

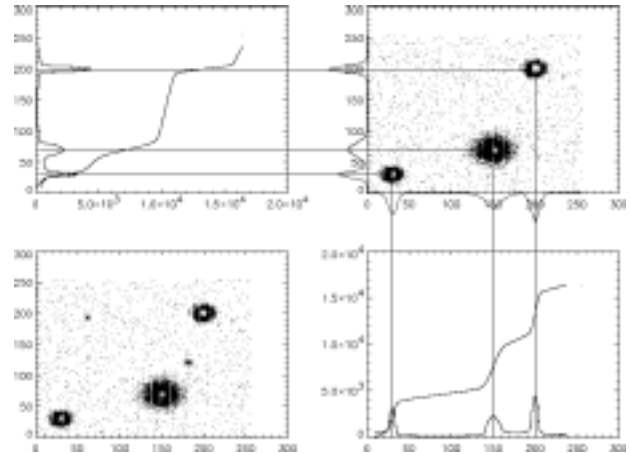


Figure 1: *Merging Gradient Algorithm on synthetic data set.*

but with many centers belonging to the same Gaussian (typically when the structures are not enough regular and with big densities). The upper right plot in figure1 illustrates this effect, showing the detection of three centers for the Gaussian of largest density. On the other hand, we will obtain, by a large smoothing window (which means a greater time consumption), single center detection for each Gaussian structure. However, some structures, as Gaussian of greater variance or lower density, may not be detected and we will loose precision on the center's value. Currently, this parameter is estimated heuristically. However, a correct estimation of this parameter could be performed.

The inability of finding a good estimate of the number of clusters when the structures are too different has little consequence when this algorithm is used only to initialise a more powerful, but slower, clustering algorithm such as "Melting" algorithm presented in the next chapter.

3.1.1.3. Enhanced algorithm for estimation of number of clusters

This fast center algorithm estimator may also be used to explore large data sets by estimating directly the number of Gaussian structures existing in the data and their center's value. We assume the data to be a mixture of Gaussians. The problem, to be solved, is to detect

Gaussian structures with different variances, densities, regularities, with only one maximum associated with each one of them.

3.1.1.3.1. Removing the centers which migrates

A way to face this problem is to observe the evolution of the centers value given by the merging gradient estimator algorithm, while we compute their new value.

To compute them, we first create classes associated to each center value. Each class regroups the smoothed data that present a minimum distance to each center value. The new center's values are calculated as the gravity center of each class.

Let's suppose we have detected all the structures with at least one center associated by an appropriate smoothing window. Since a "unclustered mass" remains (noise or other type of structures which have no center directly associated but only a distant center value), we will observe after the computing of the new center values a fast migration of center that share the same Gaussian structure and divide it into more than one class.

These "extra centers" will move to the barycenter of the "unclustered mass". The lower left plot of figure 1 illustrates these migrations. In this case, a complete K-means, initialised with the center estimated by the fast merging gradient algorithm presented in the upper right plot of figure 1, was applied to show clearly these migration phenomena. We can clearly see that two extra centers, belonging to the Gaussian with the greatest density, have moved to the barycenters of the "unclustered noise".

Therefore, the idea is to keep updating the centers, by removing those that migrate farther than a fixed limit, while we iterate the procedure describe above.

This procedure will end when any center will migrate farther than this limit. There will be finally remaining only single centers associated to each of the Gaussians previously detected.

The choice of the migration limit depends on the topology of the smoothed data. In the case the data is composed of a mixture of Gaussians with very different densities, this procedure might not be very efficient, because the attraction of the high density Gaussians will be too powerful and we could loose first all the centers associated to the small density Gaussians. This procedure will be more efficient for a mixture of Gaussians of similar densities. This case will be approached by using a large smoothing window, but small structures might not be detected any more. However, in all the cases the migration limit can be adjusted in a way to avoid losing significant centers but with the disadvantage of keeping insignificant centers. We choose here a heuristic migration limit. However, an estimation of this parameter, by for example a maximum likelihood estimator, can be computed to optimise this choice.

3.1.1.3.2. Injection of an attractor

For the case in which the data consists only of a mixture of Gaussians without any noise, the "extra centers" previously detected won't be attracted by any "unclustered mass", and any migration will be observed or only migration of centers associated to low densities in direction of the high density regions of the feature space. Furthermore, the attractor will more strongly exert its influence in its surrounding area than far from it.

To balance these problems, uniformed distributed noise can be injected in the feature space to favour as equally as possible the removal of the "extra centers". The quantity of noise-injected must be adjusted so that it attracts only the "extra centers" This noise mustn't drown or modify significantly any of the structures detected (i.e. its density must be much lower). The quantity of noise injected constitutes another parameter that can be estimated. Here the estimation was again only heuristic.

4. EXPERIMENTAL RESULTS

4.1. Merging gradient algorithm applied on a SPOT image

In this paragraph, we apply the precedent algorithm on a sample 256*256 of a 4 Bands Spot4 image from a region near Bucharest. The original image is presented in figure 2a. The repartition of the multispectral data in the feature space is illustrated in figure 3. The projections of the densities on the 4 channels are plotted in the upper figures.

Three different center estimations have been computed leading to 142, 18 and 4 cluster centers. The classification resulting of these clustering are presented in respectively figure 2b, c and d. For the classification b, c and d, the parameters of smoothing were chosen respectively equal to {430,650,750}, the migration limits were fixed to {70,39,57} and the quantity of noise injected was equal to {1e4, 1e4, 26e3}.

We observe a super-estimation of the number of clusters in the first case. The classification with 4 classes is a sub-estimation of the number of clusters. The classification with 18 classes is a good fast number of clusters estimation. The center locations are presented in lowest plots for 4 classes and in the upper plots for 18 classes of figure 3.

The time computation was for the example with 142 classes done 47 sec on a "300 MHz SUNW, UltraSPARC-II". As a comparison, the K-means algorithm was computed with the same conditions and last 2'35 sec.

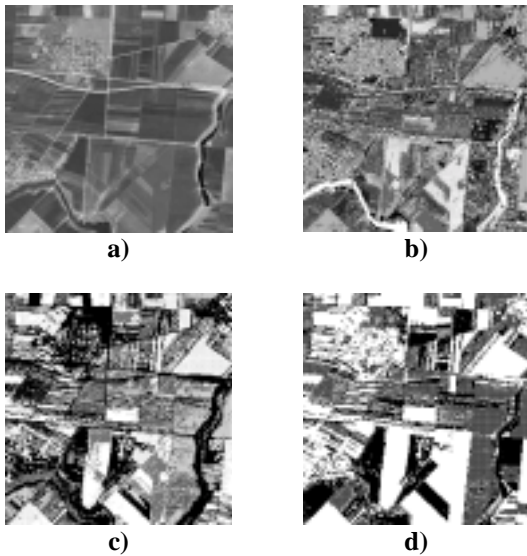


Figure 2: a) Original image (band 1, 2 and 3), classification with: b) 142 classes, c) 18 classes, d) 4 classes

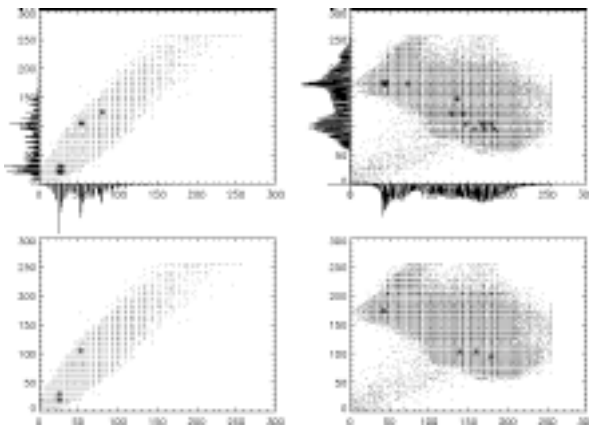


Figure 3: center location, for classification with 18 classes (up) and with 4 classes (down), in feature space: band1-2 (left), bands3-4 (right)

4.2. Melting applied on the same SPOT image and on a SAR image

The following is an application of the Melting algorithm with as initialization the above center estimation result on the same SPOT image. We also apply the melting procedure on a SAR image. The formulation is computationally intensive. For each image, the labeling of the various regions can be treated as a clustering problem.

For all images, SPOT and SAR, the pixel intensity is normalized so that a full intensity of 255 corresponds to 1.0 before doing the analysis.

With propose structure we obtain a sequences of hierarchical image, so we have more information of

classification than only with one image. We can see what clusters merge together, how many clusters we have at each temperature and we can choose what is the good number of clusters.

In the classical solution, when we need the initial number of clusters we can lose clusters, because we don't know the best number of clusters or we can have many clusters without points.

The sequences of hierarchical image classification in figure 4 are for bifurcation points in figures 5 - 6 and in figure 11 for figures 12,13.

Trajectories list the clustering one after another. Cutting a trajectory at any level defines a clustering and identifies clusters.

- Input*
1. Beta and step for beta
 2. Original image
 3. Center of clusters (initial configuration)
 4. Tree structure

- Output*
1. Sequences of images classification
 2. Graphics of bifurcation points

4.2.1. SPOT image

The four intensities form a feature vector for each pixel, (y_1, y_2, y_3, y_4) .

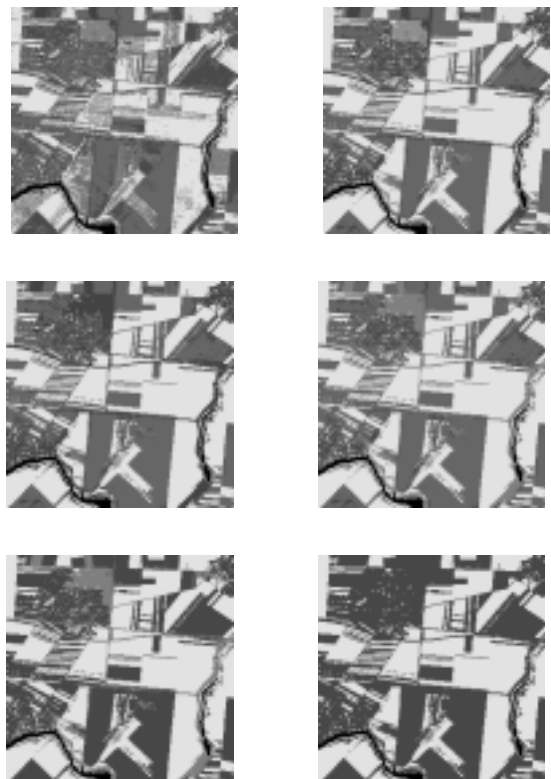


Figure 4 : figure contains labeled images at initial $\beta=500$ with decremental step $\Delta\beta = 1.05$

Clustering trajectory in scale space is the plot of intensity, which is between 0 and 1, versus inverse of temperature, that increases. The plot is for each component of point.

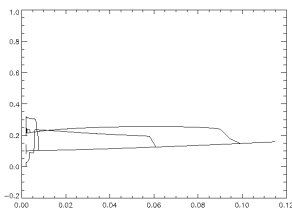


Figure 5: y_1 components of the trajectories of the cluster centers versus scale

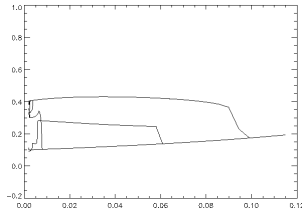


Figure 6: y_2 components of the trajectories of the cluster centers versus scale

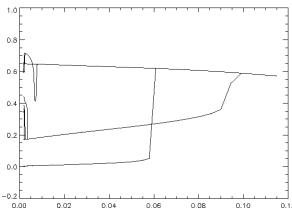


Figure 7: y_3 components of the trajectories of the cluster centers versus scale

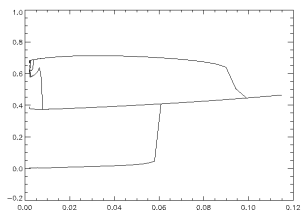


Figure 8: y_4 components of the trajectories of the cluster centers versus scale

4.2.2. SAR image

The following is an example for a Synthetic Aperture Radar - SAR image and Digital Elevation Model - DEM, but in this case, for beginning, each data point is set as a cluster center, like Wong algorithm. The two intensities form a feature vector for each pixel, (y_1, y_2) .

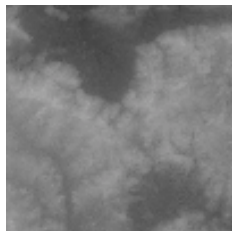


Figure 9: Digital Elevation Model - DEM

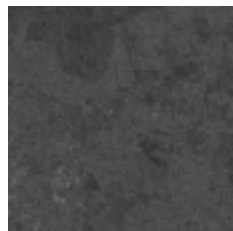


Figure 10: Synthetic Aperture Radar - SAR

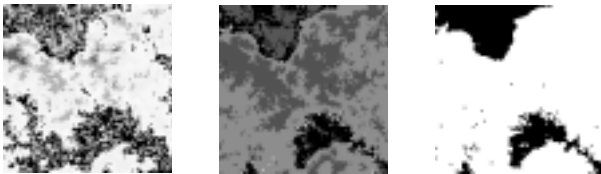


Figure 11 : figure contains labeled images at initial $\beta=2000$ with decremental step $\Delta\beta=1.05$

Clustering trajectory in scale space is the plot of intensity, which is between 0 and 1, versus inverse of temperature, that increases. The plot is for each component of point.

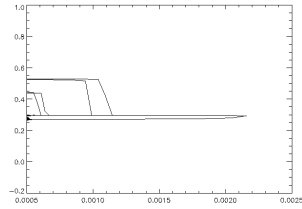


Figure 12: y_1 component of the trajectories of the cluster centers versus scale

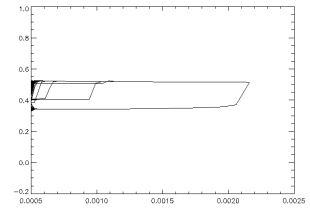


Figure 13: y_2 component of the trajectories of the cluster centers versus scale

5. CONCLUSIONS

In our application, the implementation of the algorithm is a multi-tree structure and with it, we can access easily and in a fast way to the informations, by rebuilding the image information content at any temperature. Therefore, we can visualize the clusters of image and we can choose the best number of clusters corresponding to the latter.

With the fast cluster center estimation algorithm we reduce the computational cost which allows us to start the melting procedure with the appropriate number of clusters according to this computation cost.

The multi-tree structure presents the possibility to accelerate the procedure by adjusting the error allowing cluster centers to merge together .

ACKNOWLEDGMENTS

We thank Alain Giros and CNES for providing us the SPOT data.

REFERENCES

1. Anil K. Jain, Richard C. Dubes, "Algorithms for Clustering Data", *Michigan State University*, 1988
2. "Digital Pattern Recognition", *Communication and Cybernetics*, 2001
3. James C. Tilton and William T. Lawrence, "Interactive Analysis of Hierarchical Image Segmentation," *Proceedings of the 2000 International Geoscience and Remote Sensing Symposium (IGARSS '00)*, Honolulu, HI, Jul. 24-28, 2000.
4. M. Schröder, H. Rehrauer, K. Seidel and M. Datcu, "Interactiv Learning and Probabilistic Retrieval in Remote Sensing Image Archives", *IEEE Trans. on Geoscience and Remote Sensing*, pp. 2288--2298, 2000

5. P.D.Fox, "On Merging Gradient Estimation with Mean-Tracking Techniques for Cluster Identification", 1997
6. Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Recognition"
7. Yiu-fai Wong and Edward C. Posner, , "A new Clustering Algorithm Applicable to Multispectral and Polarimetric SAR Images", *IEEE Transactions on Geoscience and Remote Sensing* , vol. 31, no. 3, May 1993.

Multimedia Knowledge Integration, Summarization and Evaluation

Ana B. Benitez
Department of Electrical Engineering
Columbia University
New York, NY 10027, USA
ana@ee.columbia.edu

Shih-Fu Chang
Department of Electrical Engineering
Columbia University
New York, NY 10027, USA
sfchang@ee.columbia.edu

ABSTRACT

This paper presents new methods for automatically integrating, summarizing and evaluating multimedia knowledge. These are essential for multimedia applications to efficiently and coherently deal with multimedia knowledge at different abstraction levels such as perceptual and semantic knowledge (e.g., image clusters and word senses, respectively). The proposed methods include automatic techniques (1) for interrelating the concepts in the multimedia knowledge using probabilistic Bayesian learning, (2) for reducing the size of multimedia knowledge by clustering the concepts and collapsing the relationships among the clusters, and (3) for evaluating the quality of multimedia knowledge using notions from information and graph theory. Experiments show the potential of knowledge integration techniques for improving the knowledge quality, the importance of good concept distance measures for clustering and summarizing knowledge, and the usefulness of automatic measures for comparing the effects of different processing techniques on multimedia knowledge.

KEYWORDS

Multimedia knowledge, knowledge integration, knowledge summarization, knowledge evaluation, concept distance, concept clustering, Bayesian networks

1. INTRODUCTION

This paper focuses on the integration, summarization and evaluation of multimedia knowledge representing perceptual or semantic information about the world depicted by, or related to an annotated image collection. Existing techniques are domain specific and do not generalize to arbitrary multimedia knowledge. Knowledge is usually defined as facts about the world and is often represented as concepts and relationships among the concepts, i.e., semantic networks. Concepts are abstractions of objects, situations, events or perceptual patterns in the world (e.g., a color pattern and concept Car); relationships represent interactions among concepts (e.g., color pattern one visually similar to color pattern two, and "sedan" specialization of "car").

Automatic knowledge integration, summarization and evaluation are essential for multimedia applications because multimedia applications often deal with multimedia knowledge at different abstraction levels such as perceptual and semantic knowledge (e.g., image clusters and word senses, respectively), which are usually extracted using different techniques. This diverse multimedia knowledge needs to be integrated to be used in a coherent and meaningful way by applications. Furthermore, it is often necessary to reduce the multimedia knowledge in order to keep the most representative and useful multimedia knowledge, before or after the knowledge integration. Hence, ways to quantify the consistency, completeness and conciseness of the multimedia knowledge are essential to evaluate and compare any of these knowledge integration and summarization techniques.

Related work on multimedia knowledge integration includes generic pattern classification techniques. In particular, Bayesian Networks (BNs) allow the discovery of the statistical structure of a domain but they are not optimized for multimedia. There is a lot of work in the literature on building and fine-tuning classifiers for recognition of objects and scenes in images [17,20,22], among other multimedia; however, these are usually constrained to a specific domain and trained on skewed data sets. Prior work on multimedia knowledge summarization has been limited to efforts in network and concept reduction such as EZWordNet [14] and VISAR [7]. EZWordNet.1-2 are coarser versions of the English dictionary WordNet generated by collapsing similar word senses and by dropping rare word senses [14]. This process is governed by five rules manually designed by researchers for WordNet so they are not applicable to other knowledge bases or other kinds of knowledge such as perceptual knowledge. WordNet organizes English words into sets of synonyms (e.g., "rock, stone") and connects them with semantic relations (e.g., generalization) [15]. VISAR is a hypertext system for the retrieval of textual captions [7]. One of the functionalities of the VISAR system is the representation of the retrieved citations as a network of key concepts and relationships. Several reduction operators are used in this process (e.g., replace two concepts for a common ancestor) but the reduction operators are again manually defined and

lacking generality. Furthermore, the methodology followed by some of the reduction operators is not clearly specified. Prior work relevant to multimedia knowledge evaluation includes manual evaluation of semantic ontologies [9] and automatic but application-oriented evaluation of multimedia knowledge [1].

This paper presents new methods for integrating, summarizing and evaluating multimedia knowledge. In contrast to prior work, our techniques are automatic and generic applying to any multimedia knowledge that can be expressed as a set of concepts (e.g., image clusters and word senses), relationships among concepts (e.g., feature descriptor similarity, and generalization and aggregation relations), and instances of concepts (i.e., images and/or text representing the concepts). These methods are developed and used within the IMKA (Intelligent Multimedia Knowledge Application) system [4], which aims at extracting useful knowledge from multimedia and implementing intelligent applications that use that knowledge. The IMKA system uses the MediaNet framework to represent multimedia knowledge [5], which is presented in the next section.

In the IMKA system, the integration of multimedia knowledge consists of discovering new relationships between the concepts in the knowledge. The proposed approach for multimedia knowledge integration is based on building meta-classifiers for the concepts and learning statistical dependencies among them using a Bayesian network. The summarization of multimedia knowledge aims at reducing the size of the knowledge (in terms of number of concepts and relationships) by grouping similar concepts together. The IMKA system summarizes multimedia knowledge by calculating the distances between concepts using a novel concept distance measure, by grouping similar concepts into super-concepts, and by collapsing the relationships among super-concepts. Knowledge summarization could either precede or proceed knowledge integration; in fact, multimedia knowledge can be integrated and summarized in multiple stages and in different order. This paper also proposes automatic techniques for measuring the consistency, the completeness and the conciseness of multimedia knowledge based on information theory and graph notions such as entropy and graph density. Experiments show the potential of knowledge integration techniques for improving the knowledge quality, the importance of good concept distance measures for clustering and summarizing knowledge, and the usefulness of automatic measures for comparing the effects of different processing techniques on multimedia knowledge.

The paper is organized as follows. Section 2 defines and exemplifies multimedia knowledge by presenting the multimedia knowledge representation framework MediaNet. Sections 3, 4 and 5 describe the proposed methods for multimedia knowledge integration,

summarization and evaluation, respectively. Section 6 presents the experiment setup and results in evaluating the proposed techniques. Finally, section 7 concludes with a summary and a discussion of future work.

2. MEDIANET

MediaNet is a unified knowledge representation framework that uses multimedia information for representing semantic and perceptual information about the world. The main components of MediaNet include concepts, relations among concepts, and media representing concepts and relationships. Examples of media are images, text and feature descriptors such as color histogram. MediaNet extends and differs from related work such as the Multimedia Thesaurus [21] in two ways: (1) in combining perceptual and semantic concepts in the same network, and (2) in supporting perceptual and semantic relationships that can be represented by media.

Concepts can represent either semantically meaningful objects (e.g., car) or perceptual patterns in the world (e.g., texture pattern). MediaNet models the traditional semantic relations such as generalization and aggregation but adds additional functionality by modeling perceptual relations based on feature descriptor similarity and constraints (e.g., condition on the distance of the color histograms). For example, perceptual knowledge for an image collection could be image clusters constructed based on visual and text feature descriptor similarity, and feature descriptor similarity and statistical relationships among the clusters [2]. Semantic knowledge for an annotated image collection could be the senses of the words in the textual annotations and semantic relationships among them as given by the electronic dictionary WordNet; the sense of each word could be disambiguated by matching the textual annotations of all the images in a cluster with the definitions of each possible sense [3]. In MediaNet, both concepts and relationships are defined and/or exemplified by multimedia information such as images, video, audio, graphics, text, and audio-visual feature descriptors. Feature descriptors can also be associated to the multimedia content (e.g., color histogram for images and tf*idf for textual annotations).

An example of multimedia knowledge represented using MediaNet is shown in Figure 1. Weights and probabilities can be assigned to the concepts, relationships, and media representations in MediaNet to capture positive and negative examples of concepts and user feedback, in other words, the process of extracting semantics from percepts (i.e., automatic text annotation using visual feature descriptors). MPEG-7 is an international standard for the description of multimedia that has the potential to revolutionize current multimedia representation and applications [16]. Multimedia knowledge expressed using

the MediaNet framework can be encoded using MPEG-7 description tools, in particular, using the tools for describing semantics and models of multimedia [5].

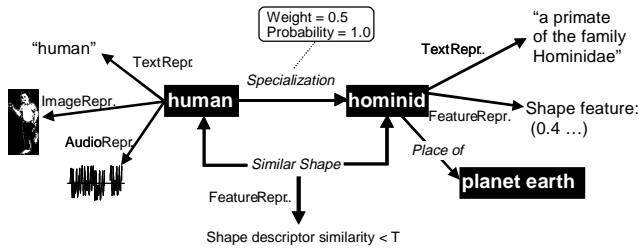


Figure 1: Example of multimedia knowledge.

3. MULTIMEDIA KNOWLEDGE INTEGRATION

The process of integrating multimedia knowledge consists of discovering relationships among concepts in multimedia knowledge to enable applications to make a coherent and meaningful use of diverse multimedia knowledge. As described in the previous section, the input multimedia knowledge is a set of concepts and relationships among concepts where both concepts and relationships can be either semantic or perceptual, and represented by different media such as images and text. Feature descriptors can also be associated with the images and the textual annotations.

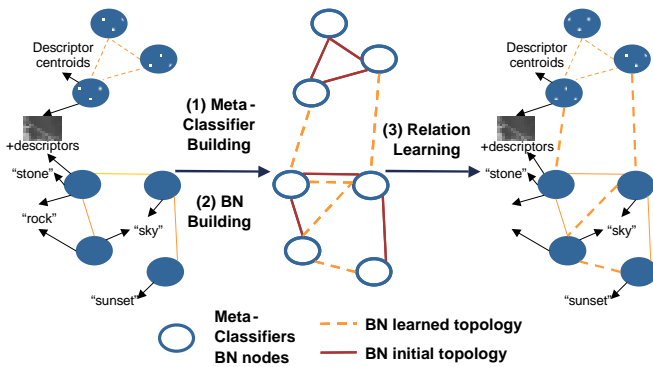


Figure 2. Multimedia knowledge integration process.

The proposed approach for multimedia knowledge integration consists of three steps, as shown in Figure 2: (1) building meta-classifiers for the concepts, (2) building a Bayesian Network (BN) whose nodes are the trained meta-classifiers and whose initial topology is the one of the known multimedia knowledge; and (3) adding the learned statistical relationships from the Bayesian network to the multimedia knowledge. This section describes each step. In Figure 2, dotted ellipses and dash lines represent perceptual concepts and relationships, respectively; plain

ellipses and plain lines represent semantic concepts and relationships, respectively; and arrow lines represent media representations of concepts. Other figures in this paper follow the same conventions.

3.1 Meta-Classifer Building

In the first step, one or more classifiers are built for each concept and, from these, a meta-classifier per concept. Meta-classifiers are trained to predict the presence of concepts in images or their associated textual annotations based on their visual and text feature descriptors.

A classification algorithm is a procedure for selecting a hypothesis from a set of alternatives that best fits a set of observations [8]. Classifiers basically learn how to predict the class (i.e., the value of the class attribute) of an input (given feature attributes of the input). The IMKA system uses a diverse set of classification algorithms: Naïve Bayes, Support Vector Machine (SVM), Neural Network (NN) and k-Nearest Neighbor (KNN) classifiers. The rationale for selecting each algorithm follows. The Naïve Bayes classifier is a very simple classifier. SVM and NN classifiers are slow at training but quick at classification. The KNN classifier can be trained quickly but it is slow at classification. Finally, the NN classifier requires large training sets whereas the KNN classifier does not.

A classifier is trained to predict the presence of a concept in an image based on a given combination of visual and textual feature descriptors associated with the image or its textual annotations. Therefore, the feature attributes input to each classifier for an image are a subset of the feature descriptors associated with the image. The class attribute that the classifier is trying to predict will have labels such as {presence, no presence} or {strong presence, weak presence, no presence} that indicate different strengths of the presence of a concept in an image. In the case of two-class classifiers (e.g., SVMs), several classifiers are used to learn more than two classes by using the one-per-class coding technique [8]. Multiple classifiers can be trained for the same concept using different combinations of feature descriptors or different classification algorithms. All the classifiers for a concept are combined into a meta-classifier, if needed, using bagging, boosting or stacking techniques [8]

The input feature attributes for building the classifiers of a concept are the visual and text feature descriptors associated with the images in the multimedia knowledge. The IMKA system uses several visual and text feature descriptors [2]. The supported visual feature descriptors are color histogram, Tamura texture, and edge direction histogram globally for images; and mean LUV color, aspect ratio, number of pixels, and position locally for automatically-segmented image regions. The IMKA system also implements two of the most popular schemes for representing textual annotations: tf*idf, term frequency

weighted by inverse document frequency; and $\log tf \cdot \text{entropy}$, logarithmic term frequency weighted by Shannon entropy of the terms over the documents. The feature descriptors can be normalized before being inputted to the classifiers by adjusting the mean and variance of each bin to zero and one, respectively. Feature descriptor normalization is desirable especially when classifiers deal with multiple feature descriptors.

Apart from the feature attributes, each image is associated a score indicating the strength of the presence of each concept in the image. These concept-presence scores are quantized uniformly into a given number of levels, which correspond to the labels of the class attribute for the classifiers. The concept-presence scores are automatically initialized during the multimedia knowledge extraction process, e.g., likelihood that a sense is the real meaning of a word annotating an image [3]. The initial values are propagated along the multimedia knowledge network. For example, if an image contains the concept Dog with a given probability, it also contains the concept Animal with, at least, the same probability because concept Animal is a generalization of concept Dog. In the IMKA system, concept-presence scores can be propagated not only through specialization/generalization relations but also through any relation from the relationship's source to target and/or vice versa given some weights. These propagation relation weights can be either learned or specified by an expert. Common values for propagation relation weights are shown in Table 1.

3.2 Bayesian Network Building

The second step in the multimedia knowledge integration process is to build a Bayesian network using the meta-classifiers constructed in the previous step and the network of multimedia knowledge.

Bayesian Networks (BNs), also known as Belief Networks, are directed graphical models that allow representing joint probability distributions of several random variables in a compact and efficient way [8]. The nodes of a Bayesian network represent the random variables, which are specified by conditional probability distributions. In the case of discrete random variables, the conditional probability distribution of a node is a table that lists the probability that the child node takes on each of its different values for each combination of the values of its parents. Several conditional independence assumptions apply to Bayesian networks. The lack of arcs among nodes represents conditional independence among the nodes. Moreover, a node in a Bayesian network is independent of its ancestors given its parents.

A Bayesian network is fully specified by the topology or structure of the graph, and the parameters of each conditional probability distribution. It is possible to learn both the structure and the parameters of a Bayesian

network for a given domain; however, the former is much harder than the latter. Learning the structure of Bayesian networks is especially hard when there is not prior knowledge of the Bayesian network's topology. However, once constructed for a domain, a Bayesian network can be used for probabilistic inference or reasoning about the domain; it can answer arbitrary questions about any conditional or joint probability of one or more of the random variables.

Bayesian networks are used during the multimedia knowledge integration process to learn statistical dependencies among concepts in the multimedia knowledge. Two reasons prompted the selection of Bayesian networks for this task. First, there are algorithms to learn statistical dependencies among the nodes in a Bayesian network by learning the structure of a Bayesian network. If the nodes in a Bayesian network represent concepts, then, the algorithms are actually learning statistical relationships among the concepts. The second reason is that once built, the Bayesian network can answer arbitrary probabilistic questions about the concepts, thus functioning as a knowledge classifier in itself.

A Bayesian network is built for multimedia knowledge that needs to be integrated as follows. The nodes of the Bayesian network are the meta-classifiers built as described in section 3.1; each node is thus indirectly representing a concept in the multimedia knowledge. The values of the nodes are the class labels of the meta-classifiers. The topology of the Bayesian network is initialized to the topology of the multimedia knowledge network; this is the best guess for the network topology based on prior knowledge. The initial multimedia knowledge from an image collection could be, for example, the perceptual and semantic knowledge directly extracted from the collection [2,3] or some multimedia knowledge summary. Bayesian networks cannot have directed cycles so certain arcs in the initial network may need to be removed to avoid directed cycles. The IMKA system uses the Markov Chain Monte Carlo (MCMC) algorithm called Metropolis-Hastings (MH) [10] to learn the topology of the Bayesian network. The training data for learning the Bayesian network is obtained by classifying the images in the multimedia knowledge using all the meta-classifiers.

3.3 Relationship Learning

The third step in the multimedia knowledge integration process is to add the newly learned statistical relationships among concepts to the multimedia knowledge.

The learned topology of the Bayesian network basically reveals important statistical relationships among the concepts in the multimedia knowledge. These relationships are compared with the known relationships among the concepts in the multimedia knowledge. A

statistical relationship is added to the multimedia knowledge for each arc between two concepts in the Bayesian network that does not already have a corresponding relationship in the initial multimedia knowledge. New statistical relationships could be added to the multimedia knowledge for each arc in the learned Bayesian network; however, some of these statistical dependencies are likely to be caused by already known relationships among the concepts.

4. MULTIMEDIA KNOWLEDGE SUMMARIZATION

This section presents techniques for automatically summarizing arbitrary multimedia knowledge by reducing the knowledge size in grouping similar concepts together. During this process, the number of concepts and relationships in the multimedia knowledge is reduced by grouping similar concepts into super-concepts and collapsing the relationships among the concepts in two super-concepts into a super-relationship.

The proposed approach for multimedia knowledge summarization consists of three steps, as shown in Figure 3: (1) obtaining the distances among the concepts in the multimedia knowledge; (2) clustering concepts based on the concept distances; and (3) reducing the concepts and the relationships in the multimedia knowledge based on the concept clusters. This section discusses each step in detail. In a preliminary stage, the least frequent concepts can be discarded from the multimedia knowledge and weights can be assigned to concepts for personalized knowledge summarization.

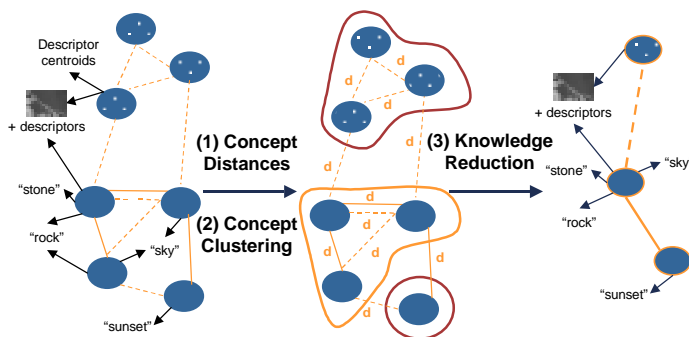


Figure 3. Multimedia knowledge summarization process.

4.1 Concept Distances

The first step in summarizing multimedia knowledge is to calculate the distances among concepts in the multimedia knowledge. Concept distances are calculated based on the concept statistics and the topology of the multimedia knowledge.

There are many proposed methods for calculating semantic distance or similarity among concepts in semantic concept networks such as WordNet. Some methods rely uniquely on the hierarchical specialization/generalization relationships among concepts [12,13] whereas others take into account all the semantic relations [19]. There are methods that use exclusively the concept network topology [13,19] while others combine both concept network topology information and text corpus statistics (e.g., concept probabilities) [12]. The most commonly used concept network for calculating semantic relatedness is WordNet [12,13,19]. Recent work evaluated five semantic distance measures using WordNet [6], including [12] and [13], in a real-word spelling error correction system in which [12] was found to outperform the rest.

The semantic measure described in [12] only considers the specialization/generalization concept hierarchy in WordNet. The weight or distance of the relationship between a child concept c and a parent concept $\text{par}(c)$ is the Information Content (IC), as defined in information theory, of the child concept given the parent concept, i.e., of encountering an instance of the child concept c given an instance of the parent concept $\text{par}(c)$, as follows:

$$\begin{aligned} \text{dist}(c, \text{par}(c))_{\text{Jiang}} &= \text{IC}(c/\text{par}(c)) = -\log(p(c/\text{par}(c))) \\ &= -\log(p(c)) + \log(p(\text{par}(c))) \end{aligned} \quad (1)$$

where $p(c)$ is the probability of encountering an instance of concept c . It is important to note that an instance of a child concept is always an instance of the parent concept and, therefore, $p(c \cap \text{par}(c)) = p(c)$. Then, the distance between any two concepts c and c' in the concept hierarchy reduces to the following expression:

$$\begin{aligned} \text{dist}(c, c')_{\text{Jiang}} &= \\ &2 * \log(p(\text{dcp}(c, c'))) - (\log(p(c)) + \log(p(c'))) \end{aligned} \quad (2)$$

where $\text{dcp}(c, c')$ is the deepest common ancestor of both concepts c and c' .

The IMKA system uses a novel concept distance measure that also uses concept statistics but is not limited to specialization/generalization concept relationships. The proposed concept distance measure generalizes measure [12] to an arbitrary concept network with different relations among concepts similar to measure [19]. Assuming binary relations, the distance of a relationship r between concept c and concept c' is the summation of the information content of concept c given concept c' and relationship r , and of the information content of concept c' given c and relationship r , as follows:

$$\begin{aligned} \text{dist}(c, c', r) &= \text{IC}(c/c', r) + \text{IC}(c'/c, r) \\ &= -\log(p(c/c', r)) - \log(p(c'/c, r)) \end{aligned} \quad (3)$$

where $p(c)$ is still the probability of encountering an instance of concept c ; $p(c/c', r)$ is the probability of encountering an instance of concept c given an instance of concept c' through relationship r . The intuition behind Equation (3) is the following: if a relationship makes two concepts almost interchangeably, i.e., $p(c/c', r)$ and $p(c'/c, r)$ are close to 1, the concepts are very similar given that relationship; if not, they are dissimilar. The distance between any two concepts is calculated as the total distance of the shortest distance path between the two concepts in the concept network. Therefore, the proposed concept distance satisfies the non-negative and inequality properties of a distance function.

If the concept network is a specialization/generalization concept hierarchy, the proposed concept distance measure (see Equation (3)) simplifies to the semantic distance measure [12] (see Equation (2)). In this case, concept c' is the parent of concept c , $c' = \text{par}(c)$, and r is the specialization/generalization relationship among them. The proof is straight forward realizing that an instance of concept c is always an instance of the parent concept $\text{par}(c)$ and, therefore, $\log(p(\text{par}(c)/c, r))$ is zero.

There are different approaches toward calculating the probabilities of concepts such as WordNet's senses in a text corpus. The approach often used in conjunction with Equation (2) obtains the frequency of each concept c as follows:

$$\text{freq}(c)_{\text{Richardson}} = \sum_{w \in \text{words}(c)} \frac{\text{freq}(w)}{|\text{concepts}(w)|} \quad (4)$$

where $\text{words}(c)$ is the set of words representing all the descendants of concept c in the generalization concept hierarchy including concept c , $\text{freq}(w)$ is the frequency of concept w in the text corpus (i.e., word occurrence), and $\text{concepts}(w)$ is defined as the set of concepts represented by word w [18]. As for WordNet's senses, this approach assumes concepts are represented by one or more words (e.g., "rock, stone"), and that the same word can represent more than one concept at the same time (e.g., "rock, stone" and "rock, candy"). Concept probabilities are then calculated from the concept frequencies as follows:

$$p(c)_{\text{Richardson}} = \frac{\text{freq}(c)}{N} \quad (5)$$

where N is the total number of distinct words representing, at least, one concept. Please, note that a concept that is an ancestor for all the rest of the concepts will have a probability of exactly 1.

Another way to understand this approach is that, first, strict concept frequencies are found for each concept without taking into account the specialized concepts or descendants; then, concept frequencies are propagated recursively through the specialization/generalization concept hierarchy from child concepts to direct parent concepts; and, finally, concept probabilities are calculated using Equation (5). In formulistic terms, this means that Equation (4) can be also expressed as follows:

$$\text{freq}(c)_{\text{Richardson}} = \sum_{c' \in \text{descendant } s(c)} \text{freq}'(c') \quad (6)$$

given

$$\text{freq}'(c)_{\text{Richardson}} = \sum_{w \in \text{words}'(c)} \frac{\text{freq}(w)}{|\text{concepts}(w)|} \quad (7)$$

where $\text{words}'(c)$ is defined as the set of words strictly representing concept c , without considering the words of the descendants of concept c .

The IMKA system generalizes this procedure of obtaining concept probabilities to an arbitrary concept network with several types of relationships among concepts. First, strict concept frequencies are found for each concept without taking into account related concepts. The multimedia knowledge contains the information of which concepts are instantiated in which images, and how many times a concept is instantiated in an image. For example, images are assigned to the concepts corresponding to the senses of all the words in the associated textual annotations, with the same frequency. The strict frequency of concept c is calculated as follows:

$$\text{freq}'(c) = \sum_{i \in \text{images}(c)} \text{freq}(c, i) \quad (8)$$

where $\text{freq}(c, i)$ is the number of times concept c is instantiated in image i . As an example, the concept House would have a frequency of five for an image whose textual annotations contain the word "house" five times.

In the second step, the concept frequencies are propagated in the concept network recursively through the relationships among concepts. Considering a relationship r that connects concepts c and c' , a different fraction of the frequency of concept c will be added to the frequency of concept c' based on relationship r , and vice versa. As an example, for the specialization/generalization relation, if concept c specializes concept c' , the frequency of concept c is added in full to the frequency of concept c' , but zero in the opposite direction. The propagation weights for each relation could be specified by an expert or learned automatically using machine learning techniques. In formulistic terms, the total frequency of concept c in the image collection is calculated as follows:

$$\text{freq}(c) = \text{freq}'(c) + \sum_{c' \in \text{neighbors}(c)} \sum_{r \in \text{relations}(c, c')} w(r) * \text{freq}(c') \quad (9)$$

where $\text{neighbors}(c)$ is the set of concepts directly connected to concept c through relationships, $\text{relations}(c, c')$ is the set of relationships connecting concepts c and c' , and $w(r)$ is the propagation weight for relationship r (see Table 1 for examples). To avoid loops, concepts are only allowed to contribute once to the frequency of another concept. The relations in the multimedia knowledge affect the concept frequencies and, therefore, the distances among the concepts through $w(r)$.

Finally, the concept probabilities are calculated based on the concept frequencies using the following formula:

$$p(c) = \min \left(1, \frac{\text{freq}(c)}{\sum_{c \in \text{concepts}(K)} \text{freq}'(c)} \right) \quad (10)$$

where K is the multimedia knowledge being summarized and $\text{concepts}(K)$ is the set of concepts in multimedia knowledge K . The concept frequencies are not exclusive that is the reason for dividing by the summation of strict concept frequencies instead of the summation of total concept frequencies. Also, due to the propagation of concept frequencies through relations other than specialization/generalization relations, the total frequency for some concepts may be larger than the summation of strict concept frequencies.

4.2 Concept Clustering

The second step in the multimedia knowledge summarization process is to cluster the concepts based on the distances among them. The concepts are clustered into a given number of clusters, the desired number of concepts in the multimedia knowledge summary.

The IMKA system supports several data clustering algorithms such as the k-means algorithm, the Ward algorithm, the k-Nearest-Neighbor algorithm (KNN), the Self-Organizing Map algorithm (SOM) and the Linear Vector Quantization algorithm (LVQ). A modified KNN clustering that generates a given number of clusters is selected for clustering the concepts. The KNN clustering algorithm was selected to cluster concepts in multimedia knowledge because of the continuity and the non-globular shape of the resulting clusters. Moreover, the KNN clustering algorithm does not use or require a specific distance function. The input of the KNN clustering algorithm [11] is the number of shared neighbors k_s , and the k nearest neighbors, in order from closest to farthest, for each data item to be clustered. The algorithm groups every pair of data items that have at least k_s shared neighbors. The vote of shared neighbors can be weighted according to their positions in the ordered k nearest

neighbors (e.g., sharing the second neighbor counting twice as much as sharing the third neighbor). In the KNN clustering algorithm, the number of resulting clusters is determined indirectly by the value of k_s .

The KNN clustering algorithm is modified slightly to generate a given number of clusters. Whereas the KNN clustering algorithm merges the clusters of two data items with at least k_s shared neighbors, the modified KNN clustering algorithm merges the clusters of the two data items with the largest number of shared neighbors until a given number of clusters is reached. Weighting of shared neighbors is also supported as well as the reduction of the number of shared neighbors based on data item weights. If a data item is more important (i.e., it has a higher weight), then, the data item will have fewer shared neighbors and be clustered with fewer other data items; it will tend to maintain its own identity. A centroid for each cluster is obtained as the data item in the cluster with maximum accumulated weighted shared neighbors to the rest of the data items in the cluster.

The concepts in the multimedia knowledge are clustered using the modified KNN clustering algorithm as follows. The input to the clustering algorithm is the desired number of concepts in the multimedia knowledge summary, and the k nearest concepts for each concept. Different shared neighbor weighting schemes [11] can be selected as well as individual weights for the concepts during clustering. The result of the concept clustering is a set of concept clusters and a centroid for each cluster.

4.3 Knowledge Reduction

The final step in the multimedia knowledge summarization process consists of generating the multimedia knowledge summary using the concept clusters and distances among concepts.

Once the clusters of concepts have been obtained, the multimedia knowledge summary is generated as follows. Each cluster becomes a super-concept in the summary and inherits the text and image representations of the cluster members. The most important text representation of the super-concept is the one of cluster centroid. If all the members of a cluster are semantic concepts, the super-concept will be labeled a semantic concept; otherwise, it will be labeled as a perceptual concept. The type of the super-concept is set to the type of the cluster centroid (e.g., visual concept based on color histogram similarity). Super-relationships are created between pairs of super-concepts based on the relationships between their cluster centroids in the original multimedia knowledge. The type of the super-relationship between two super-concepts is set to the type of the largest-distance relationship between the cluster centroids (e.g., generalization), as a worst-case scenario. Another possible approach for setting the type of a super-relationship would be selecting the most dominant

relationship (e.g., the one that appears most often between the concepts grouped by the two super-concepts).

5. MULTIMEDIA KNOWLEDGE EVALUATION

This section proposes several automatic application-independent techniques for evaluating the goodness of multimedia knowledge based on information and graph theory notions. These follow criteria used to manually evaluate and assess semantic ontologies and knowledge bases [9]. In contrast, many multimedia applications evaluate the quality of their multimedia knowledge by assessing the performance of complete applications using that knowledge, for example, automatic annotation performance of images [1].

A review on previous work on ontology evaluation has identified five criteria for the manual evaluation and assessment of semantic ontologies [9]. These criteria are the following: consistency, completeness, conciseness, expandability and sensitiveness. Expandability refers to the efforts required to add a new definition to an ontology, without altering the properties in the ontology. Sensitiveness relates to how small changes in a definition alter the set of well-defined properties guaranteed in an ontology. These two criteria are dependent on the way the knowledge is constructed, entered and maintained in the ontology so they are not considered in this section. This section proposes automatic ways for measuring the other three criteria -consistency, completeness and conciseness- for multimedia knowledge.

5.1 Consistency

Consistency refers to whether it is possible to obtain contradictory conclusions from valid input definitions. In terms of concept distances, the consistency of multimedia knowledge can be evaluated by calculating the spread of the total distances of the k shortest distance paths between every pair of concepts with respect to the shortest distance path. The larger the distance spread among concepts, the more inconsistent or contradictory the different paths connecting the concepts.

In formulistic terms, the proposed way to measure the inconsistency of multimedia knowledge K is as follows:

$$ICST(K) = \frac{\sum_{c,c' \in \text{concepts}(K)} \sum_{i=1}^{i=k} (d(c,c',i) - d(c,c',1))^2}{|\text{concepts}(K)|^2 * k} + 1 \quad (11)$$

where $\text{concepts}(K)$ is the set of concepts in multimedia knowledge K , k is the number of shortest distance paths considered between concepts, and $d(c,c',i)$ is the distance

between concepts c and c' through path i . The k shortest distance paths are ordered from shortest to longest distance starting at $i = 1$ at to $i = k$. The lower $ICST(K)$ for multimedia knowledge K , the more consistent the multimedia knowledge.

5.2 Completeness

Completeness refers to the completeness of both the ontology and the definitions in the ontology. The two proposed ways of evaluating the completeness of multimedia knowledge try to quantify the uniformity of the multimedia knowledge using entropy and graph density. The more uniform the multimedia knowledge, the more complete.

The first proposed way to calculate the uniformity of multimedia knowledge is by calculating the entropy of concepts, as follows:

$$CPT_H(K) = - \sum_{c \in \text{concepts}(K)} p(c) * \log(p(c)) \quad (12)$$

where $p(c)$ is the probability of concept c obtained as described in section 4.1. The higher $CPT_H(K)$ for multimedia knowledge K , the more complete the multimedia knowledge.

The second proposed way to calculate the uniformity of multimedia knowledge adapts the formula for graph density to weighted relationships, as follows:

$$CPT_D(K) = \frac{\sum_{r \in \text{relations}(K)} \text{weight}(r)}{|\text{concepts}(K)| * (|\text{concepts}(K)| - 1)} \quad (13)$$

where $\text{relations}(K)$ is the set of relationships in multimedia knowledge K , and $\text{weight}(r)$ is the weight of relationship r . If $d(r)$ is the distance of relationship r and d_{\max} is the maximum distance for a relationship, the weight of relationship r is obtained as follows:

$$\text{weight}(r) = \frac{d_{\max} - d(r)}{d_{\max}} \quad (14)$$

The higher $CPT_D(K)$ for multimedia knowledge K , the more complete the multimedia knowledge.

Another way to measure the completeness of the semantic part of multimedia knowledge would be to compare it with an existing ontology or thesaurus, preferably, in the same domain for which the multimedia knowledge was constructed (e.g., News or Nature). However, thesauri do not exist for every domain. Comparing the semantic knowledge with general-purpose thesaurus such as WordNet is also not desirable because these generic thesauri often treat different domains with different

degrees of detail (e.g., good coverage of Animal species but limited coverage of News-related concepts in WordNet).

5.3 Conciseness

Conciseness refers to whether all the information in the ontology is precise, necessary and useful. The conciseness of multimedia knowledge can be evaluated by applying Single-Value Decomposition (SVD) to the concept distance matrix to find the rank of the matrix. The number of non-null eigen values is compared with the number of concepts. The closer the number of non-null eigen values to the number of concepts, the more concise the multimedia knowledge.

In formulistic terms, the proposed way to calculate the inconsistency of multimedia knowledge K is as follows:

$$ICCS(K) = \frac{|\text{concepts}(K)| - \text{rank}(M)}{|\text{concepts}(K)|} \quad (15)$$

where M is the concept distance matrix, and rank(M) is the rank of the matrix M. The lower ICCS(K) for multimedia knowledge K, the more concise the multimedia knowledge.

6. EXPERIMENTS

Semantic and perceptual multimedia knowledge was integrated and summarized for a collection of images with associated textual annotations. The semantic and perceptual multimedia knowledge was generated for the annotated image collection using the techniques described in [2] and [3], respectively. The proposed multimedia knowledge evaluation measures were used to compare the proposed approaches with respect to several baseline approaches. The knowledge evaluation measures were also evaluated in these experiments by comparing their values for knowledge extracted from the image collection with the ones for random knowledge.

6.1 Experiment Setup

The test set was a collection of 25 images of plants from the Berkeley's CalPhotos collection (<http://elib.cs.berkeley.edu/photos/>). The images had short annotations in the form of keywords or well-formed phrases, as the example shown in Figure 4.

Perceptual knowledge was extracted by clustering the images using the k-means clustering algorithm based on the color histogram of the images, the log tf*entropy of the textual annotations and an integrated feature vector with both descriptors, and by finding relationships among the concepts based on statistical relations among the clusters [2]. Semantic knowledge was constructed by

disambiguating the sense of the words in the textual annotations using WordNet and the image clusters [3]. Relationships among the semantic concepts were discovered based on the relationships among words senses in WordNet. The resulting multimedia knowledge had 75 semantic concepts, 15 perceptual concepts, 67 generalization relations, 16 aggregation relations and 15 association relations.



Figure 4. Example of a plant image with corresponding textual annotations.

Summaries of different sizes were generated from the extracted multimedia knowledge using the propagation relation weights shown in Table 1, among others. Additional statistical relationships were discovered for one of the multimedia knowledge summaries using different classifiers – Naïve Bayes, SVM and 3-Nearest Neighbors (3NN) classifiers – trained on the integrated color histogram/log tf * entropy feature descriptor. The concept-presence scores were quantized into two values representing the presence and the absence of concepts in images, respectively.

Table 1: Propagation weights for some relations from source to target and vice versa.

Relation	Source to Target	Target to Source
Equivalence	1.0	1.0
Generalization	0.0	1.0
Aggregation	0.5	0.5
Statistical	0.25	0.25

The criteria to evaluate the multimedia knowledge integration and summarization were ICST(K), CPT_H(K), CPT_D(K) and ICCS(K) obtained as described in section 5. The performance of the proposed methods was compared to several baseline approaches. The baseline approach for multimedia knowledge summarization used the semantic distance [12] instead of the proposed concept distance. For multimedia knowledge integration, the baseline approach used the ZeroR classifier (which predicts the majority class). The four measures for multimedia knowledge evaluation were also evaluated by comparing the results obtained for the multimedia knowledge extracted from the image collection and for a randomized version of the multimedia knowledge.

6.2 Experiments Results

Table 2, Table 3 and Table 4 show the values for ICST(K), CPT_H(K) and CPT_D(K) obtained in the experiments evaluating the proposed techniques for evaluation, summarization and integration of multimedia knowledge, respectively. The values of ICCS(K) have been omitted because they were zero in all the instances.

Table 2 shows the results for the multimedia knowledge generated from the image collection using the proposed concept distance ($\text{dist}(c,c')$, see Equation (3)) and the semantic distance [12] ($\text{dist}(c,c')_{\text{Jiang}}$, see Equation (2)), and a random version of this multimedia knowledge. The random multimedia knowledge was generated by randomly changing the vertices of the relationships in the knowledge maintaining the types of the vertices. For example, if relationship r connected concept c and image i in the original multimedia knowledge, relationship r would connect any randomly chosen concept and image in the random multimedia knowledge. As expected, the random multimedia knowledge provides higher entropy than the extracted multimedia knowledge. On the other hand, the results for the distance spread and graph density of the extracted multimedia knowledge were better using the proposed concept distance. The semantic distance [12] did not perform very well because it is very conservative in calculating distances among concepts using only specialization/generalization relations.

Table 2: Inconsistency and completeness results for extracted multimedia knowledge using the proposed concept distance and the semantic distance [12], and for random multimedia knowledge.

	ICST	CPT_H	CPT_D
Extracted			
$\text{dist}(c,c')$	16.32	9.14	0.0122
$\text{dist}(c,c')_{\text{Jiang}}$	16.68	6.65	0.0084
Random	16.50	13.77	0.0119

Table 3 shows the results in summarizing the extracted multimedia knowledge into different number of concepts (i.e., knowledge summaries of 3, 9 and 18 concepts) using the proposed concept distance and the semantic distance [12]. Comparing the results in Table 2 and Table 3, the summarization of multimedia knowledge seems to increase the graph density and reduce the concept entropy. The summaries obtained using the proposed concept distance seem to consistently provide better overall results. As an example, although the graph density is higher for the summary of size 3 using semantic distance [12], the entropy of this summary is very small; the contrary seems to happen for the summary of size 18. Interestingly, the results for the summaries generated using semantic distance [12] show important oscillations

compared to the ones obtained with the proposed concept distance, which are more stable.

Table 3: Inconsistency and completeness results in summarizing extracted multimedia knowledge into different number of concepts using the proposed concept distance and the semantic distance [12].

	Distance	ICST	CPT_H	CPT_D
3	$\text{dist}(c,c')$	15.82	0.14	0.1666
	$\text{dist}(c,c')_{\text{Jiang}}$	1.95	0.08	0.4998
9	$\text{dist}(c,c')$	15.92	1.79	0.0833
	$\text{dist}(c,c')_{\text{Jiang}}$	0.00	1.10	0.0000
18	$\text{dist}(c,c')$	16.43	1.04	0.2157
	$\text{dist}(c,c')_{\text{Jiang}}$	14.87	2.53	0.0196

Finally, Table 4 shows the results obtained in integrating the multimedia knowledge summary of nine concepts (whose results are in the second row of Table 3) using different classification algorithms. The table also includes the number of new statistical relationships discovered using each classifier. The results for the ZeroR classifier (which predicts the majority class) are provided for baseline comparison. The tendency seems to be the following: the fewer statistical relationships are added to the multimedia knowledge, the larger the entropy and the distance spread, and the smaller the graph density of the integrated knowledge. The Naïve Bayes and SVM classifiers seem to provide the best overall results, which consistently range from average to good. It is also important to note the different effects of using different classifiers in the knowledge quality. For example, Naïve Bayes improves upon the non-integrated multimedia knowledge in all measures (second row of Table 3). The general tendency seems to be for the distance spread to decrease importantly, the entropy to decrease slightly, and the graph density to increase slightly when adding the new statistical relationships.

Table 4: Inconsistency and completeness results in integrating the multimedia knowledge summary of nine concepts using different classifiers. Column Rels is the number of new statistical relationships discovered using each classifier.

	ICST	CPT_H	CPT_D	Rels
Naïve Bayes	1.47	1.59	0.2500	12
SVM	1.23	0.64	0.2777	14
3NN	16.26	1.93	0.1250	3
ZeroR	1.24	0.07	0.3194	17

Some global conclusions that can be drawn from the experimentation follows. First, all the knowledge evaluation measures are useful in comparing different

multimedia knowledge, concept distance measures and classifiers, among others, except for the inconsistency measure. The inconsistency measure was not very useful for the multimedia knowledge in these experiments because it lacked equivalence relationships among concepts. However, the large variation of the results especially observed for knowledge summaries of different size seem to indicate the need to review the definitions of some of these measures. Second, the discovery of new statistical relationships using classifiers and Bayesian networks usually improves the quality of the knowledge. However, the use of different classifiers has different effects on the results, which might be due to the fact that the Bayesian network is learned for the meta-classifiers and not the concepts themselves. The Bayesian network could be learned using both the meta-classifiers and the concepts (i.e., the actual presence or absence of a concept in the images); however, this would require the unfeasible task of generating the ground truth of which concepts appear in which images. Third, summarizing multimedia knowledge seems to increase the graph density and decrease the concept entropy. The use of different concept distances in the knowledge summarization process seems to have a very important impact in the quality of the resulting summaries. The proposed concept distance seems to provide fairly consistent results for different summary sizes during knowledge summarization and different classifiers during knowledge integration.

7. CONCLUSIONS

This paper has presented novel techniques for automatically integrating, summarizing and evaluating arbitrary multimedia knowledge. In particular, it has proposed (1) a novel way to integrate classifiers and Bayesian networks to discover statistical relationships among concepts; (2) a new technique for calculating distances among concepts used by a modified KNN algorithm to cluster concepts with the purpose of generating summaries of multimedia knowledge; and (3) automatic ways of measuring the quality of multimedia knowledge in terms of consistency, completeness and conciseness. Experiments have shown the potential of knowledge integration techniques for improving the knowledge quality, the importance of good concept distance measures for clustering and summarizing knowledge, and the usefulness of automatic measures for comparing the effects of different processing techniques on multimedia knowledge.

Current work is focused on extending the evaluation of these techniques to more images, evaluation measures, classification algorithms and propagation relation weights, among others. Other important current work aims at improving the efficiency of the implementation of these techniques in terms of processing time and memory usage as well as the scalability of these methods for a large

number of images and concepts by developing heuristic approximations of some of proposed knowledge integration and summarization techniques. Future work will consist of implementing and evaluating applications that use the constructed multimedia knowledge for image classification and retrieval, automated concept illustration, and multimedia knowledge browsing, as well as, proposing a complexity-constraint framework for personalizing the quality values of the multimedia knowledge including complexity to specific user applications. Some of the remaining open issues are the extraction of multimedia knowledge from dynamic content such as video and audio, and the dynamic update of the knowledge based on user feedback or other external knowledge resources.

ACKNOWLEDGMENTS

This research is partly supported by a Kodak fellowship awarded to the first author of the paper.

REFERENCES

1. Barnard, K., P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M.I.Jordan, "Matching Words and Pictures", submitted to Special Issue on Text and Images, JMLR, 2002; also available at <http://www.cs.berkeley.edu/~kobus/research/publications/JMLR/JMLR.pdf>, 2002.
2. Benitez, A.B., and S.-F. Chang, "Perceptual Knowledge Construction From Annotated Image Collections", *International Conference On Multimedia & Expo (ICME-2002)*, Lausanne, Switzerland, Aug 26-29, 2002; also Columbia University ADVENT Technical Report #001, 2002.
3. Benitez, A.B., and S.-F. Chang, "Semantic Knowledge Construction From Annotated Image Collections", *International Conference On Multimedia & Expo (ICME-2002)*, Lausanne, Switzerland, Aug 26-29, 2002; also Columbia University ADVENT Technical Report #002, 2002.
4. Benitez, A.B., S.-F. Chang, and J.R. Smith, "IMKA: A Multimedia Organization System Combining Perceptual and Semantic Knowledge", *ACM International Conference on Multimedia (ACM MM-2001)*, Canada, Ottawa, Sep 30-Oct 5, 2001.
5. Benitez, A.B., J.R. Smith, and S.-F. Chang, "MediaNet: A Multimedia Information Network for Knowledge Representation", *SPIE Conference on Internet Multimedia Management Systems (IS&T/SPIE-2000)*, Vol. 4210, Boston, MA, Nov 6-8, 2000.
6. Budanitsky, A., and G. Hirst, "Semantic Distance in WordNet: An Experimental, Application-oriented

- Evaluation of Five Measures", *Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*, Pittsburgh, PA, June 2001.
7. Clitherow, P., D. Riecken, and M. Muller, "VISAR: A System for Inference and Navigation in Hypertext", *ACM Conference on Hypertext*, Pittsburgh, PA USA, Nov. 5-8, 1989.
 8. Duda, R.O., P.E. Hart, D.G. Stork, "*Pattern Classification*", John Wiley & Sons, Second Edition, United States of America, 2001.
 9. Gomez-Perez, A., "Evaluation of Taxonomic Knowledge in Ontologies and Knowledge Bases", *Workshop on Knowledge Acquisition (KAW-1999)*, Alberta, Canada, Oct. 16-21, 1999.
 10. Hastings, W.K., "Monte Carlo Sampling Methods Using Markov Chains and their Applications", *Biometrika*, Vol. 57, No. 1, pp. 97-109, 1970.
 11. Jarvis, R.A., and E.A. Patrick, "Clustering Using a Similarity Measure Based on Shared Near Neighbors", *IEEE Transaction on Computers*, Vol. c-22, No. 11, Nov. 1973.
 12. Jiang, J.J., and D.W. Conrath, "Semantic Similarity based on Corpus Statistics and Lexical Taxonomy", *International Conference on Research in Computational Linguistics*, Taiwan, 1997.
 13. Leacock, C., and M. Chodorow, "Combining Local Context and WordNet Similarity for Word Sense Identification", *Fellbaum*, pp. 265-283, 1998.
 14. Mihalcea, R., and D. Moldovan, "Automatic Generation of a Coarse Grained WordNet", *Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*, Pittsburgh, PA, June 2001.
 15. Miller, G.A., "WordNet: A Lexical Database for English", *Communications of the ACM*, Vol. 38, No. 11, pp. 39-41, Nov. 1995.
 16. MPEG Requirements Group, "MPEG-7: Context, Objectives and Technical Roadmap, V.12", ISO/IEC JTC1/SC29/WG11 MPEG99/N2861, Vancouver, July 1999.
 17. Paek, S., and S.-F. Chang, "The Case for Image Classification Systems Based on Probabilistic Reasoning", *IEEE International Conference on Multimedia and Expo (ICME-2000)*, New York, NY, USA, July/Aug 30-2, 2000.
 18. Richardson, R., and A.F. Smeaton, "Using WordNet in a Knowledge-Based Approach to Information Retrieval", Working paper, CA-0395, School of Computer Applications, Dublin City University, Ireland, 1995.
 19. Sussna, M., "Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network", *International Conference of Information and Knowledge Management (CIKM-1993)*, pp. 67-74, 1993.
 20. Szummer, M., and R. Picard, "Indoor-Outdoor Image Classification", *IEEE International Workshop in Content-Based Access to Image and Video Databases*, Bombay, India, Jan. 1998.
 21. Tansley, R., "The Multimedia Thesaurus: Adding A Semantic Layer to Multimedia Information", Ph.D. Thesis, Computer Science, University of Southampton, Southampton UK, August 2000.
 22. Vailaya, A., A. Jain, and H.J. Zhang, "On Image Classification: City vs. Landscape", *IEEE Workshop on Content-Based Access of Image and Video Libraries*, Santa Barbara, CA, USA, June 1998.

Object Boundary Detection for Ontology-based Image Classification^{*}

Lei Wang, Latifur Khan, and Casey Breen
Department of Computer Science
University of Texas at Dallas, TX 75083-0688
Email: [leiwang, lkhan, casey]@utdallas.edu

ABSTRACT

Technology in the field of digital media generates huge amounts of non-textual information, audio, video, and images, along with more familiar textual information. The potential for exchange and retrieval of information is vast and daunting. The key problem in achieving efficient and user-friendly retrieval in the domain of image is the development of a search mechanism to guarantee delivery of minimal irrelevant information (high precision) while insuring that relevant information is not overlooked (high recall). The traditional solution to the problem of image retrieval employs content-based search techniques based on color, histogram, texture or shape features. The traditional solution works well in performing searches in which the user specifies images containing a sample object, or a sample textural pattern, in which the object or pattern is indexed. One can overcome this restriction by indexing images according to meanings rather than objects that appear in images, although this will entail a way of converting objects to meanings. We have solved this problem of creating a meaning based index structure through the design and implementation of a concept-based model using domain dependent ontologies. An ontology is a collection of concepts and their interrelationships which provide an abstract view of an application domain. With regard to converting objects to meaning the key issue is to identify appropriate concepts that both describe and identify images. For this, first we need to identify all object boundaries accurately that appear in images. We propose an automatic scalable object boundary detection algorithm based on edge detection and region growing techniques. We also propose an efficient merging algorithm to join adjacent regions using an adjacency graph to avoid the over-segmentation of regions. To illustrate the effectiveness of our algorithm in automatic image classification we implement a very basic system aimed at the classification of images in the sports domain. By identifying objects in images, we show that our approach works well when objects in images have less complex organization.

1. INTRODUCTION

The development of technology in the field of digital media generates huge amounts of non-textual information, such as audio, video, and images, as well as more familiar textual information [17]. The potential for the exchange and retrieval of information is vast, and at times daunting. In general, users can be easily overwhelmed by the amount of information available via electronic means. The need for user-customized information selection is clear. The transfer of irrelevant information in the form of documents (e.g. text, audio, video) retrieved by an information retrieval system and which are of no use to the user wastes network bandwidth and frustrates users. This condition is a result of inaccuracies in the representation of the documents in the database, as well as confusion and imprecision in user queries, since users are frequently unable to express their needs efficiently and accurately. These factors contribute to the loss of information and to the provision of irrelevant information. Therefore, the key problem to be addressed in information selection in the domain of image is the development of a search mechanism which will guarantee the delivery of a minimum of irrelevant information (high precision), as well as insuring that relevant information is not overlooked (high recall).

Images consist of various objects, each of which may be used to effectively classify the image. The unstructured format of images tends to resist standard categorization and classification techniques. Traditional systems used to store and process multimedia images provide no means of automatic classification. The ability of these systems to retrieve relevant documents based on search criteria could be greatly increased if they were able to provide an accurate and semantic description of an image based on image content.

The traditional solution to the problem of image retrieval employs content-based search technique based on color, histogram, texture or shape features. The traditional solution works well in performing searches in which the user specifies images containing a sample object, or a sample textural pattern [9, 24, 28, 29, 30]. Should a user ask for an image depicting a basketball game, the results become less accurate. This is due to the fact that

though an image may contain a basketball, it does not necessarily depict a basketball game. In order to overcome the shortcomings of traditional technique in responding to image classification we have designed and implemented a concept-based model using ontologies [3, 17, 18, 4, 19, 20]. This model, which employs a domain dependent ontology, is presented in this paper. An ontology is a collection of concepts and their interrelationships, which can collectively provide an abstract view of an application domain [5, 14, 15].

In our system we would like to address two distinct questions: the extraction of the semantic concepts from the images and the construction of an ontology. With regard to the first problem, the extraction of semantic concepts, the key issue is to identify appropriate concepts that describe and identify images. We would like to make sure that irrelevant concepts will not be associated and matched, and that relevant concepts will not be discarded. In other words, it is important to ensure that high precision and high recall will be preserved during concept selection. To the best of our knowledge there are no attempts to connect images and concepts through the use of ontologies in any traditional image retrieval systems. We propose an automatic mechanism for the selection of these concepts (for more details see [3, 4]). In ontologies each concept is described by a set of features (objects). To select concept(s) for each image, we need first to identify object boundaries. For this, an object detection algorithm is invoked. In this paper we only address the problem of the extraction of object boundary. Although we detect object boundaries of images, we will not identify or label these objects. For this, we use neural networks to identify objects that appeared in images. Neural networks prove to be an effective method used to automatically find a wide range of patterns in sample data. After the objects have been identified, their identifications are fed into a concept selection module using ontologies to select appropriate concepts.

We propose an automatic scalable object boundary detection algorithm. Our algorithm works in three stages. First, we detect all edge pixels in images and divide pixels into two sets, edge pixel and region pixel sets. Second, we grow a region from the region pixel set surrounded by edges taken from the edge pixel set. Finally, we may merge adjacent regions using an adjacency graph to avoid over segmentation of regions and to detect boundary of objects accurately. To illustrate the effectiveness of our algorithm in automatic image classification we

implement a very basic system aimed at the classification of images in the sports domain. By identifying objects in images, we show that our approach works well when objects in images have less complex organization.

Section 2 of this paper discusses work related to image segmentation and ontologies for use in image retrieval, as well as the current systems used for image processing. Section 3 describes ontologies, and how they may be used to specify interrelationships among concepts that help draw meaningful conclusions about images. Section 4 describes outline of our approach. Section 5 presents elaborately our approach to detect object boundary. Section 6 presents preliminary result of our approach. Section 7 presents our conclusion and possible areas of future work.

2. RELATED WORK

Several systems exist today that attempt to classify images based on their content. Successful classification of an image and its contents relates directly to how well relevant images may be retrieved when a search is performed. Most image storing systems such as QBIC [24] and VisualSEEK [28, 29] limit classification mechanism to describing an image based on metadata such as color histograms [30], texture, or shape features [2, 25]. These systems have high success in performing searches in which the user specifies images containing a sample object, or a sample texture pattern. Should a user ask for an image depicting a basketball game, the results become less accurate. This is due to the fact that though an image may contain a basketball, it does not depict a basketball game. Systems that only contain metadata regarding the objects contained in an image cannot provide an accurate classification of the entire image.

Other systems attempt to provide images with a more precise description by analyzing other elements surrounding the images, such as captions [26, 27], or HTML tags on web pages [37]. These systems use this information to help classify the image and give it a meaningful description. This approach, tied together with metadata on images such as histograms, texture, and color sampling has the potential to yield high precision results in image classification. Examining the textual descriptions associated with an image provides additional information that may be used to help better classify the image. Unfortunately, this approach does not take into account the connections among individual

objects present in a sample image. Such connections provide useful information in the form of relationships among objects present in the image, which could be used to help classify the image's content.

To classify images we first need to segment images to detect objects. For this, simple color based segmentation techniques described in [13, 16, 31, 32, 34, 35, 36] may be used effectively to find regions rather than objects in a sample image. For example, Y. Deng et al. [36] propose a statistical method for segmenting color images based on a "J value." For region merge, agglomerative clustering technique is used. On the other hand, in our approach our main concern is to detect an object boundary in an image. For this, we detect edge pixels, and then use these pixels to locate regions. Furthermore, to avoid regions which are over-segmented, we propose a new method based on the use of an adjacency graph which is similar to [34]. However, to check the adjacency of two regions A. Trmeau et al. [34] use a minimum bounding rectangle that may identify some non adjacent regions as adjacent (false positive). We use a matrix method, which may substantially avoid false positives.

3. ONTOLOGIES

An ontology is a specification of an abstract, simplified view of the world that we wish to represent for some purpose [15]. Therefore, an ontology defines a set of representational terms that we call *concepts*. Inter-relationships among these concepts describe a target world. An ontology can be constructed in two ways, domain dependent and generic. CYC [22], WordNet [23], and Sensus are examples of generic ontologies. For our purposes, we choose a domain-dependent ontology. A domain-dependent ontology provides concepts in a fine grain, while generic ontologies provide concepts in coarser grain. The fine-grained concepts allow us to determine specific relationships among features in images that may be used to effectively classify those images.

Figure 1 illustrates an example ontology for the sports domain [11]. This ontology may be obtained from generic sports terminology and domain experts. The ontology is described by a directed acyclic graph (DAG). Here, each node in the DAG represents a concept. In general, each concept in the ontology contains a label name and feature vector. A feature vector is simply a set of features and their weights. Each feature may represent an object of an

image, such as a basketball or baseball. Note also that this label name connected to the feature is unique in the ontology. Furthermore, this label name is used to serve as an association of concepts to images. The concept of football may be further expanded to objects present in a football game (i.e. the features of the concept). For instance, a green field, goalposts, and football players would indicate the image is a football game. Should only one or two of the features common to a football game (as specified in the ontology) be present, a less specific classification of the image would be given. In other words, a more generic concept will be assigned to the image. An image containing only a football would be classified as an image containing a football, not as a football game. Furthermore, the weight of each feature of a concept may not be equal. In other words, for a particular concept some feature may serve as more discriminating as compared to some other; it will be assigned higher weight. For example, in the concept of a game of football the weight of goalpost feature is higher than the weight of the feature, green field.

3.1 Inter-relationships

In Ontologies, concepts are interconnected by means of inter-relationships. If there is a inter-relationship R , between concepts C_i and C_j , then there is also a inter-relationship R' between concepts C_j and C_i . In Figure 1, inter-relationships are represented by labeled arcs/links. Three kinds of inter-relationships are used to create our ontology: IS-A, Instance-Of, and Part-Of. These correspond to key abstraction primitives in object-based and semantic data models [1].

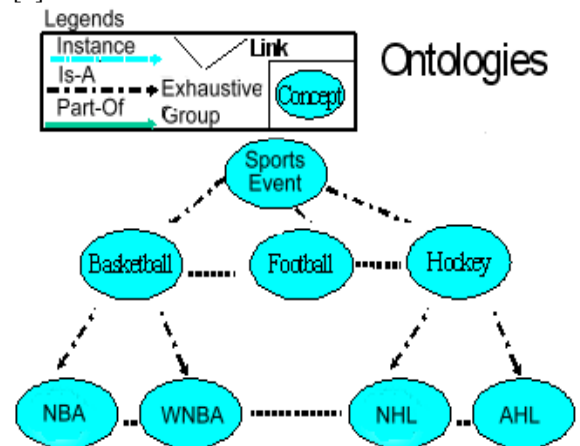


Figure 1. A Portion of an Ontology for the Sport Domain

IS-A: This inter-relationship is used to represent concept inclusion. A concept represented by C_j is

said to be a IS-A inter-relationship between C_i and C_j if it goes from generic concept C_i to specific concept, C_j represented by a broken line. Specialized concepts inherit all the properties of the more generic concept and add at least one property distinguishes them from their generalizations. For example, "NBA" inherits the properties of its generalization, "Professional" but is distinguished from other leagues by the type of game, skill of participant, and so on.

Instance-Of: This is used to show membership. A C_j is a member of concept C_i . Then the inter-relationship between them corresponds to an Instance-Of denoted by a dotted line. Player, "Wayne Gretzky" is an instance of a concept, "Player." In general, all players and teams are instances of the concepts, "Player" and "Team" respectively.

Part-Of: A concept is represented by C_j is Part-Of a concept represented by C_i if C_i has a C_j (as a part) or C_j is a part of C_i . For example, the concept "NFL" is Part-Of "Football" concept and player, "Wayne Gretzky" is Part-Of "NY Rangers" concept. Once the concepts have been fully identified in an ontology they may be used to draw a meaningful conclusion about an image based on its content. Objects identified by the neural network are used to develop relationships. These relationships specify useful information that is used to accurately classify a sample image.

4. PROPOSED SYSTEM

Our system combines the use of ontologies and neural networks as object identifiers to provide a high level of precision in the automatic classification of an image based on its content. This system circumvents the low precision classification techniques of other systems by examining the actual objects within an image and using them to discover relationships that reveal information useful in classifying the entire image. The concepts behind these relationships are held in our knowledge base of domain-dependant ontologies as described in section 3. Before feeding to ontologies or neural network, object boundaries are required to be identified in images. We now outline the steps taken to successfully process and classify an input image presented to our system.

4.1 Our Approach

In our system we would like to address two distinct questions: the extraction of the semantic concepts from the images and the construction of an

ontology. With regard to the first problem, the extraction of semantic concepts, the key issue is to identify appropriate concepts that describe and identify images. We propose an automatic mechanism for the selection of these concepts [3]. In ontologies each concept is described by a set of features (objects). To select concept(s) for each image, we need first to identify object boundaries. For this, an object detection algorithm (box 1 in Figure 2) is invoked. In this paper we only address the problem of the extraction of object boundary (see section 5). However, we will briefly touch upon some other issues.

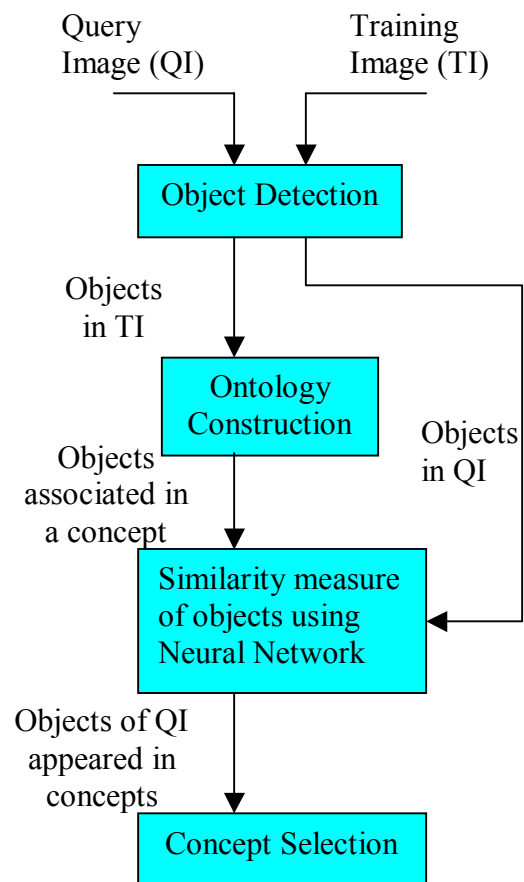


Figure 2. Flow of Our System

After identifying object boundaries in a query image to select concepts from ontologies, we identify objects that appear in the image using neural networks (box 3 in Figure 2) [4]. Neural networks prove to be an effective method used to automatically find a wide range of patterns in sample data [8]. Given a large amount of input data to work with, a neural network can automatically find the most dominant patterns of information. In

most cases, a neural network takes an input vector and maps it onto an output pattern. The result is similar to a black box that takes an input and produces the desired output. In the case of a neural network, the inside of this black box is actually a set of adjustable weights, each of which is applied to the input data in an attempt to map this data to the correct output. The ability of a neural network to map an input image to a specified output category makes neural networks a popular method for object identification.

After the objects have been identified, their identifications are fed into a concept selection module (box 4 in Figure 2). The ontologies use this information to provide a meaningful description of the image by selecting concepts based on image content (i.e., individual objects within the image). Our concept selection mechanism includes a novel, scalable disambiguation algorithm using a domain specific ontology. This algorithm will prune irrelevant concepts while allowing relevant concepts to become associated with images [3].

With regard to the second problem, we would like to build ontologies automatically (box 2 in Figure 2). This will be part of future work. For this, we will rely on a self-organizing tree (SOTA) that constructs a hierarchy from top to bottom [21]. To construct the tree we need to measure similarity between images. We would like to propose similarity between images based on the objects appeared in images similar to vector space model. Furthermore, each object in an image will be treated as a keyword along with its weight.

5. IMAGE SEGMENTATION

First, several pre-processing steps must be carried out to prepare the individual objects as input into the neural network. One of them is image segmentation. In our approach image segmentation process has three steps. First, we need to extract color edges from areas of different color. Second, based on the color edges we discovered in step one, we divide the image into several sub-regions by using region-growing techniques. In the final step, adjacent regions having the similar colors are merged together.

5.1 Edge Detection

In our method, we use the I color space [33]. Edge pixels are discovered by values of intensity, hue and saturation. So, at first, we need to apply color

conversion to transform all image pixels from the RGB color space to the I space. I, H and S stand for the value of intensity, hue and saturation correspondingly.

1	2	1
0	0	0
-1	-2	-1

HOE

1	0	-1
2	0	-2
1	0	-1

VOE

2	1	0
1	0	-1
0	-1	-2

NOE

0	1	2
-1	0	1
-2	-1	0

SOE

(x-1, y-1)	(x-1, y)	(x-1,y+1)
(x, y-1)	(x, y)	(x, y+1)
(x+1,y-1)	(x+1,y)	(x+1,y+1)

Figure 3: IHS Definitions

In Figure 3, HOE, VOE, NOE, and SOE stand for horizontal, vertical, northeast diagonal and northwest diagonal edge patterns respectively.

Using Figure 3 as a guide, we make the following definitions to carry out our calculations,

$$\begin{aligned}
 \text{HOE}(x, y)_i &= |I(x-1,y-1) + 2I(x,y-1) + I(x+1,y-1) \\
 &\quad - I(x-1,y+1) - 2I(x,y+1) - I(x+1,y+1)| \\
 \text{VOE}(x, y)_i &= |I(x-1,y-1) + 2I(x-1,y) + I(x-1,y+1) \\
 &\quad - I(x+1,y-1) - 2I(x+1,y) - I(x+1,y+1)| \\
 \text{NOE}(x, y)_i &= |I(x,y-1) + 2I(x-1,y-1) + I(x-1,y) \\
 &\quad - I(x+1,y) - 2I(x+1,y+1) - I(x,y+1)| \\
 \text{SOE}(x, y)_i &= |I(x,y-1) + 2I(x+1,y-1) + I(x+1,y) \\
 &\quad - I(x-1,y) - 2I(x-1,y+1) - I(x,y+1)| \\
 \text{MOE}(x, y)_i &= \max \{ \text{HOE}(x, y)_i, \text{VOE}(x, y)_i, \\
 &\quad \text{NOE}(x, y)_i, \text{SOE}(x, y)_i \}
 \end{aligned}$$

If $\text{MOE}(x, y)_i$ is greater than a threshold T_1 , the pixel (x, y) is an edge pixel [7]. Similarly, we use the same method to find values for H and S. If the value of MOE for H and S is more than threshold T_H and T_S correspondingly, the pixel (x, y) is also an edge pixel. The three thresholds discussed above are determined through experimentation. They may be adjusted to achieve better edge detection result. The pseudo code of edge detection is as follows.

```

Read image and save it in a two dimensional array
Pixel[imageWidth][imageHeight]
for (int y = 0; y < imageHeight; y++) {
  for (int x = 0; x < imageWidth; x++) {
    if ( (MOE(x, y)_I > T_I) OR (MOE(x, y)_H > T_H)
    OR (MOE(x, y)_S > T_S) )
  }
}

```

```

    Pixel[x][y] is an edge pixel
else
    Pixel[x][y] is an region pixel
}
}

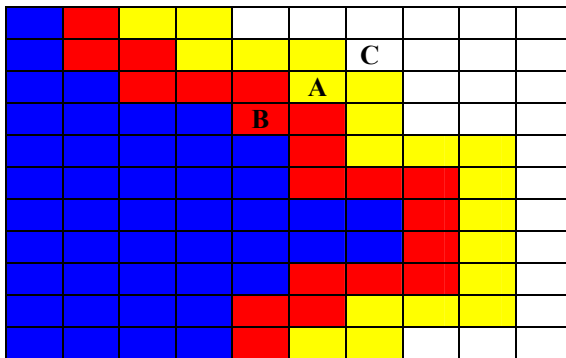
```


Figure 4. Pseudo code for Edge Detection


After edge detection, all image pixels are divided into two sets; the edge pixel set (EPS) and the region pixel set (RPS). We move on to the region growing calculations.


5.2 Region Growing

The detected edges cut the image into a set of regions. We pick a pixel from the RPS randomly as a seed for a new region, R_i . During region growing of R_i , all pixels in this region are moved out from the RPS and are assigned to this newborn region. After this region is fully grown, if the RPS is not empty, the algorithm simply picks a pixel randomly as a seed for another new region. This process continues until all pixels in the RPS are placed in a set of regions.



 &
 Pixels in the growing region R

 &
 Pixels not assigned yet


 Boundary pixels of the region R



 Outer neighbor pixels of the region R

Figure 5. Region Growing

The growth of the regions must satisfy certain criteria. If the criteria cannot be satisfied, the growth in the given direction will be stopped. A. Trémeau et al. introduced three criteria for region growing, one local homogeneity criterion (LHC) and two average

homogeneity criteria (AHC) [34]. We define p as the pixel to be processed, R is the set of pixels in the current region (possibly not fully-grown) and V is the subset of pixels from the current region which are neighbors to p . LHC states the color differences between p and its neighbors in R is sufficiently small. AHC1 states that the color difference between p and the mean of the colors in V is sufficiently small. AHC2 states that the color difference between p and the mean of the colors in R is sufficiently small. Each of the 3 criteria must be satisfied for p to be merged into R .

Growth of a region is as follows. First, the seed pixel is the only pixel that the region R has. Pixels of R are fallen into two categories such as boundary pixel (BP) and inner pixel (IP). A pixel is boundary pixel if at least one pixel among its 8 neighbor pixels is not in the region it belongs. On the other hand, a pixel is inner pixel if all its 8 neighbor pixels are in the region it belongs. At the beginning, the seed pixel is the only boundary pixel of the region. Next, we check the availability of 8 neighbor pixels of this boundary pixel. A pixel is available only when it is contained in RPS. This means the pixel is not an edge pixel and has not been assigned to some other region yet. If any of these pixels is available and satisfies the criteria, the pixel is qualified to be a member of R . After addition of a pixel into region R , it will be a new boundary pixel of the region. The inner pixels and boundary pixels of the region are also required to update. For example, in Figure 5, after adding pixel A into region R , A will be a new boundary (red) pixel. Pixel C will be a current neighbor (yellow) pixel of boundary pixel, A. Thus, pixel B is not a boundary pixel any more and will be an inner (blue) pixel. Based on these two characteristics, we keep checking and updating boundary pixels until the region stops to extend. Then, we can say the region is fully grown. The pseudo code is as follows.

```

int i = 0;
while (RPS is not empty) {
    i++;
    pick a pixel from RPS randomly as a seed
    and assign it to new set  $R_i$ 
    for each boundary pixel( $r$ ) of  $R_i$  {
        for each neighbor pixel( $n$ ) of  $r$  that is
        not in BP and IP
            if (LHC and AHC are satisfied for  $n$ )
            { Move the pixel,  $n$  from RPS to  $R_i$ ;
              Update RPS and  $R_i$ ; }
    }
}

```

Figure 6. Pseudo Code for Region Growing

5.3 Merging Adjacent Regions

We still encounter several shortcomings. First, it is possible to achieve some noise regions which may not be the true region. Second, it is still possible to cut one object into several sub regions even if it has a unique color. For example, a basketball could be divided into several sub regions due to its black lines (see second image of Figure 10). Intuitively, these two problems can be solved by merging adjacent regions. At first, we need to construct a region adjacency graph (RAG) based on regions [34]. In a RAG each vertex represents a sub region. An edge will appear to connect the two vertices, which stand for two adjacent regions. (Shown in Figure 7) The edges are weighted by color difference between these two regions.

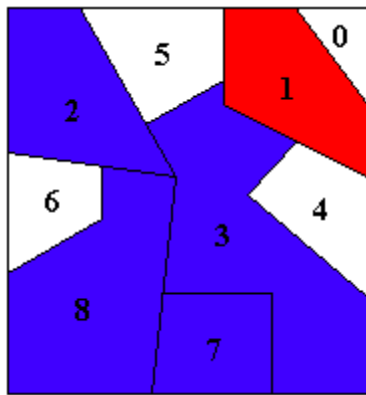
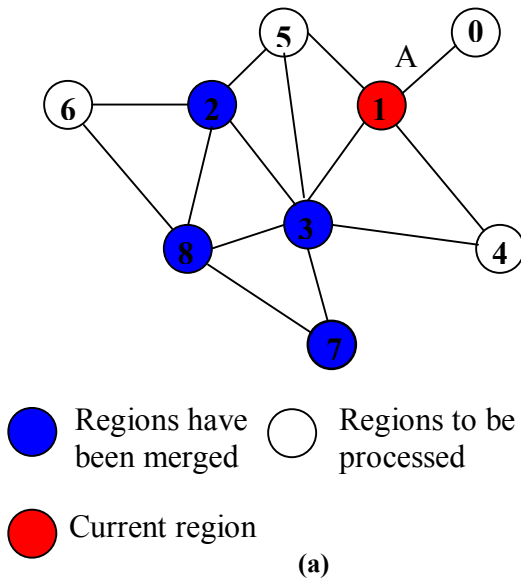


Figure 7. Region Adjacency Graph

To construct RAG, we have to know whether any two given regions are adjacent or not. Two following approaches can be used.

5.3.1 Minimum Bounding Rectangle Technique (MBRT)

In this approach, minimum bounding rectangle has been constructed [35]. Two regions are considered to be adjacent to each other if their minimum bounding rectangles overlap. Minimum bounding rectangle of a region not only encompasses the region but may also surround some regions which may contribute false positive (not true adjacent regions).

5.3.2 Matrix Oriented Technique (MOT)

Here we keep a two dimensional matrix where each cell corresponds to a pixel. Furthermore, content of the cell corresponds to a region index where the pixel belongs. Note that for edge pixel we have a special treatment: -1 will be used as a region index. To find adjacent regions, we simply scan matrix row-by-row and column-by-column. For example, in Figure 8, each gray pixel labeled by -1 is edge pixel, other pixels are region pixels and the number indicates the region index in which the pixel belongs to.

-1	-1	5	5	5	5	-1	3
2	-1	5	5	5	5	-1	3
2	-1	-1	5	5	-1	-1	3
2	2	-1	5	5	-1	3	3
2	2	-1	5	5	-1	3	3
2	2	2	-1	-1	3	3	3
2	2	2	2	2	-1	-1	3
2	2	2	2	2	2	-1	3
2	2	2	2	2	2	2	-1

(a)

2	-1	5	5	5	5	5	-1
2	-1	5	5	5	5	-1	-1
2	-1	5	5	5	5	-1	3
2	-1	5	5	5	5	-1	3
2	-1	-1	-1	-1	-1	-1	3
2	-1	4	4	4	4	-1	3
2	-1	4	4	4	4	-1	3
2	-1	4	4	4	4	4	-1
2	-1	4	4	4	4	4	-1

(b)

Figure 8. Examples of Adjacent Regions Detection

When we scan through the matrix row by row and column by column, and if the region index changes from a to b (say), we can say that the region a is adjacent to region b. For example, when we scan the first row in Figure 8(a), we know that region 5 and 3 are adjacent to each other. When we scan the seventh column in Figure 8(a), we know region 3 and 2 are adjacent. This method is easy to implement and the computation complexity is $O(n)$. On the other hand, MOT has a shortcoming. In some special cases, it may detect regions adjacent wrongly. For example, in Figure 8(b), when we scan the fifth row in the matrix, region 2 and 3 are declared as adjacent. However, these two regions are separated by six edge pixels. Now, the issue will arise such as: What is the maximum number of edge pixels used as a separator to determine that two regions are adjacent? This threshold depends on the edge detection result and the region size scale.

With regard to the first problem (i.e., noise region), based on the adjacency graph, first we identify noise regions. If a region only contains a small number of pixels, we declare this region is a noise region. For this, we merge the noise region to one of its neighbor regions that has smallest color difference. With regard to the second problem (i.e., over segmentation of sub regions), we merge adjacent regions by using a modified minimum spanning tree algorithm (MMSTA). In the MMSTA a threshold t_w is defined (see Figure 9). Furthermore, a tree will be constructed by adding an additional constraint: weight of each edge in the tree will fall below t_w . All regions in the tree compose an object. This is because color difference between a region and all its neighbor regions in the tree falls below t_w .

```

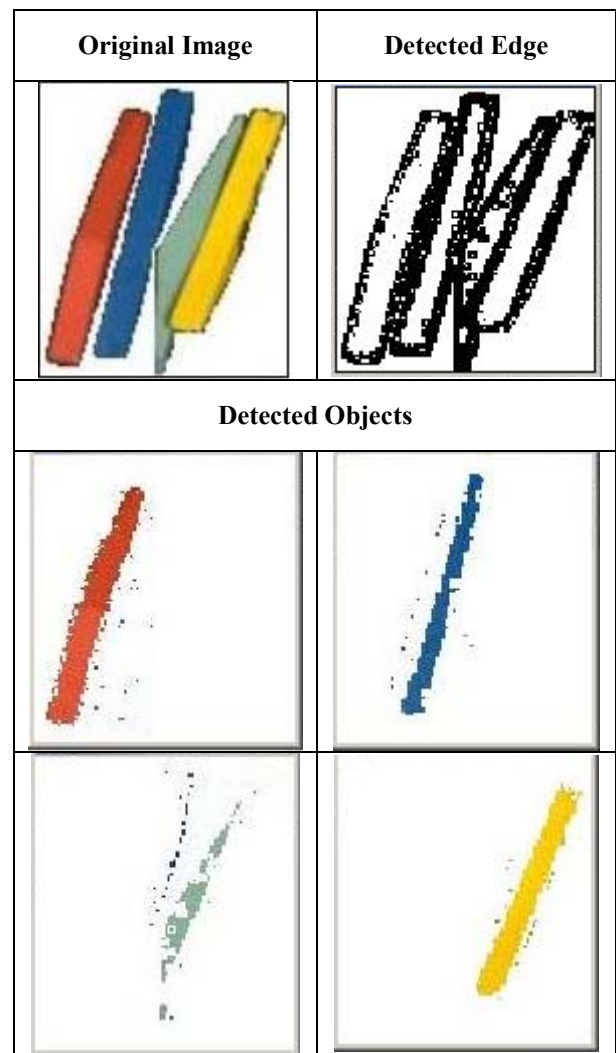
Calculate average color value for each  $R_i$ ;
Construct a RAG;
Define  $T_w$ ;
Sort all edges;
while ( still have edges and vertex not
added in the tree) {
    For each edge in order, test
    whether it creates a cycle in the
    tree we have thus far built or the
    weight is more than  $T_w$  -
    if so
        discard;
    else
        add to the tree.
}

```

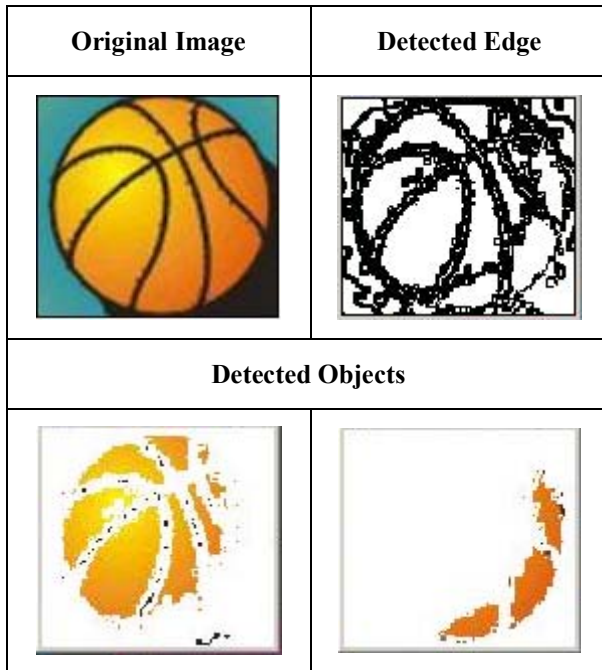
Figure 9. Pseudo Code for Merge Adjacent Regions

6. EXPERIMENTAL PRELIMINARY RESULTS

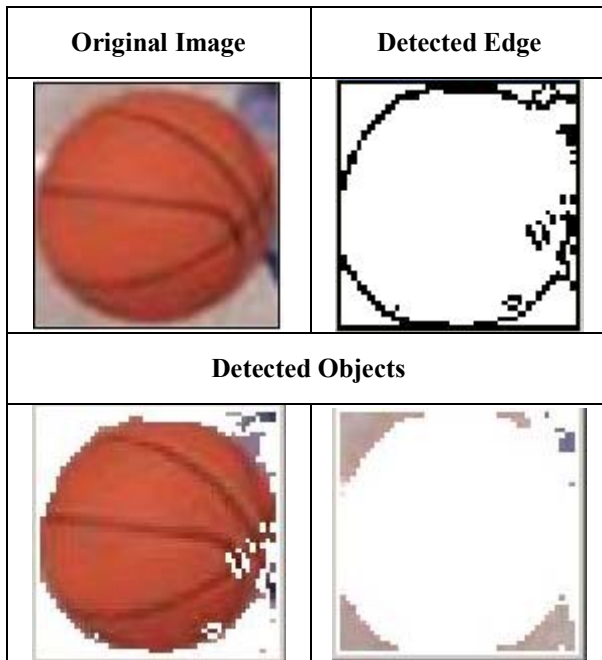
The object detection algorithm was tested using sample images found on the Internet. Here we reported results for only 4 images due to space limitations. These four images consist of varying degree of complex objects. The first image consists of 4 simple objects. The second and third images consist of basketball objects along with a set of lines. The fourth image consists of net, and player. Figure 10 shows these 4 images and displays detected objects. For each image, the original test images and edge detection results are shown first; and then all major detected objects are displayed.



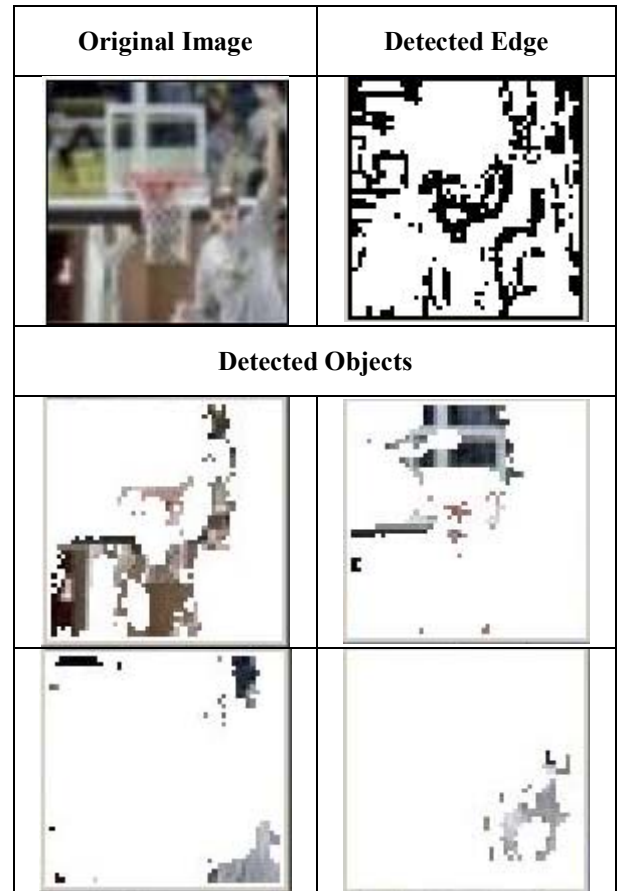
(a)



(b)



(c)



(d)

Figure 10. Image Segmentation Results

In the first image, each object has a unique color. We detected the four major objects correctly. The second and third images are more complicated, but the color distribution of the object is still simple, so the test results are also satisfactory. In the third image, objects are correctly classified. On the other hand, in the second image regions are correctly identified. However, merging adjacent regions algorithm fails to merge adjacent regions due to substantial change of hue property. Therefore, rather than unified one object two splitted objects are shown. Note that in the fourth image our algorithm fails to detect all objects correctly due to the presence of too many objects along with varying color.

7. CONCLUSIONS AND FUTURE WORKS

The success of ontology-based image classification model entirely depends on the detection of object boundaries. We have proposed an automatic

scalable object boundary detection algorithm based on edge detection, and region growing techniques. We have also proposed an efficient merging algorithm to join adjacent regions using adjacency graph to avoid over segmentation of regions. To illustrate the effectiveness of our algorithm in automatic image classification, we implement a very basic system aimed at the classification of images in the sports domain. By identifying objects in images, we have shown that our approach works well when objects in images have less complex organization. We would like to extend the work in the following directions. First, we would like to build ontologies automatically based on object similarity. Next, we will update weight of objects automatically appeared in images.

ACKNOWLEDGEMENTS

This research has been funded in part by NSF grant, NGS-0103709 with additional support from the Embedded Systems Center at University of Texas at Dallas.

REFERENCES

- [1] G. Aslan and D. McLeod, "Semantic Heterogeneity Resolution in Federated Database by Metadata Implantation and Stepwise Evolution", *The International Journal on Very Large Databases*, Vol. 18, No. 2, October 1999.
- [2] R. Barber, W. Equitz, C. Faloutsos, M. Fickner, W. Niblack, D. Petkovic, and P. Yanker, "Query by Content for Large On-Line Image Collections", *IEEE Journal*, 1995.
- [3] C. Breen, L. Khan, Arun Kumar and Lei Wang, "Ontology-based Image Classification Using Neural Networks," to appear in *SPIE*, Boston, MA, July 2002.
- [4] C. Breen, L. Khan and Arun Kumar, "Image Classification Using Neural Networks and Ontologies," to appear in *IEEE DEXA, International Workshop on Web Semantics*, France, Sept 2002.
- [5] M. A. Bunge, "Treatise on Basic Philosophy: Ontology: The Furniture of the World", Reidel, Boston, 1977.
- [6] S. F. Chang, J. R. Smith, "Extracting Multi-Dimensional Signal features for Content-Based Visual Query", in Proc. of *Visual Communications and Image Processing '95*, SPIE Volume 2501, pp. 995-1006, ed. T. Wu Lance, Bellingham, WA: The International society for Optical Engineering, 1995.
- [7] L. H. Chen, S. Chang, "Learning Algorithms and Applications of Principal Component Analysis", Image Processing and Pattern Recognition, Chapter 1, C. T. Leondes, Academic Press, 1998.
- [8] J. E. Dayhoff, "Neural Network Architectures An Introduction", VNR Press, 1990.
- [9] C. Djeraba, "When Image Indexing Meets Knowledge Discovery", in Proc. of *The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, Boston, MA, August 2000.
- [10] Joaquin Dopazo, Jose Maria Carazo, "Phylogenetic Reconstruction using an unsupervised growing Neural Network that adopts the Topology of a Phylogenetic Tree", *Journal of Molecular Evolution*, Volume 44, pp. 226-233 1997.
- [11] ESPN CLASSIC, <http://www.classicsports.com>.
- [12] Fritzke, Bernd, "Growing cell structures - a self-organizing network for unsupervised and supervised learning", *Neural Networks*, Volume 7, pp. 1141-1160 1994.
- [13] Y. Gong and H. J. Zhang, "An Effective Method for Detecting Regions of Given Colors and the Features of the Region Surfaces", in Proc. of *Symposium on Electronic Imaging Science and Technology: Image and Video Processing II*, pp. 274-285, San Jose, CA, February 1994, IS&T/SPIE.
- [14] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications Knowledge Acquisition," *An International Journal of Knowledge Acquisition for Knowledge-based Systems*, Volume 5, no. 2, June 1993.
- [15] N. Guarino, C. Masolo, and G. Vetere, "OntoSeek: Content-based Access to the Web," *IEEE Intelligent Systems*, Volume 14, no. 3, pp. 70-80, 1999.
- [16] N Ito, Y. Shimazu, T. Yokoyama, and Y. Matushita, "Fuzzy Logic Based Non-Parametric Color Image Segmentation with Optional Block Processing", in Proc. of *ACM*, 1995.
- [17] A. K. Jain, "Fundamentals of Digital Image Processing", Prentice Hall, Englewood Cliffs, NJ, 1989.
- [18] L. Khan, "Structuring and Querying Personalized Audio using Ontologies," in Proc. of *ACM Multimedia*, vol. 2, pp. 209-210, Orlando, FL, Nov 1999.
- [19] L. Khan and D. McLeod, "Audio Structuring and Personalized Retrieval Using Ontologies," in Proc. of *IEEE Advances in Digital Libraries, Library of Congress*, pp. 116-126, Bethesda, MD, May 2000.
- [20] L. Khan and D. McLeod, "Efficient Retrieval of Audio Information from Annotated Text Using Ontologies," in the Proc. of *ACM SIGKDD*

Workshop on Multimedia Data Mining, Boston, MA, August 2000.

[21] T. Kohonen, "Self -Organizing Maps", Second Edition, Springer 1997.

[22] D. B. Lenat, "Cyc: A Large-scale investment in Knowledge Infrastructure", *Communications of the ACM*, pp. 33-38, Volume 38, no. 11, Nov 1995.

[23] G. Miller, "Wordnet: A Lexical Database for English", in Proc. of *Communications of CACM*, Nov 1995.

[24] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, G. Taubin, "The QBIC Project: Querying Images by Content Using Color, Texture, and Shape", in Proc. of *Storage and Retrieval for Image and Video Databases*, Volume 1908, pp. 173-187, Bellingham, WA, 1993.

[25] A. Pentland, R.W. Picard, S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases", in Proc. of *Storage and Retrieval for Image and Video Databases II*, Volume 2185, pp. 34-47, Bellingham, WA, 1994.

[26] N. Row, and B. Frew, "Automatic Classification of Objects in Captioned Depictive Photographs for Retrieval", *Intelligent Multimedia Information Retrieval*, Chapter 7, M. Maybury, AAAI Press, 1997.

[27] A. F. Smeaton and A. Quigley, "Experiments on Using Semantic Distances between Words in Image Caption Retrieval," in Proc. of *The Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1995.

[28] J. R. Smith, S. F. Chang, "Automated Binary Texture Feature Sets for Image Retrieval", in Proc. of *The International Conference On Acoustic Speech and Signal Processing (ICASSP)*, pp. 2241-2244, Atlanta, GA, 1996.

[29] J. R. Smith, S. F. Chang, "Tools and Techniques for Color Image Retrieval", in Proc. of *The Symposium on Electronic Imaging: Science and Technology Storage and Retrieval for Image and Video Databases IV*, pp. 426-437, San Jose, CA, 1996.

[30] M. J. Swain, D. H. Ballard, "Color Indexing", *International Journal of Computer Vision*, 7(1), pp. 11-32, 1991.

[31] D. Tseng and C. Chang, "Color Segmentation Using Perceptual Attributes", in Proc. of *11th International Conference on Pattern Recognition*, pp. 228-231, Amsterdam, Holland, September 1992, IAPR, IEEE.

[32] S. Wong and W. K. Leow, "Color Segmentation and Figure-Ground Segregation of Natural Images", *IEEE Journal*, 2000.

[33] D. C. Tseng and C. H. Chang, "Color segmentation using perceptual attributes," In Proc. of *11th International Conference on Pattern Recognition*, pages 228-231, Amsterdam, HOLLAND, September 1992. IAPR, IEEE.

[34] A. Trémeau and P. Colantoni, "Regions adjacency graph applied to color image segmentation," *IEEE Transactions on Image Processing*, 1998.

[35] S. Wong and W. Leow, "Color segmentation and figure-ground segregation of natural images," in *Proc. Int. Conf. on Image Processing (ICIP 2000)*, volume 2, pages 120--123, 2000.

[36] Y. Deng, B.S. Manjunath, and H. Shin, "Color image segmentation", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[37] C. Frankel, M.J. Swain and V. Athitsos, "WebSeer: An Image Search Engine for the World Wide Web," *University of Chicago Technical Report TR-96-14*, July 31, 1996.

[38]. Chakrabarti, K., Ortega-Binderberger, M., Porkaew, K & Mehrotra, S. (2000) Similar shape retrieval in MARS. Proceeding of IEEE International Conference on Multimedia and Expo.

[39]. G. Lu and A. Sajjanhar, Region-based shape representation and similarity measure suitable for content-based image retrieval. Springer Verlag Multimedia Systems, 1999.

[40]. Ricardo Baeza-Yates, Berthier Ribeiro-Neto, *Modern Information Retrieval*, ISBN 0-201-39829-X, 1999.

Mammography Classification by an Association Rule-based Classifier

Osmar R. Zaiane
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
email: zaiane@cs.ualberta.ca

Maria-Luiza Antonie
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
email: luiza@cs.ualberta.ca

Alexandru Coman
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
email: acoman@cs.ualberta.ca

ABSTRACT

This paper proposes a new classification method based on association rule mining. This association rule-based classifier is experimented on a real dataset; a database of medical images. The system we propose consists of: a pre-processing phase, a phase for mining the resulted transactional database, and a final phase to organize the resulted association rules in a classification model. The experimental results show that the method performs well reaching over 80% in accuracy. Moreover, this paper illustrates, by comparison to other published research, how important the data cleaning phase is in building an accurate data mining architecture for image classification.

KEY WORDS

Mammography Mining, Image Classification, Document Categorization, Association Rules, Medical Images

1. Introduction

Association rule mining is one of the most important tasks in Data Mining and it has been extensively studied and applied for market basket analysis. In addition, building computer-aided systems to assist medical staff in hospitals is becoming of high importance and priority for many researchers. This paper describes the use of association rule mining in an automatic medical image classification process.

This paper presents a new method for building a classification system. It is based on association rule mining and it is tested on real datasets in an application for classifying medical images. This work is a significant extension and improvement of the system and algorithm we developed and presented in [1]. The novelty is in the data cleaning and data transformation techniques as well as in the algorithm used to discover the association rules. This paper illustrates the importance of data cleaning in applying data mining techniques in the context of image content mining.

The high incidence of breast cancer in women, especially from developed countries, has increased significantly in recent years. The etiologies of this disease are not clear and neither are the reasons for the increased number of cases. Currently there are no methods to prevent breast

cancer, that is why early detection represents a very important factor in cancer treatment and allows reaching a high survival rate. Mammograms are considered the most reliable method in early detection of cancer. Due to the high volume of mammograms to be read by physicians, the accuracy rate tends to decrease and automatic reading of digital mammograms becomes highly desirable. It has been proven that double reading of mammograms (consecutive reading by two physicians or radiologists) increased the accuracy, but at high costs. That is why the computer aided diagnosis systems are necessary to assist the medical staff to achieve high efficiency and effectiveness.

The methods proposed in this paper classify the digital mammograms into three categories: normal, benign and malign. The normal ones are those characterizing a healthy patient, the benign ones represent mammograms showing a tumor, but that tumor is not formed by cancerous cells, and the malign ones are those mammograms taken from patients with cancerous tumors. Generally, the most errors occur when a radiologist must decide between the benign and malign tumors. Digital mammograms are among the most difficult medical images to be read due to their low contrast and differences in the types of tissues. Important visual clues of breast cancer include preliminary signs of masses and calcification clusters. Unfortunately, at the early stages of breast cancer, these signs are very subtle and varied in appearance, making diagnosis difficult, challenging even for specialists. This is the main reason for the development of classification systems to assist specialists in medical institutions. Since the data that physicians and radiologists must deal with increased significantly, there has been a great deal of research done in the field of medical images classification. With all this effort, there is still no widely used method to classify medical images. This is because this domain requires high accuracy. Also misclassifications could have different consequences. False negatives could lead to death while false positives have a high cost and could cause detrimental effects on patients. For automatic medical image classification, the rate of false negatives has to be very low if not zero. It is important to mention that manual classification of medical images by professionals is also prone to errors and the accuracy is far from perfect. Another important factor that influences the

success of automatic classification methods is working in a team with medical specialists, which is desirable but often not achievable. The consequences of errors in detection or classification are costly. Mammography reading alone cannot prove that a suspicious area is malignant or benign. To decide, the tissue has to be removed for examination using breast biopsy techniques. A false positive detection causes an unnecessary biopsy. Statistics show that only 20-30 percentage of breast biopsy cases are proved cancerous. In a false negative detection, an actual tumor remains undetected that could lead to higher costs or even to the cost of a patient's life.

In addition, the existing tumors are of different types. These tumors are of different shapes and some of them have the characteristics of normal tissue. All these things make the decisions that are made on such images even more difficult. Different methods have been used to classify and detect anomalies in medical images, such as wavelets [3, 13], fractal theory [7], statistical methods [5] and most of them used features extracted using image processing techniques [11]. In addition, some other methods were presented in the literature based on fuzzy set theory [2], Markov models [6] and neural networks [4, 8]. Most of the computer-aided methods proved to be powerful tools that could assist medical staff in hospitals and lead to better results in diagnosing a patient. We have presented preliminary experiments using our first generation associative classifier on mammograms in [1]. The classification accuracy achieved then was 69.11%. Our new method for visual feature extraction and modelling as well as our new algorithm presented in this paper allows us to achieve an accuracy of 80.33%. Moreover, our new method manages to model the classifier in a reasonable number of rules (10 times less than the previous version), thus allowing a medical professional to update the rules manually to encode their own expertise and reach even better accuracy.

The rest of the paper is organized as follows. Section 2 describes the feature extraction phase as well as the cleaning phase. The following section presents the new association rule-based method used to build the classification system. Section 4 describes how the classification system is built using the association rules mined. Section 5 introduces the data collection used and the experimental results obtained, while in the last section we summarize our work and discuss some future work directions.

2. Data Cleaning and Feature Extraction

This section summarizes the techniques used to enhance the mammograms as well as the features that were extracted from images. The result of this phase is a transactional database to be mined in the next step of our system. Indeed, we model the images with a set of transactions, each transaction representing one image with the visual features extracted as well as other given characteristics along with the class label.

2.1 Pre-processing phase

Since real-life data is often incomplete, noisy and inconsistent, pre-processing becomes a necessity [10]. Two pre-processing techniques, namely Data Cleaning and Data Transformation, were applied to the image collection. Data Cleaning is the process of cleaning the data by removing noise, outliers etc. that could mislead the actual mining process. In our case, we had images that were very large (typical size was 1024 x 1024) and almost 50% of the whole image comprised of the background with a lot of noise. In addition, these images were scanned at different illumination conditions, and therefore some images appeared too bright and some were too dark. The first step toward noise removal was pruning the images with the help of the crop operation in Image Processing. Cropping cuts off the unwanted portions of the image. Thus, we eliminated almost all the background information and most of the noise. An example of cropping that eliminates the artefacts and the black background is given in Figure 1 (a-b).

Since the resulting images had different sizes, the x and the y coordinates were normalized to a value between 0 and 255. The cropping operation was done automatically by sweeping horizontally through the image. The next step towards pre-processing the images was using image enhancement techniques. Image enhancement helps in qualitative improvement of the image with respect to a specific application [9]. Enhancement can be done either in the spatial domain or in the frequency domain. Here we work with the spatial domain and directly deal with the image plane itself. In order to diminish the effect of over-brightness or over-darkness in images, and at the same time accentuate the image features, we applied the Histogram Equalization method, which is a widely used technique. The noise removal step was necessary before this enhancement because, otherwise, it would also result in enhancement of noise. Histogram Equalization increases the contrast range in an image by increasing the dynamic range of grey levels [9]. Figure 1 (c) shows an example of histogram equalisation after cropping.

2.2 Feature Extraction

The feature extraction phase is needed in order to create the transactional database to be mined. The features that were extracted were organized in a database, which is the input for the mining phase of the classification system. The extracted features are four statistical parameters: mean, variance, skewness and kurtosis; the mean over the histogram and the peak of the histogram.

The general formula for the statistical parameters computed is the following:

$$M_n = \frac{\sum (x - \bar{x})^n}{N} \quad (1)$$

where N is the number of data points and n is the order of

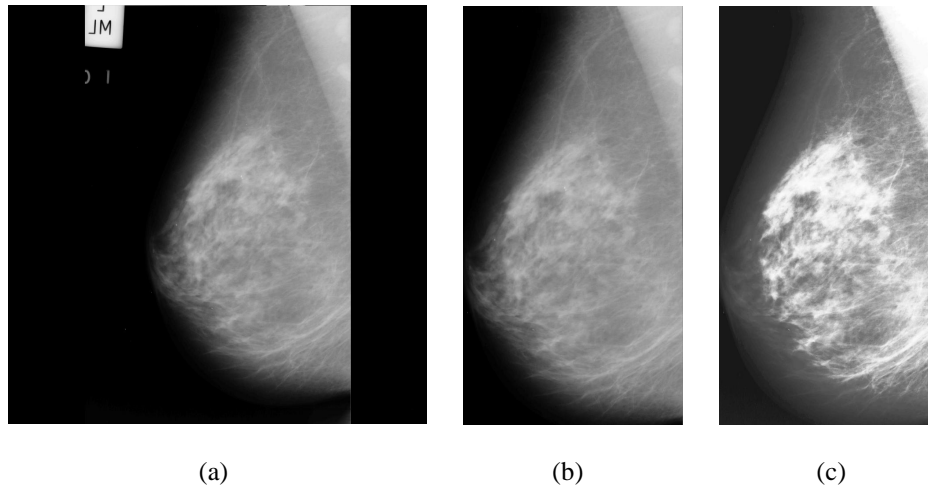


Figure 1. Pre-processing phase on an example image: (a) original image; (b) crop operation; (c) histogram equalisation

the moment. The skewness can be defined as:

$$Sk = \frac{1}{N} * \left(\frac{(x - \bar{x})}{\sigma} \right)^3 \quad (2)$$

and the kurtosis as:

$$kurt = \frac{1}{N} * \left(\frac{(x - \bar{x})}{\sigma} \right)^4 - 3 \quad (3)$$

where σ is the standard deviation.

2.3 Transactional Database Organization

All the extracted features presented above have been computed over smaller windows of the original image. The original image was split initially in four parts, as shown in Figure 2, for a better localization of the region of interest. In addition, the features extracted were discretized over intervals before organizing the transactional data set.

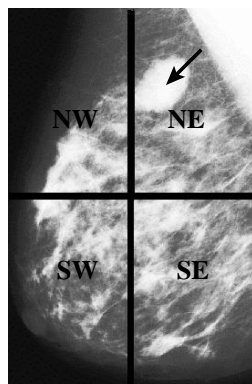


Figure 2. Mammography division

When all the features were extracted the transactional database to be mined was built in the following way. For

the normal images, all the features extracted were attached to the corresponding transaction, while for those characterizing an abnormal mammogram only the features extracted from abnormal parts were attached. (e.g. for the mammogram presented in Figure 2 only the features extracted for the NE quadrant (the arrow in the figure points to the tumor) were attached; if the mammogram would have been a normal one the features extracted for all the splits would have been attached). This new data cleaning stage allows us to find higher quality rules, discriminating better among the categories.

This is a new organization that we propose. In [1] the features of all quadrants were kept regardless of whether they were normal or cancerous. In addition some other descriptors from the original database were attached, such as breast position, type of tissue, etc. In this current work, in addition to selecting quadrants with tumors from abnormal mammograms, we also dropped those additional features from the database because some of them may not be available in other datasets, while others (breast position) proved to mislead the classification process.

3. Association Rule based Classification by Category

This section introduces the new classification method (ARC-BC=association rule based classification by category) that we propose to be applied to the image data collection. It mines the data set by classes instead of mining the entire data set at once. This algorithm was first proposed for text classification in [14].

The transactional database consists of transactions as follows. If an object O_i is assigned to a set of categories $C = \{c_1, c_2, \dots, c_m\}$ and after preprocessing phase the set of features $F = \{f_1, f_2, \dots, f_n\}$ is retained, the following transaction is used to model the object: $O_i : \{c_1, c_2, \dots, c_m, f_1, f_2, \dots, f_n\}$ and the association rules are

discovered from these transactions.

In this approach (Figure 3), each class is considered as a separate training collection and the association rule mining applied to it. In this case, the transactions that model the training documents are simplified to $O_i : \{C, t_1, t_2, \dots, t_n\}$ where C is the category considered.

In our algorithm we use a constraint so that only the rules that could be used further for classification are generated. In other words, given the transaction model described above, we are interested in rules of the form $O \Rightarrow c_i$ where $O \subseteq O$ and $c_i \subseteq C$. To discover these interesting rules efficiently we push the rule shape constraint in the candidate generation phase of the apriori algorithm in order to retain only the suitable candidate itemsets. Moreover, at the phase for rule generation from all the frequent k-itemsets, we use the rule shape constraint again to prune those rules that are of no use in our classification.

Algorithm ARC-BC Find association rules on the training set of the transactional database when the collection is divided in subsets by category

Input A set of objects (O) of the form $O_i : \{c_i, f_1, f_2, \dots, f_n\}$ where c_i is the category attached to the object and f_j are the selected features for the object; A minimum support threshold σ ; A minimum confidence threshold;

Output A set of association rules of the form $f_1 \wedge f_2 \wedge \dots \wedge f_n \Rightarrow c_i$ where c_i is the category and f_j is a feature;

Method:

- (1) $C_1 \leftarrow \{\text{Candidate 1 term-sets and their support}\}$
- (2) $F_1 \leftarrow \{\text{Frequent 1 term-sets and their support}\}$
- (3) for ($i \leftarrow 2; F_{i-1} \neq \emptyset; i \leftarrow i + 1$) do{
- (4) $C_i \leftarrow (F_{i-1} \bowtie F_{i-1})$
- (5) $C_i \leftarrow C_i - \{c \mid (i-1) \text{ item-set of } c \notin F_{i-1}\}$
- (6) $O_i \leftarrow \text{FilterTable}(O_{i-1}, F_{i-1})$
- (7) foreach object o in O_i do {
- (8) foreach c in C_i do {
- (9) $c.\text{support} \leftarrow c.\text{support} + \text{Count}(c, o)$
- (10) }
- (11) }
- (12) $F_i \leftarrow \{c \in C_i \mid c.\text{support} > \sigma\}$
- (13) }
- (14) $\text{Sets} \leftarrow \bigcup_i \{c \in F_i \mid i > 1\}$
- (15) $R = \emptyset$
- (16) foreach itemset I in Sets do {
- (17) $R \leftarrow R + \{I \Rightarrow \text{Cat}\}$
- (18) }

In ARC-BC algorithm step (2) generates the frequent 1-itemset. In steps (3-13) all the k-frequent itemsets are generated and merged with the category in C_1 . Steps (16-18) generate the association rules.

4. Building the Classifier

This section describes how the classification system is built and how a new image can be classified using this system.

First, there are presented a number of pruning techniques that were used in our experiments and second, the process of classifying a new image is described.

4.1 Pruning Techniques

The number of rules that can be generated in the association rule mining phase could be very large. There are two issues that must be addressed in this case. The first is that a huge number of rules could contain noisy information which would mislead the classification process. The second is that a huge set of rules would extend the classification time. This could be an important problem in applications where fast responses are required. In addition, in a medical application, it is reasonable to present a small number of rules to medical staff for further study. When the set of rules is too large, it becomes unrealistic to manually sift through it for editing.

The pruning methods that we employ in this project are the following: eliminate the specific rules and keep only those that are general and with high confidence, and prune some rules that could introduce errors at the classification stage. The following definitions introduce the notions used in this subsection.

Definition1 Given two rules $T_1 \Rightarrow C$ and $T_2 \Rightarrow C$ we say that the first rule is a general rule if $T_1 \subseteq T_2$.

The first step of this process is to order the set of rules. This is done according to the following ordering definition.

Definition2 Given two rules R_1 and R_2 , R_1 is higher ranked than R_2 if:

- (1) R_1 has higher confidence than R_2
- (2) if the confidences are equal $\text{supp}(R_1)$ must exceed $\text{supp}(R_2)$
- (3) both confidences and support are equal but R_1 has less attributes in left hand side than R_2

With the set of association rules sorted, the goal is to select a subset that will build an efficient and effective classifier. In our approach we attempt to select a high quality subset of rules by selecting those rules that are general and have high confidence. The algorithm for building this set of rules is described below.

Algorithm Pruning the low ranked specific association rules

Input The set of association rules that were found in the association rule mining phase (S)

Output A set of rules used in the classification process

Method:

- (1) sort the rules according to **Definition1**
- (2) foreach rule in the set S do {
- (3) find all those rules that are more specific
- (4) prune those that have lower confidence
- (5) }

The next pruning method employed is to eliminate conflicting rules, rules that for the same characteristics

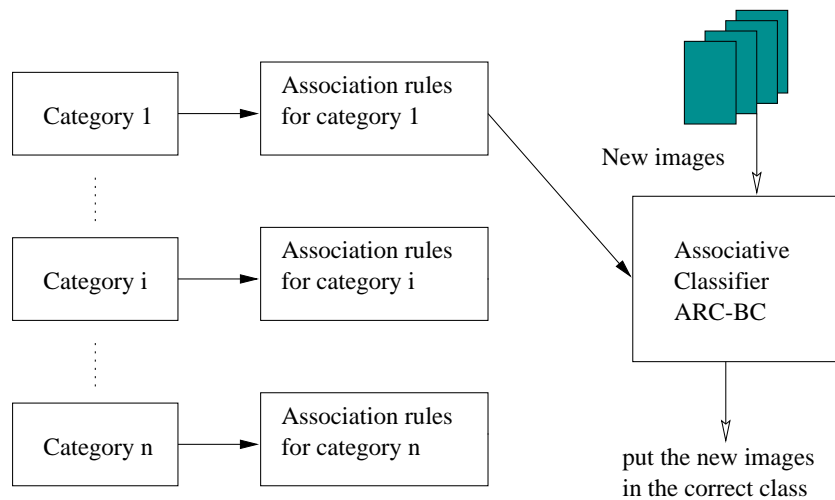


Figure 3. Classifier per category

would point to different categories. For example, given two rules $T_1 \Rightarrow C_1$ and $T_1 \Rightarrow C_2$ we say that these are conflicting since they could introduce errors. Since we are interested in a single-class classification, all these duplicates or conflicting rules are eliminated.

The pruning techniques presented above are not specific to this database, but they can be applied in other cases as well such as text documents or other transactional data.

4.2 Classifying a new image

The set of rules that were selected after the pruning phase represent the actual classifier. This categorizer is used to predict to which classes new objects are attached. Given a new image, the classification process searches in this set of rules for finding the class that is the closest to be attached with the object presented for categorization. This subsection discusses the approach for labelling new objects based on the set of association rules that forms the classifier.

A solution for classifying new objects is to attach to the new image the class that has the most rules matching this new image or the class associated with the first rule that applies to the new object.

Given an object to classify, the features discussed in Section 2 are extracted. The features in the object would yield a list of applicable rules in the limit given by the confidence threshold. If the applicable rules are grouped by category in their consequent part and the groups are ordered by the sum of rules' confidences, the ordered groups would indicate the most significant category that should be attached to the object to be classified.

The next algorithm describes the classification of a new image.

Algorithm Classification of a new image (I)

Input A new image to be classified; The associative classifier (ARC); The confidence threshold conf.t ;

Output Category attached to the new image

Method:

- (1) Foreach rule R in ARC(the sorted set of rules) do {
- (2) if R matches I then R.count++ and keep R;
- (3) if R.count==1 then first.conf=R.conf;
- (4) else if (R.conf>first.conf-conf.t)
- (5) R.count++ and keep R;
- (6) else exit;
- (7) }
- (8) Let S be the set of rules that match I
- (9) Divide S in subsets by category: S_1, S_2, \dots, S_n
- (10) Foreach subset S_1, S_2, \dots, S_n do {
- (11) Sum the confidences of rules in S_k
- (12) Put the new document in the class
- that has the highest confidence sum
- (13) }

5. Experimental Results

This section introduces the data collection that we used and the experimental results obtained using the new classification method.

5.1 Mammography Collection

The data collection used in our experiments was taken from the Mammographic Image Analysis Society (MIAS) [12]. Its corpus consists of 322 images, which belong to three categories: normal, benign and malign. There are 208 normal images, 63 benign and 51 malign, which are considered abnormal. In addition, the abnormal cases are further divided into six categories: microcalcification, circumscribed masses, spiculated masses, ill-defined masses, architectural distortion and asymmetry. All the images also include the locations of any abnormalities that may be present. The existing data in the collection consists of

the location of the abnormality (like the centre of a circle surrounding the tumor), its radius, breast position (left or right), type of breast tissues (fatty, fatty-glandular and dense) and tumor type if it exists (benign or malign). All the mammograms are medio-lateral oblique view. We selected this dataset because it is freely available, and to be able to compare our method with other published work since it is a commonly used database for mammography categorization.

5.2 Experimental Results

We have tested our classification approach with ten different splits of the dataset. For Table 1 that is presented below, the association rules are discovered setting a starting minimum support at 25% and the minimum confidence at 50%. The computation of the actual support with which the database is mined is computed in an adaptive way. Starting with the given minimum support the dataset is mined, then a set of association rules is found. These rules are ordered and used as a classifier to test the classifier on the training set. When the accuracy on the training set is higher than a given accuracy threshold, the mining process is stopped, otherwise the support is decreased ($\sigma = \sigma - 1$) and the process is continued. As a result, different classes are mined at different supports. The parameters in the tests with the results below are: minimum support 25%, minimum confidence 50% and the accuracy threshold is 95%. In the tests that we run the support varied down to 8% for some of the classes in the 10 splits. The abnormal data sets were mined at lower supports than the normal ones. That was due to the unbalanced data set, where the abnormal cases were in a lower number than the normal ones.

Classification in the first two columns of Table 1 is done by assigning the image to the category attached to the first rule (the one with the highest confidence) that applies to the test image (see Table 1 columns under '1st rule'). However, pruning techniques are employed before so that a high quality set of rules is selected. The pruning technique used in this case is a modified version of the database coverage (i.e. selecting a set of rules that classifies most transactions presented in the training set). Given a set of rules, the main idea is to find the best rules that would make a good distinction between the classes. The given set of rules is ordered. Take one rule at a time and classify the training set for each class. If the consequent of the rule indicates class c_i keep that rule, only if it correctly classifies some objects in c_i training set and doesn't classify any in the other classes. The transactions that were classified are removed from the training set.

The next columns in Table 1 are results of classification that uses the most powerful class in the set of rules. The difference is as follows: in the first two columns the set of rules that form the classifier is the set of rules extracted at the mining stage but ordered according to the confidence and support of the rules (support was normalized so that the ordering is possible even if the association rules are

found by category)(see Table 1 columns under 'ordered'); in the next two columns after the rules were ordered the conflicting rules (see Section 4.1) were removed (see Table 1 columns under 'cut rules'); in the last two columns (see Table 1 columns under 'remove specific') from the ordered set of rules the specific ones were removed if they had lower confidence (see Section 4.1).

We also present precision/recall graphs in Figure 4 to show that both false positive and false negative are very small for normal cases, which means that for abnormal images was a very small number of false negative which is very desirable in medical image classification.

The formulas for precision and recall are given below:

$$R = \frac{TP}{TP + FN} \quad (4)$$

$$P = \frac{TP}{TP + FP} \quad (5)$$

The terms used to express precision and recall are given in the contingency table Table 2, where TP stands for true positives, FP for false positives, FN for false negatives and TN for true negatives.

From the graphs presented in Figure 4 one can observe that for both precision and recall for normal cases the values are very high. In addition, we can notice from equations 4 and 5 that the values for FP and FN tend to zero when precision and recall tend to 100%. Thus, the false positives and in particular false negatives are almost null with our approach.

In Table 3 the classification is done using the association rules obtained when mining the entire dataset at once as in [1]. However, the transactional database was organized as explained in Section 2. In the first two columns the set of rules that form the classifier is the set of rules extracted at the mining stage but ordered according to the confidence and support of the rules (see Table 3 columns under 'ordered'); in the next two columns after the rules were ordered the conflicting rules (see Section 4.1) were removed (see Table 3 columns under 'cut rules').

As observed from the two tables presented above, the accuracy reached when ARC-BC is used is higher than the one obtained when the training set was mined at once with ARC-AC. However, the accuracy reached in [1] with ARC-AC was actually higher than in this case (69.11%). These results prove the importance of choosing the right data cleaning technique and data organization in reaching an effective and efficient data mining system.

Not only in accuracy does ARC-BC outperform ARC-AC, but in time measurements as well (41.315 seconds versus 199.325 seconds for training and testing for all ten splits). All tests were performed on an AMD Athlon 1.8 GHz.

Split	1st rule		ordered		cut rules		remove specific	
	#rules	accuracy	#rules	accuracy	#rules	accuracy	#rules	accuracy
1	22	76.67	1121	80.00	856	76.67	51	60.00
2	18	86.67	974	93.33	755	90.00	48	86.67
3	22	83.33	823	86.67	656	86.67	50	76.67
4	22	63.33	1101	76.67	842	66.67	51	53.33
5	33	56.67	1893	70.00	1235	70.00	63	50.00
6	16	66.67	1180	76.67	958	73.33	51	63.33
7	30	66.67	1372	83.33	1055	73.33	58	53.33
8	26	66.67	1386	76.67	1089	80.00	57	46.67
9	20	66.67	1353	76.67	1130	76.67	52	60.00
10	18	76.67	895	83.33	702	80.00	51	76.67
avg(%)	22.7	71.02	1209.8	80.33	927.8	77.33	53.2	62.67

Table 1. Classification accuracy over the 10 splits using ARC-BC

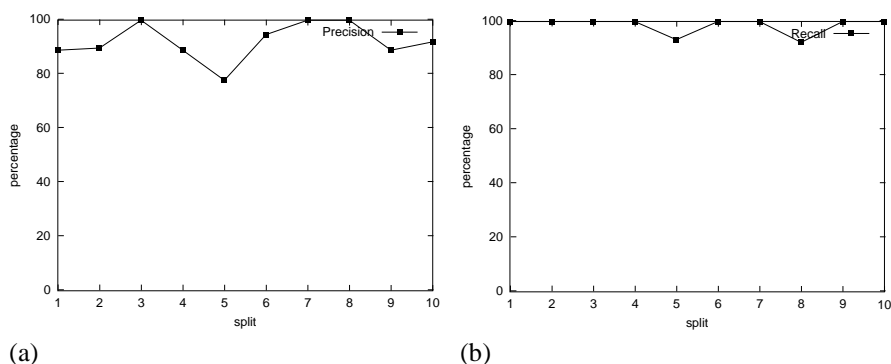


Figure 4. (a) Precision over the ten splits ; (b) Recall over the ten splits;

6. Conclusions

In this paper we proposed a new classification method applied to medical image classification. The novelty comes with the system proposed where the cleaning phase is new and prove to match well with the classification system proposed. The evaluation of the system was carried out on MIAS [12] dataset and the experimental results show that the accuracy of the system reaches 80.33% accuracy and the false negatives and false positives tend towards zero in more than half the splits.

References

- [1] Maria-Luiza Antonie, Osmar R. Zaiane, and Alexandru Coman. Application of data mining techniques for medical image classification. In *In Proc. of Second Intl. Workshop on Multimedia Data Mining (MDM/KDD'2001) in conjunction with Seventh ACM SIGKDD*, pages 94–101, San Francisco, USA, 2001.
- [2] D. Brazokovic and M. Neskovic. Mammogram screening using multiresolution-based image segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(6):1437–1460, 1993.
- [3] C. Chen and G. Lee. Image segmentation using multiresolution wavelet analysis and expectation-maximization (em) algorithm for digital mammography. *International Journal of Imaging Systems and Technology*, 8(5):491–504, 1997.
- [4] A. Dhawan et al. Radial-basis-function-based classification of mammographic microcalcifications using texture features. In *Proc. of the 17th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 535–536, 1995.
- [5] H. Chan et al. Computerized analysis of mammographic microcalcifications in morphological and feature spaces. *Medical Physics*, 25(10):2007–2019, 1998.
- [6] H. Li et al. Markov random field for tumor detection in digital mammography. *IEEE Trans. Medical Imaging*, 14(3):565–576, 1995.
- [7] H. Li et al. Fractal modeling and segmentation for the enhancement of microcalcifications in digital mam-

Category <i>cat</i>		human assignments	
		Yes	No
classifier assignments	Yes	TP	FP
	No	FN	TN

Table 2. Contingency table for category *cat*

Split	ordered		cut rules	
	#rules	accuracy	#rules	accuracy
1	6967	53.33	6090	53.33
2	5633	86.67	4772	86.67
3	5223	76.67	4379	76.67
4	6882	53.33	5938	53.33
5	7783	50.00	6878	50.00
6	7779	60.00	6889	60.00
7	7120	46.67	6209	46.67
8	7241	43.33	6364	43.33
9	7870	53.33	6969	53.33
10	5806	76.67	4980	76.67
avg(%)	6830.4	60.00	5946.8	60.00

Table 3. Classification accuracy over the 10 splits using ARC-AC[1]

mograms. *IEEE Trans. Medical Imaging*, 16(6):785–798, 1997.

- [8] I. Christoyianni et al. Fast detection of masses in computer-aided mammography. *IEEE Signal Processing Magazine*, pages 54–64, 2000.
- [9] Rafael C. Gonzalez and Richard. E. Woods. *Digital Image Processing*. Addison-Wesley, 1993. second edition.
- [10] Jiawei Han and Micheline Kamber. *Data Mining, Concepts and Techniques*. Morgan Kaufmann, 2001.
- [11] S. Lai, X. Li, and W. Bischof. On techniques for detecting circumscribed masses in mammograms. *IEEE Trans. Medical Imaging*, pages 377–386, 1989.
- [12] <http://www.wiau.man.ac.uk/services/MIAS/MIASweb.html>.
- [13] T. Wang and N. Karayiannis. Detection of microcalcification in digital mammograms using wavelets. *IEEE Trans. Medical Imaging*, pages 498–509, 1998.
- [14] Osmar R. Zaiane and Maria-Luiza Antonie. Classifying text documents by associating terms with text categories. In *In Proc. of the Thirteenth Australasian Database Conference (ADC'02)*, pages 215–222, Melbourne, Australia, 2002.

An Application of Data Mining in Detection of Myocardial Ischemia utilizing pre- and post-Stress Echo Images

PRAMOD K. SINGH

Faculty of Information Technology
University of Technology, Sydney
PO Box 123, Broadway,
NSW 2007, Australia
Email: pksingh@it.uts.edu.au

SIMEON J. SIMOFF

Faculty of Information Technology
University of Technology, Sydney
PO Box 123, Broadway,
NSW 2007, Australia
Email: simeon@it.uts.edu.au

DAVID D. FENG

School of Information
Technologies,
University of Sydney,
NSW 2006, Australia
Email: feng@it.usyd.edu.au

Abstract:

Automatic identification of endocardial and epicardial boundaries of LV has been a focus of research attention in the development of computational methods and computer support for cardiologists in identifying clinical heart disease and their diagnosis. Among heart imaging techniques, echocardiography offers significant advantages because of its low cost, portability, minimal discomfort, the absence of ionizing radiation, and its possible application for patient monitoring through real time processing. However, images generated from echocardiogram data are of poor quality. This paper presents the initial work in the development of a data mining approach for computer-assisted detection of myocardial ischemia, which includes Left Ventricle (LV) wall boundary identification, segmentation and further comparative analysis of wall segments in pre- and post stress echocardiograms.

Keywords: Echocardiograms, Image processing, Multimedia Data mining, Object identification, Ischemia

1. Introduction

The main objective of many efforts in cardiac imaging and image analysis is to access the regional function of the Left Ventricle (LV) of the heart. The general consensus is that the analysis of heart wall deformation provides quantitative estimates of the location and extent of Ischemic Myocardial Injury (IMI) [10]. Regional LV deformation can be determined using all of the principal imaging modalities, including contrast angiography, echocardiography, radio nuclide imaging, cine computed tomography (CT) and magnetic resonance (MR) imaging. Automatic identification of endocardial and epicardial boundaries of LV has been a focus of research attention in the development of computational methods and computer support for cardiologists in identifying clinical heart disease and their diagnosis.

Echocardiography offers significant advantages over all other imaging techniques. The technique is attractive

because of its low cost, portability, minimal discomfort, the absence of ionizing radiation, and its possible application for patient monitoring through real time processing [6, 11]. From a data mining point of view, data collected by echocardiograph systems includes sequence data of the heart behaviour.

Myocardial ischemia is a heart disease induced by the obstruction of one or more coronary artery. LV is affected accordingly, which present the change of contractibility of certain segments of LV in echocardiograms images but very rarely on the whole ventricle. The abnormalities can be detected by detailed examination of the dynamics of each segment of LV walls and the coordination between them.

Echocardiography is versatile; it may be combined with exercise, pharmacological, and other stressors and used in availability of circumstances less favorable to other techniques. The stress echocardiography provides a means of identifying myocardial ischemia by detection of stress-induced wall motion abnormalities by comparison of pre- and post stress images. The accuracy of stress echo cardiology in detecting significant coronary stenoses has proved to be from 80% to 90% depending on the population studies [11]. The technological revolution of ultrasound and digital technology brought this modality from a research to a clinical tool, but the interpretation of these studies remains still on subjective observation.

From data mining point of view the echo data can be viewed as video data, which consists of a sequence of echo images, synchronized by the ECG signal. The basic requirement of quantitative analysis of echo images is the complete determination of inner (endocardial) and outer (epicardial) boundaries of the LV wall. In computer vision terms the finding of LV wall boundaries in echo images is an object detection problem. An object detection process typically involves image-processing algorithms for information extraction from images and further analysis of extracted information using priori knowledge of problem domain. A typical configuration of LV wall detection system is shown in Figure 1 [3]:

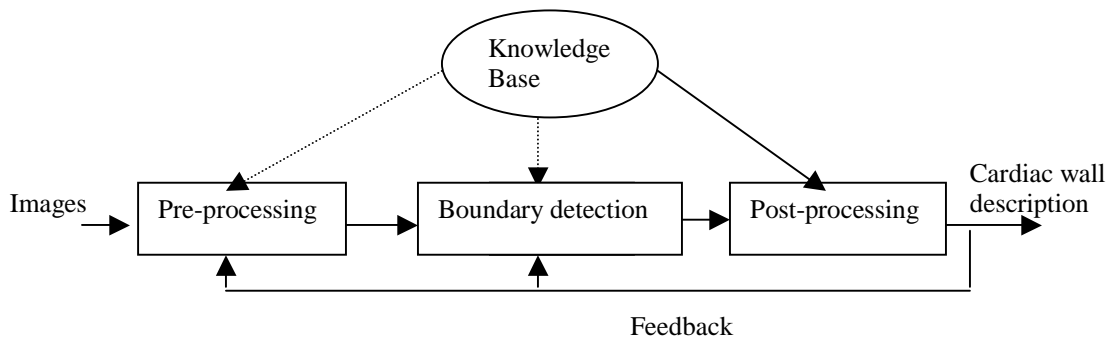


Figure 1. Typical configuration of an LV wall detection system

Algorithms that detect spatial features such as intensity edges [7] and those that detect temporal events such as image motion can provide information for the extraction of LV wall boundaries. Attributes of detected features and events are also useful in interpretation processes. A control strategy manipulates the output from the image processing algorithms to determine the boundary location. An example of the operation taken by the control strategy is the classification of each detected image edge segment as either part of the inner LV wall (endocardial boundary), part of the papillary muscle, part of outer LV wall (epicardial boundary), or an artifact due to noise.

Further, the paper discusses the background of the assessment of regional wall motion abnormalities, the data preprocessing and analysis techniques, the interpretation of the output and further work in the project

2. Assessment of Regional Wall Motion Abnormalities

The American Society of Echocardiography has recommended the use of 16 segment model of LV for assessment of wall motion abnormalities and grading the severity of segmental dysfunction of LV. In 16 segments model, LV is divided into three levels that are further subdivided to produce a total of 16 segments [2]. The three levels such as basal, mid and apical of LV are divided into three equal lengths using the papillary muscles as anatomical landmarks, as shown in Figure 2. The basal and mid levels are divided into six equal segments while the apical level is divided into four equal segments, as shown in Figure 3. The three levels of LV can be captured using parasternal short axis views of the LV in 2-dimensional echocardiography.

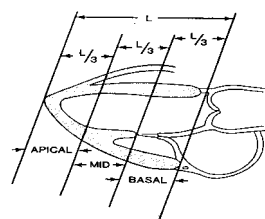


Figure 2. Division of Left Ventricle into Basal, Mid and Apical levels

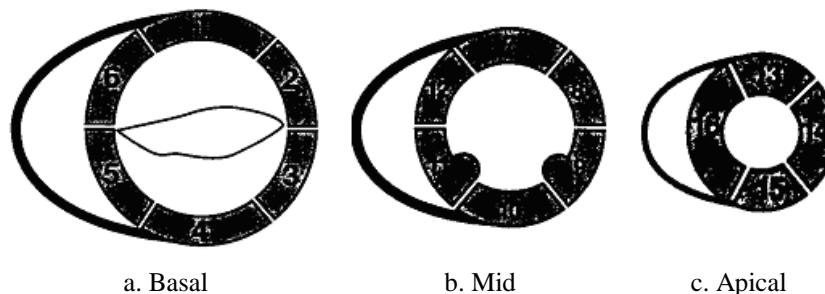


Figure 3. Parasternal Short Axis views at Basal, Mid and Apical levels

Recognition of the coronary blood supply to each individual segment of the 16 segment left ventricle aids in the identification of myocardial ischemia. Each myocardial segment can be classified by three coronary artery distributions (anterior, inferior and lateral). The obstruction of one or more coronary artery presents the change of contractibility of certain segments of LV in echocardiography images. The contractility of a segment can be correlated with the level and severity of obstruction or narrowing of relevant coronary artery. Coronary artery distribution to the 16 segment model of the LV is given in Table 1 [2]:

The normal response of the LV to stress is a uniform increase of regional wall motion, thickening and a reduction in end-systolic LV cavity size, with minimal changes in diastolic size[10]. The distinction between resting and stress induced regional wall motion abnormalities fundamentally differentiates prior myocardial infarction (MI), identified by resting akinesis (systolic increase in free wall thickness is less than normal) or dyskinesis (outward movement of wall during systole with associated systolic wall thinning) from induced ischemia, characterized by either new or worsening wall motion abnormalities.

Level	Segment No.	Segment Name	Coronary arteries and Branches
BASAL	1	Anterior	LAD
	2	Anterolateral	LAD
	3	Inferolateral	CF or OM
	4	Inferior	RC or RM
	5	Inferoseptal	RC or RM
	6	Anteroseptal	LAD
MID	7	Anterior	LAD
	8	Anterolateral	LAD
	9	Inferolateral	CF or OM
	10	Inferior	RC or RM
	11	Inferoseptal	RC or RM
	12	Anteroseptal	LAD
APICAL	13	Anterior	LAD
	14	Lateral	LAD
	15	Inferior	LAD
	16	Septal	LAD

Table 1. 16 Segment Model of LV and Coronary Artery supply to each segment.

Where LAD = left anterior descending; CF = circumflex; OM = obtuse marginal; RC = right coronary and RM = right marginal.

4. Data analysis technique

Detection of myocardial ischemia is mainly based on the quantitative analysis of the thickness of ventricle's walls in different stages of the heart cycle. The process of detection can be split into two parts – the identification of the wall boundaries, their approximation and segmentation; and the estimation of quantitative indicators based on dynamic behaviour of the segments of the LV wall in different stages of the heart cycle.

The quantitative analysis of pre- and post stress sequences of echo images are based on the identification of the complete inner (endocardial) and outer (epicardial) boundaries of the LV wall. The poor quality of the images, due to intrinsic limitation of echo imaging such as speckle noise, image drop outs, boundary discontinuity, and disturbances in the images by valves, papillary muscles, etc., makes difficult the automatic boundary

identification in echocardiograms. High noise levels are also present due to other artefacts like translation and rotation of imaging object. These noisy effects plaguing 2D data raise real troubles to any computer based feature extraction [3]. Some of the major problems are illustrated in Figure 4. As a result of the clustering threshold a typical boundary detection algorithm will produce, in the context of ventricle wall identification a number of regions that need further steps for identification and approximation of the wall boundaries:

- Closed contours on the ventricle wall – such regions require aggregation into a larger cluster
- Closed contours inside the ventricle – for the analysis of such regions do not belong to the wall in consideration and have to be filtered
- Parts of the wall that are not detected, i.e. contours that include part of the wall as an internal part of the cluster

- Parts of the wall that are identified as boundaries of the ventricle, but are not separated from the rest.

As illustrated in Figure 4, due to the limitations of current echo imaging technology the straight forward application of bitmap clustering and contour detection algorithms

may identify only parts of the ventricle wall. Hence, the proposed object extraction technique in echocardiogram images includes the following stages:

- Image data pre-processing and cleaning
- Contour detection and segment computation

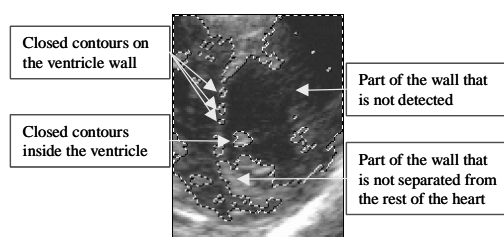


Figure 4. Issues in the identification of ventricle boundaries in echocardiographic images

Image pre-processing

Echo images have very poor signal-to-noise ratio because of the above-mentioned limitations of echo imaging. Pre-processing is required to reduce noise level and to make homogeneous regions uniform. Image pre-processing includes adjusting of colour (in the case of echo images - grey-scale) balances and tonal corrections by adjusting the values of the highlight and shadow pixels in the image, setting an overall tonal range that allows for the sharpest detail possible throughout the image (in extreme

cases this can be a black/white separation with respect to a particular threshold, as illustrated in Figure 5, where the threshold for the clusters is computed on the basis of the grey values of the pixels in the corresponding cluster).

There are several implementations of filters but mathematical morphology [7] using opening and closing concepts proved to be more effective technique for emphasizing the epicardial and endocardial boundaries of LV walls in end systolic and end diastolic frames of pre- and post stress echocardiograms.

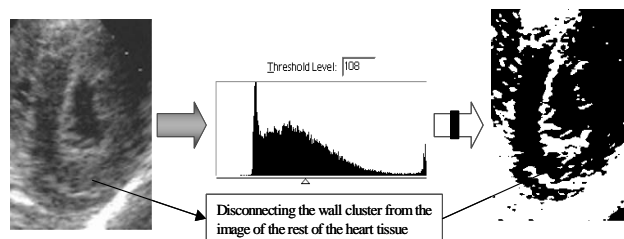


Figure 5. Example of simple image pre-processing step that facilitates the contour detection.

After filtering, the first step is to find the coordinate centre in interior of the cavity where wall contours are being searched. Further the images are converted from Cartesian coordinate system to polar coordinates. Once an image is converted to polar form the so called distance function is found, by defining some special characteristic (first maximum, maximum value, etc.) for each radius and drawing the resulting function [9]. A different distance function is evaluated for each contour. The starting function for inner contour (endocardium) where maximum value of each radius has been used to define the distance function. The goal of the algorithm is to find the

best possible functions for both the inner and outer contours from these starting distance functions.

Contour Detection and Segment Computation

Several approaches for detection of LV boundaries in 2D echocardiographs have been reported such as optical flow[9], snakes[4], simulated annealing[5], dynamic programming[8] and possibly others, but unfortunately none of them are effectively applicable to real application

due to their respective inherent complexity and applicability problems. Nevertheless these techniques in echocardiogram images suffer mainly from usual poor quality of images. Also they are computationally intensive [6].

The algorithm used in this paper combines the detection of endocardial and epicardial boundaries, and the computation of the area of a segment of LV wall. It is based on a modified form of two-phase relaxation active contour detection technique [1]. The algorithm for detection of contours and computation of area of segmental wall of LV has the following steps:

1. Detection of initial points on epicardial and endocardial boundaries in the image using two different threshold values.
2. Closing the contour using active contours.
3. Dividing the area covered under epicardial and endocardial boundaries in to equal six or four segments depending on the level of image view (e.g. six segments in basal level image of LV).
4. Computation of pixels covered in one segment.

As a result of this algorithm we can approximate the area of a segment of LV wall, which can be further used for 2D or 3D modelling of the LV.

Object analysis, evaluation of the LV condition and interpretation of results

The area value of a segment in an end systolic image and in an end diastolic image of pre- and post stress (peak) echocardiograms are most important for monitoring LV wall motion. The effective change of LV wall from rest to stress echo is uniform at all segments. These measurements have obvious medical importance in detection of ischemic effect of heart. The detection algorithm has been explained as follows:

Let Contractility of segments of LV wall be $C = \{ES, ED, S_n, A_{sn}, A_{dn}\}$, where ES indicates an "End Systolic" image; ED indicates an "End Diastolic" image; S_n is number of segments of epicardial boundary (either 4 or 6); A_{sn} is the area covered between the epicardial and endocardial boundaries in N-th segment in the "End Systolic" image; A_{dn} is area covered between the epicardial and endocardial boundaries in N-th segment in the "End Diastolic" image. C can be expressed as $C = |A_{sn} - A_{dn}|$. Let C_{rn} and C_{on} be the contractility of segment n in pre (r) and post (o) stress images respectively. Then the variance in contractility Δ of segment N is expressed as follows:

$$\Delta_n = C_{on} - C_{rn}$$

If Δ_n is zero then segment N may have ischemic affect. If Δ_n is negative then segment N may have ischemic affect and requires further comparison between Δ_n and Δ_m , where $m \in (S_n - n), m \neq n$ to evaluate the scale of damage of a segment. If Δ_n is positive then segment N may be normal but further Δ_n should be compared with Δ_m , where $m \in (S_n - n), m \neq n$ for confirmation. Even if a segment has shown the positive variance of contractility but the contractility of that is less than the other ones the segment may have affect of ischemia.

Change of contractility of segments in stress echo images in comparison to rest echo images should be uniform. A segment may have variation in contractility with reference to other segments due to abnormalities in the LV [2]. Based on the above ratios the segmental wall motion can be classified as follows:

- normal - if normal motion at rest with normal/increased wall motion after stress;
- akinesis - if there is absence of inward motion;
- dyskinesis - if paradoxical wall motion in systole;
- hypokinesis if marked reduction in endocardial motion.

A test can be considered positive if wall motion is other than normal. The quantitative measurements can be correlated with the severity of myocardial infarction of the LV wall, which may be induced by narrowing or obstructions of connected coronary arteries to the segment.

5. Discussion and future work

The paper presents the initial work in the development of a 'smart cardiographer' to assist cardiologists, based on the analysis of echocardiogram images and video sequences. The wall detection algorithms utilise the video sequence data, when the actual analysis is based on the ratios between the wall contours on a specific images ("End Systolic" and "End Diastolic" images). The proposed algorithm provides scope of quantitative analysis of segmental LV function for more accurate clinical diagnosis and management of ischemic affect of heart. Another important perspective of this study is the evaluation of the role of continuous non-invasive monitoring of arterial blood pressure and restriction.

The work on the 'smart cardiographer' includes also the development of media integration model and visual presentation of the results. The media integration is connected with data modelling for multimedia data. The visual presentation of the results involves the analysis of

human computer interaction issues related to the medical experts in the area.

References

1. Acharya B, Mukherjee J, and Majumdar AK, "Two-phase relaxation approach for extracting contours from noisy echocardiogram images", in Proc. Int'l Conf. Pattern Recog. and Digital Tech. (ICAPRDT 99), pp 144-148, 1999.
2. Anderson B "The Normal Examination and Echocardiographic Measurements", Edition 1, MGA Graphics, 2000.
3. Chu CH and Delp EJ, "Automatic Interpretation of Echocardiograms – A computer vision Approach", IEEE ISCAS, pp 2611-2614 1988.
4. Cohen LD and Cohen I. "Finite element methods for active contour models and balloons for 2D and 3D images", IEEE Transactions on Pattern Analysis and Machine Intelligence, 15, pp 1131-1147, 1993.
5. Friedland N and Adam D. "Automatic ventricular cavity boundary detection from sequential ultrasound images using simulated annealing", IEEE Transactions on Medical Imaging, 8(4), pp 344-353, 1989
6. Giachetti A. "Online analysis of echocardiographic image sequences", Medical Image Analysis, vol 1, pp 1-25, 1996.
7. Klingler JW Jr., Vaughan CL, Fraker TD and Andrews LT, "Segmentation of Echocardiographic Images Using Mathematical Morphology", IEEE Transactions on Biomedical Engineering, Vol35 No 11, November 1988.
8. Maes L, Bijnens B, Suetens P and Van de Werf F. "Automated contour detection of the left ventricle in short axis view in 2D echocardiograms", Machine Vision and Applications, 6(1), pp 1-9, 1993.
9. Mailloux G and AB et. al. "Computer analysis of heart motion from 2-dimensional echocardiograms", IEEE Transactions on Biomedical Engineering, 34(5), pp 356, 1987.
10. Marrwich TH, "Stress Echocardiography", in the book "Comprehensive Cardiovascular Medicine, edited by Eric J. Topol, Lippincott". Lippincott Raven Publication, Philadelphia 1998. pp 1407-1436.
11. Papademetris X, Sinusas AJ, Dione DP and Duncan JS, "Estimation of 3D Left Ventricle Deformation from Echocardiography", Medical Image Analysis, 5(2001) 12-28.
12. Skorton DJ, Collins S, Garcia E, Geiser EA, Hillard W, Koppeo W, Linker D, and Schwartz G, "Digital signal and image processing in Echocardiography," American Heart Journal, 11(6), pp 1266-1283, 1985.
13. Torres L and Gasull A. "Temporal Automatic Edge Detection of Echocardiographic Images", Proceedings of IEEE Conference on Computers in Cardiology 1990, pp 2149-2152.

FROM DATA TO INSIGHT: THE COMMUNITY OF MULTIMEDIA AGENTS

Gang Wei
Accenture Technology Labs
161 N. Clark Street
Chicago, IL 60601
gang.wei@accenture.com

Valery A. Petrushin
Accenture Technology Labs
161 N. Clark Street
Chicago, IL 60601
valery.a.petrushin@accenture.com

Anatole V. Gershman
Accenture Technology Labs
161 N. Clark Street
Chicago, IL 60601
anatole.v.gershman@accenture.com

ABSTRACT

Multimedia Data Mining requires the ability to automatically analyze and understand the content. The Community of Multimedia Agents project (COMMA) is devoted to creating an open environment for developing, testing, learning and prototyping multimedia content analysis and annotation methods. It serves as a medium for researchers to contribute and share their achievements while protecting their proprietary techniques. Each method is represented as an agent that can communicate with the other agents registered in the environment using templates that are based on the Descriptors and Description Schemes in the emerging MPEG-7 standard. This allows agents developed by different organizations to operate and communicate with each other seamlessly regardless of their programming languages and internal architecture. A Development Environment is provided to facilitate the construction of media analysis methods. The tool contains a Workbench using which the user can integrate the agents to build more sophisticated systems, and a Blackboard Browser that visualizes the processing results. It enables researchers to compare the performance of different agents and combine them to build more powerful and robust system prototypes. The COMMA can also serve as a learning environment for researchers and students to acquire and test cutting edge multimedia analysis algorithms. Thus the efficiency of research in this area can be improved by sharing of media agents.

KEYWORDS

Multimedia content analysis; Agent; MPEG-7; XML Schema

1. INTRODUCTION

The extraction of information from multimedia data is of vital importance with the explosive growth of digitized image, audio and video data. It requires the ability to automatically analyze, understand and annotate multimedia content. A large number of approaches have been proposed in this area, ranging from simple measures like color histogram for image, pitch/energy for audio signal, to more sophisticated systems like emotion recognition in audio [1],

and automatic summarization of TV programs [2] and topic detection and tracking using audio transcripts [3]. However, the capability of the current techniques is still far from the requirement of many applications in practice, especially in term of intelligence level and robustness. For example, even the most advanced face recognition algorithms can easily be fooled by a little makeup or environmental changes. Those challenges are calling for the consolidation of the research efforts in this area. We believe that the reliable understanding of multimedia content has to be achieved by the interaction of a number of specialized, effective and relatively primitive modules (agents) that address different aspects of the content. A number of research efforts have been made in this direction, producing encouraging results, such as the TV genre classification based on face and superimposed text detection in [4], and the use of both audio and video information to analyze multimedia content [5]. To enable the cross-organization sharing and integration of agents, three major issues need to be addressed. First, the data format between the agents should be compatible to allow communication with each other. The coming standard Multimedia Content Description Interface (MPEG-7) [6] promises to provide a unified base for multimedia content description for both producers and consumers. Second, agents should not expose the proprietary techniques of the inventors. Finally, a development environment is needed to facilitate the manipulation of the agents and visualization of the processing results.

Agents are defined as active, persistent software components that perceive, reason, act, and communicate [7]. Agent-based approach proved to be very useful in many applications. We found that the concept of agent is highly valuable for multimedia analysis. Most of the multimedia processing systems uses agents (in the above mentioned sense) implicitly or explicitly [8, 9].

2. MOTIVATION

Multimedia content analysis requires expertise in a number of fields such as image and video processing, audio processing, speech recognition, linguistics, information retrieval and knowledge management. The range of

expertise spans from DSP techniques for feature extraction to methods for knowledge representation, integration and inference. Unlikely a researcher or a research laboratory can cover the required range of expertise to develop a multimedia analysis system from scratch. Usually, each lab concentrates on its own research agenda using commercial tools (if available) or borrowing some experimental tools from other researchers to develop a rounded-up multimedia analysis prototype. Borrowing from the others is not easy due to the variety of platforms, programming languages, data exchange formats and unwillingness of companies to disseminate their intellectual property unprotected. A lucky researcher can get a tool that covers a particular task, for example, face detection; an unlucky researcher has to implement a tool by himself. In any case, the researcher will have only one (or two, if any) face detector, in spite of his awareness that two dozens of such tools exist in the world. This scarcity of media analysis tools and difficulty finding them motivated our COMMA project. The project's general objective is to create a virtual community of researchers, who exchange their multimedia analysis tools and test data. The Community's objective is to consolidate efforts and expedite research and education in multimedia analysis. To facilitate exchanging and combining media analysis tools the following requirements are held:

- The Community provides a library of multimedia analysis agents. Any community member can submit and download agents.
- Agents exist in formats that can be directly used as modules to build larger systems, however the proprietary techniques are hidden from the user.
- Copyrights belong to the agents' authors or their organizations.
- The Community is located on the World Wide Web and agents are program-accessible from any Internet-able workstation.
- The Community provides templates for agents' outputs that facilitate communication among agents and allow building hierarchies of agents.
- The Community provides open source tools for creating agents and visualizing their performance. These tools can be freely downloaded from the Community Web site.

Currently we foresee the following stages in developing the COMMA project.

Stage 1. Simple Agents. Agents at this stage perform the tasks assigned by the human users. The objectives is to:

- Develop tools for creating agents and visualizing their work.

- Create the development environment. Users can deploy agents and build more sophisticated high-level systems by connecting them together.
- Develop templates for the communication between agents' based on MPEG-7.
- Accumulate initial "critical mass" of agents.

Now the Accenture Technology Labs have released a first version of the agent development and visualization tools for Windows 2000/XP platform. And we collaborate with several Universities to create an initial library of agents. After this we shall launch the Community's Web site.

The Community at this stage can serve to both researchers and students. A researcher can compare his/her approach to the known approaches presented in the agent library, combine agents to create a high-level agent, and do a rapid prototyping of a system that solves a particular problem. A student can learn about different approaches to solve a problem, get experience in building media analysis algorithms and systems, and learn up-to-date data representation technologies, such as XML and MPEG-7.

Stage 2. Intelligent Agents. Agents will not only be able to act on assigned tasks, but also automatically synthesize by themselves to solve a specified problem. This will require the description of the agent at the knowledge level, and we plan to use techniques such as Resource Description Framework (RDF) as in [10] or the emerging DARPA Agent Markup Language (DAML) as in [11] to represent the ontology of the agents.

Stage 3. Distributed Agents. The further step is to develop formal specifications, interfaces and tools that allow distributed agents to find each other on the Web to communicate and solve a specified problem. At this stage the Community of researchers will be extended to the Community of Multimedia Agents to justify the title of the project. Some research steps have been made in this direction for simple business-oriented agents [12].

3. ARCHITECTURE

Figure 1 shows the architecture of the system. The Community of Multimedia Agents provides the user two components: the Agent Library and the Development Environment. The agent library contains a set of agents, preferably in executable form and an agent description file, which describes the set of agents in XML. The Development Environment is an application for Windows ME/2000/XP platforms. It consists two parts, namely the Workbench and the Blackboard Browser, responsible for the creation of multimedia analysis processes with agents and the visualization of the results, respectively. The user provides the multimedia files to be processed. Three types of media are allowed: still images, audio files, and video files. Each media object is associated with a "Metadata

Sheet” in XML format, which is a directory of the processing results produced by the agents. When an agent is applied to the media file, the Workbench updates the corresponding Metadata Sheet by adding a record. The Blackboard visualizes the results to the user by the interpreting of the Metadata Sheet.

To start using the system a COMMA member should download the Development Environment application and the agents to a local computer. Then the user can build multi-agent media analysis processes in the Workbench by loading media files and connect agents.

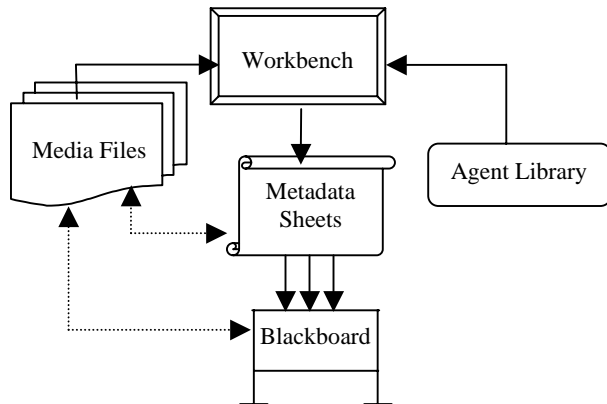


Figure 1. The COMMA Architecture.

4. AGENT LIBRARY

COMMA provides a library of multimedia processing and analysis agents that serve as building modules for more sophisticated, powerful and robust systems. Each agent exists as an individual executable application developed by different researchers and organizations. To enable the agents to communicate and collaborate with each other, we defined the specifications of the agent interface and the XML-based schema for agent description.

4. 1. Agent Interface

The agent interface specification includes two aspects, namely the *syntactic* interface and the *signature* interface. The former addresses the lower-level “physical” characteristics of the agents. The signature interface, in contrast, represents relatively higher-level features of the data to be processed or results that are produced by the agents.

The syntactic interface requires each agent to be an application that can be invoked through a command line, e.g., a console executable program. Any programming language can be used for developing an agent. The system allows also using any interpretive language for agent development, but installation of the interpreting program should be done separately.

Seen at the signature level, an agent in COMMA is a filter that either takes the raw data of the media directly or the

processing results produced by other agents as input, and generates its own processing results that can be used for the possible consumption by other agents. As shown in Figure 2, the signature interface of an agent contains three visible parts, namely Input Pins, Output Pins and Tuners.

An agent must have one or more input pins and output pins for data flow. There are different types of pins depending on the natures of the data. For example, if an agent performs face detection on MPEG video, it has one input pin of type “MPEG” and an output pin of type “Visual Object Information”. Pins of the same type are considered to be compatible with each other. In the Workbench, the user can build multi-agent systems by connecting the input pin of one agent to a compatible output pin of another agent. Thus the agents can collaboratively process the media content by sharing data. We created templates for the data format different pin type based on MPEG-7 standard so that agents with compatible pins can communicate with each other.

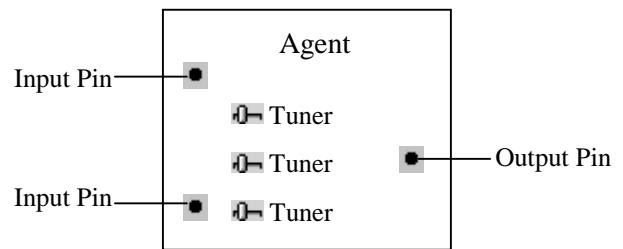


Figure 2. Signature Interface of an Agent

Tuners are used for adjusting technical configurations of agents to give them flexibility. An agent may include zero or more Tuners. Each tuner has a default value recommended by the inventor of the agent to ensure good performance in general cases, while the users can change it to meet their particular needs. For example, when a researcher designs an agent that detects traffic signs on the road for driving assistance, he may prefer to have a balanced recall (the ratio of detected signs among all signs) and precision (the ratio of real signs among all claimed signs), while in practice it is usually desirable to detect as many sign as possible, even though at the cost of producing more false alarms.

4. 2. Agent Description

The executable agents are not self-describing, and thus for the Development Environment to know how to manage them, we defined the XML schema to describe their characteristics, under which the agents are represented in a formalized way understandable not only to human users, but also to the Development Environment.

The organization of the Agent Description Schema is presented in Figure 3. Under the schema, each agent has a

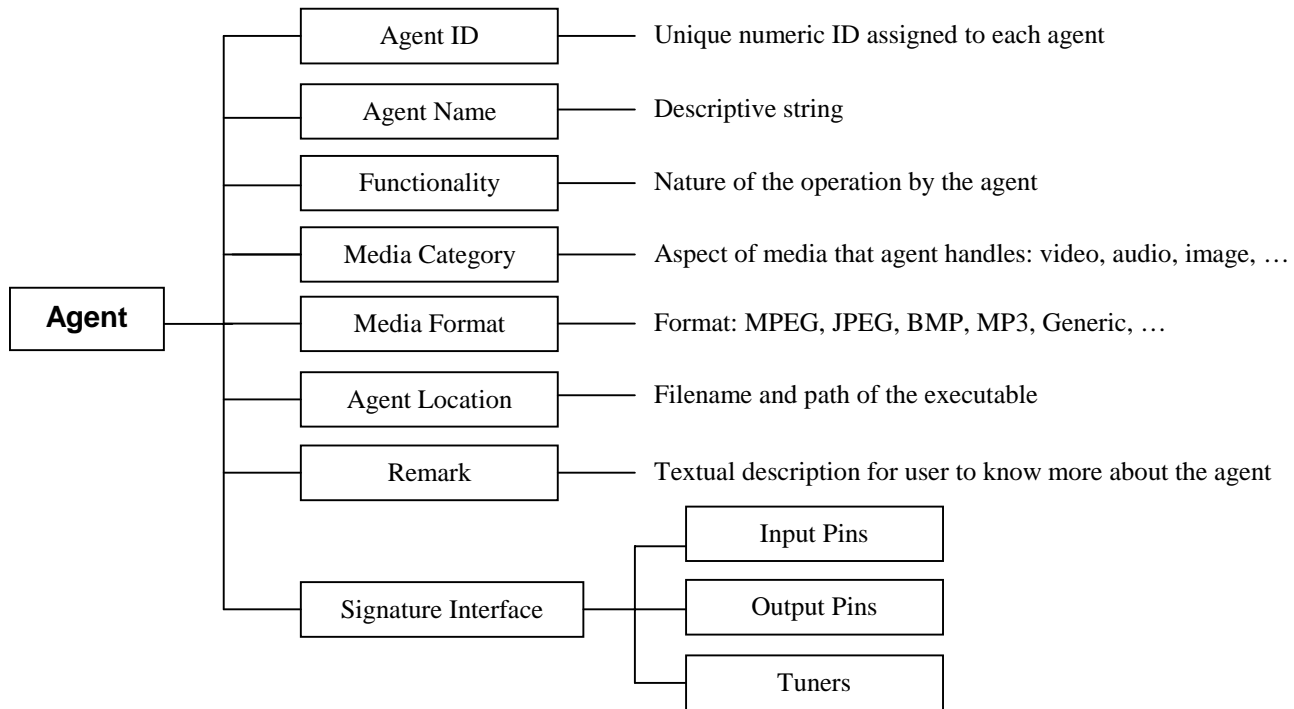


Figure 3. Major Components of the Agent Description Schema

unique numerical ID for retrieval purpose. Other major elements include *Functionality*, *Media Category/Format*, *Agent Location* and *Remark*. The *Functionality* is based on the nature of the operation conducted by the agent, e.g., classification (assign media data into predefined categories), event detection (find certain events in video or audio segments) and object tracking. The *Media Format* attribute indicates what formats of the media files can be processed by the agent, such as MPEG, AVI, BMP, or WAV. The *Media Category*, in contrast, illustrates the general aspect of media the agent deals with, e.g., video, audio or image. For example, consider two agents that both apply to MPEG clip. The first one classifies the camera motion and the second one performs speech recognition. The *Media Category* of the first agent is "video" while that of the second one is "audio". The *Agent Location* is the path and filename of the executable file corresponding to the agent. The *Remark* attribute provides a brief introduction about the agent in plain words to let the user know about the agent in a more natural way. The agent description schema also includes the signature interface, including the input, output pins and the tuners, which has been mention above. Each agent is represented as an XML node in the agent directory. The Development Environment of COMMA contains a GUI tool through which the agent contributor can register new agents by filling out a form. The tool automatically encodes the information provided into the XML description.

5. DEVELOPMENT ENVIRONMENT

The Development Environment provides means for registering media files and agents, and two major tools: a *Workbench* for developing media annotation processes, and a *Blackboard Browser* for visualizing results.

5.1. The Workbench

The Workbench allows a user to select and combine existing agents as building blocks to construct multi-agent systems. The user starts by selecting a media file. The media file is represented as a rectangle with a number of dots at the bottom. The largest dot corresponds to the raw media data. The other smaller dots, if any, are the processing results previously produced by agents. Those results are recorded in the Metadata Sheet for the media file and can be used as inputs to other agents to avoid repeated computation and significantly reducing overhead, especially for time-consuming video processing algorithms. The Workbench filters the agent library and displays only the agents that can process the media. The agents are organized by their functionality in a tree structure in the top-left area as shown in Figure 4. The user can load an agent to the working space by highlighting it and clicking the "Load" button. Each agent is represented as a rectangle with input and output pins displayed as dots at the top and bottom, respectively.

The user can build media annotation processes by connecting the media and agents. Figure 4 gives an example

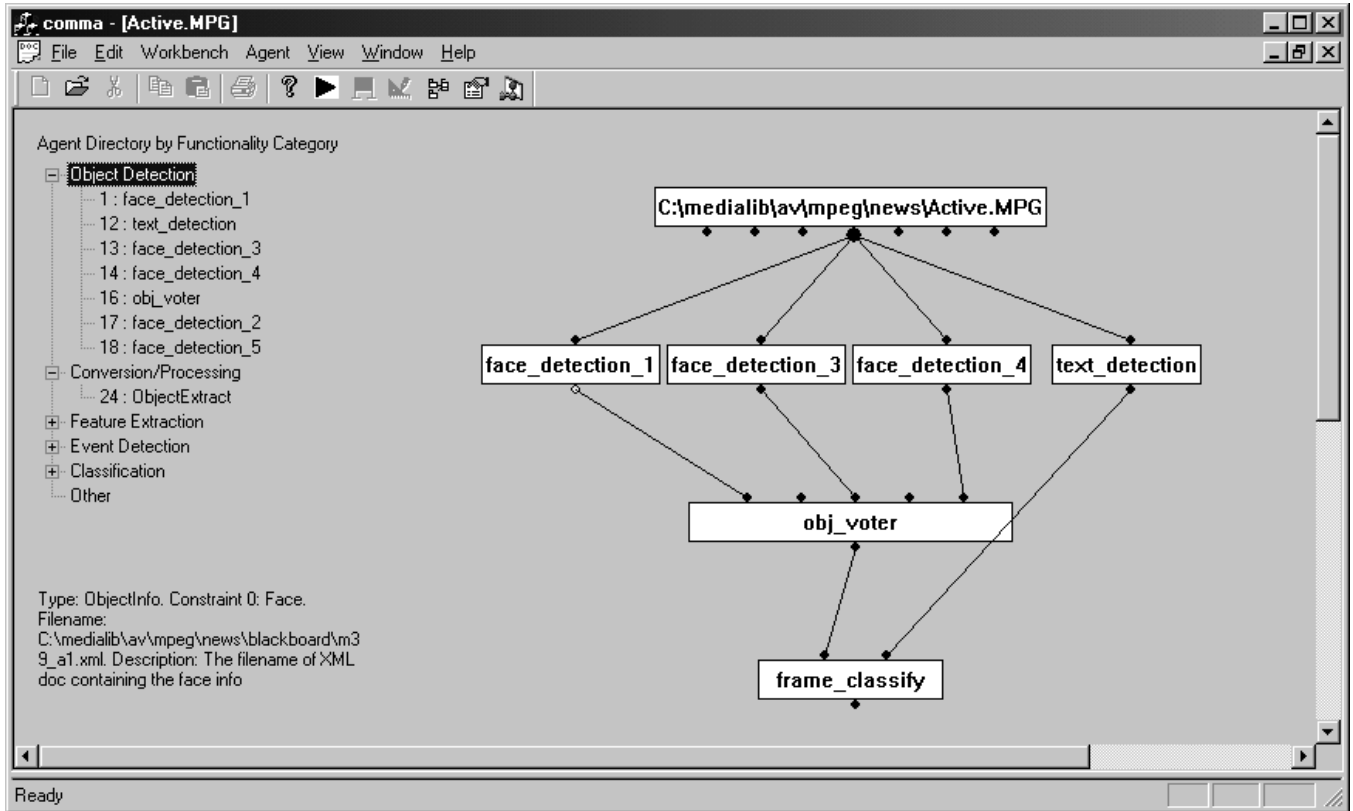


Figure 4. Working in a workbench window.

of integrating agents to build more intelligent and robust system. Consider the scenario where a researcher needs to create an agent that assign the video frame into predefined categories (e.g., “frame with face only”, “frame with text only”, “frame with both text and face”). Without the Community of Multimedia Agent, the researcher may have to re-implement some face and text detection algorithms or creating his own. In the environment of COMMA, he can simply design an agent that takes the results of face and text detection agents as input, and produces classification labels, like the “*frame_classify*” agent in Figure 4. Compared with developing every component from the scratch, a lot of time and efforts can be saved. The user can also save the system composed of agents as a script and later load it as a “macro-agent”.

On the other hand, with the availability of more than one face detection agents, their results can be combined to obtain more reliable performance. Since the face agents may employ various algorithms, e.g., neural network, color-shape analysis, each may have its own strength and weakness at different occasions, and we can expect to improve the overall accuracy by having a voting committee among them. This can be accomplished by the “*obj_voting*” agent in Figure 4, which accepts the results of up to 5 object-detection agents. It has a parameter (tuner) that

specifies the mode of “voting”, which could be “or” (a frame has a face detected if at least one of the agents detects a face), “and” (if all agents detect a face) or “majority” (if the majority of agents detect a face). It has been proved that a voting committee can produce more accurate results than any of its members when the errors of the members are uncorrelated with each other [13]. Therefore with the growth of the agent library, COMMA users are better equipped to address for the complexity of the problem, and we can eventually overcome the challenges in the area of multimedia processing research.

5.2. Blackboard Browser

The Blackboard Browser visualizes the results produced by the agents to provide insight about the media content and let the user have an intuitive evaluation of the performance of the agents. Each agent can generate one or more XML files through its output pins, and the data formats conform to the MPEG-7 based templates associated with the pin types. The location of these result files are recorded in the Metadata Sheet of the media file, and thus the Blackboard Browser can retrieve and visualize them by parsing the Metadata Sheet.

Figure 5 shows a Blackboard window for a video file. It contains video browser on the right side, a current frame

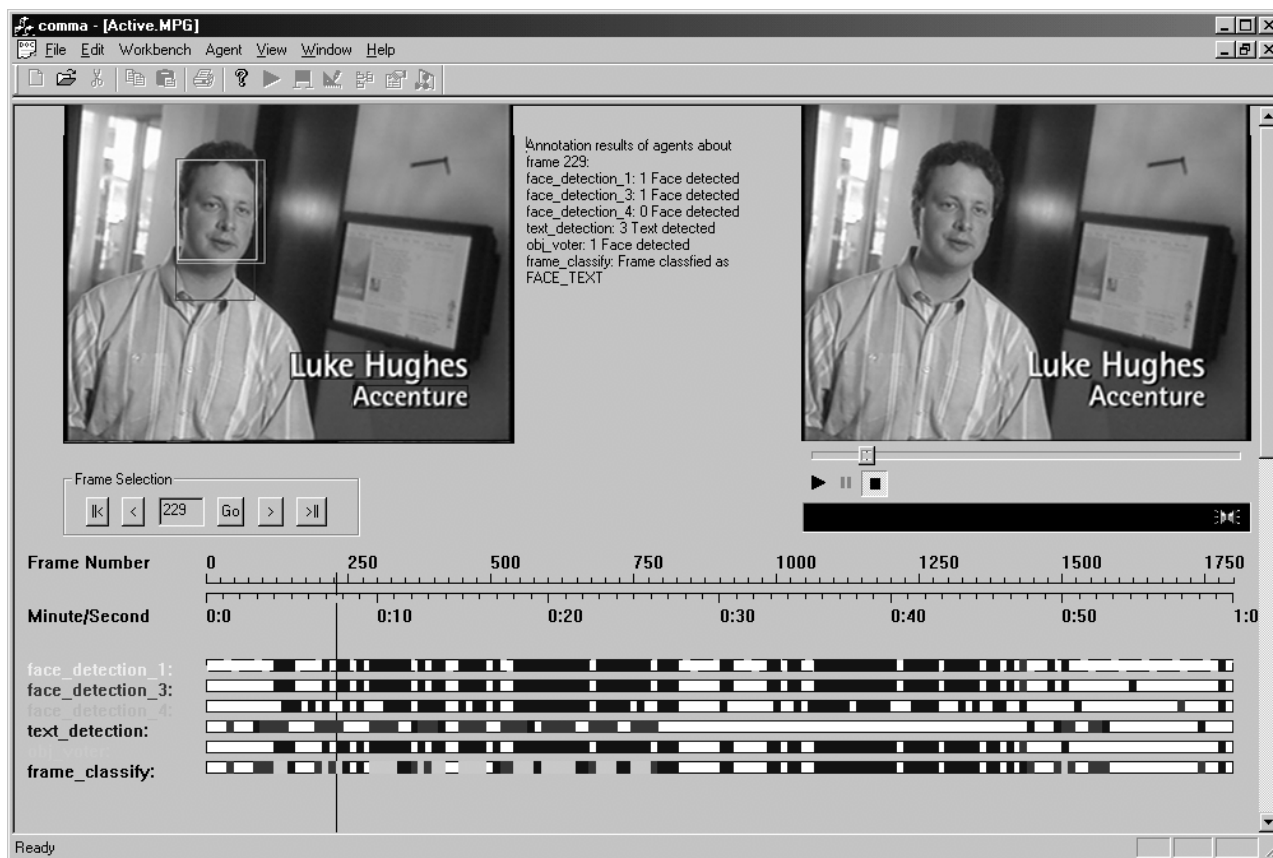


Figure 5. Blackboard Browser window.

image on the left side that presents agents' results, and a summary of agents' findings for the current frame in the middle of the screen. The user can watch the results for any frame using the navigation buttons. Below the frame and time scales are the summaries of agents' findings for the whole clip. For example, the summary of a face detection agent is presented in a form of the categorical color bar. Each frame can be categorized as "no faces detected" (white color), "one face detected" (blue color), and "multiple faces detected" (red color). The same color code is used for the text detection agent's results. A user can explore how a particular detection agent works by clicking on the agent's summarization strip and watch the results represented on the current frame picture as a rectangular that frames a detected face or text. Or by clicking on the time scale the user can watch the results of all agents simultaneously on the same picture.

6. COMMUNITY OF LEARNERS

One of the COMMA project main objectives is to create a community of researcher and students in the multimedia processing problem domain. This social aspect of the project is very important for its success. The environment should encourage people to interact, exchange agents and

ideas, discuss topics of interest, and advertise relevant events, such as workshops, conferences, training sessions that target both academic and business research and development. That is why we are paying a great attention to information that is provided by the COMMA Web site. This information includes related business and academic news, overviews of achievements of lead laboratories and researchers, event and job announcements, book and paper recommendations, tutorials, and glossary of specialized terms. It also includes a directory of community member e-mail addresses and chat rooms for real-time discussions. Altogether the tools and information form a socio-technical learning environment that could be beneficial for researchers, teachers and students.

7. SUMMARY AND FUTURE WORK

The Community of Multimedia Agents is a community of researchers and an open environment that allows researchers to share their achievements in multimedia annotation field while protecting their intellectual property. Our work has three major contributions. First, its agent library of gives researchers access to tools to handle the complexity of multimedia data and absolves them from implementing existing algorithms. Second, the

Development Environment facilitates the development of multimedia analysis methods by enabling the researchers to link agents without concerning about low-level technical issues; it also visualizes the agent result to give the user insight about the media content and agent performance. Third, by improving the accessibility and reusability of multimedia processing agents, the value of each research achievement is maximized.

The future extension of our work will go in three directions. First, we are projecting a change in the interaction mechanism between agents. Presently in COMMA the data flow between agents is one-way, and thus the error made by one agent will propagate to others. A promising solution is to allow agents to confirm or negate the results of each other and reach an “agreement” that is the most consistent to the context [14]. Second, we will introduce intelligence to the agents so that they may not only be assembled by human, but also integrate by themselves to generate a solution to a problem. Third, the agents will be distributed as web services, which will give better control of the agents to the inventors and facilitate their upgrade.

REFERENCES

- [1] V.A. Petrushin. Emotion Recognition in Speech Signal: Experimental Study, Development, and Application, In Proc. 6th International Conference on Spoken Language Processing (ICSLP 2000), Beijing, 2000. Vol. IV, pp 222-228
- [2] M.T. Maybury (Ed.) *Intelligent Multimedia Information Retrieval*, AAAI Press/MIT Press, Menlo Park, CA / Cambridge, MA, 1997.
- [3] O.V. Ibrahimov, I.K. Sethi, and N. Dimitrova. Clustering of Imperfect Transcripts using a Novel Similarity Measure, In Coden A.R., Brown E.W. and Srinivasan S. (Eds.), *Information Retrieval: Techniques for Speech Applications*, LNCS vol. 2273, Springer-Verlag, 2002, pp. 23-35.
- [4] N. Dimitrova, L. Agnihotri, and Gang Wei, Video Classification using Object Tracking, *International Journal of Image and Graphics*. Vol. 1, No. 3 (2001), pp. 487-505.
- [5] Yao Wang, Zhu Liu, and Jin-Cheng Huang, “Multimedia Content Analysis Using both Audio and Video Clues”, *IEEE Signal Processing Magazine*, IEEE Inc., New York, NY, pp. 12-36, vol. 17, No 6, November 2000.
- [6] José M. Martínez, Overview of the MPEG-7 Standard, <http://mpeg.telecomitalia.com/standards/mpeg-7/mpeg-7.htm>
- [7] M.N. Huhns and M.P. Singh, “*Agents and Multiagent Systems: Themes, Approaches, and Challenges*”, In Huhns M.N. and Singh M.P. (Eds.), *Readings in Agents*, Morgan Kaufman, San Francisco, CA, 1998.
- [8] A.J. Hauptmann and M.J. Witbrock, “InforMedia: News-on-Demand Multimedia Information Acquisition and Retrieval”, In [2], pp. 215-239.
- [9] B. Merialdo and F. Dubois, “An Agent-based Architecture for Content-Based Multimedia Browsing”, In [1], pp. 281-294.
- [10] W3C Candidate Recommendation, “Resources Description Framework (RDF) Schema Specification 1.0.”, March 2001
- [11] W3C Notes, “DAML+OIL (March 2001) Reference Description “, March 2001
- [12] J. Heflin and J. Hendler, “A Portrait of the Semantic Web in Action”, *IEEE Intelligent Systems*, vol. 16, No. 2, pp. 54-59, March/April 2001.
- [13] L. K. Hansen and P. Salomon. “Neural network ensembles”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990
- [14] D. Li. “Integrated Multimedia Analysis”. Ph.D. Dissertation. Wayne State University, 2001

A CONTENT BASED VIDEO DESCRIPTION SCHEMA AND DATABASE NAVIGATION TOOL

SADIYE GULER

Northrop Grumman Information Technology / TASC
55 Walkers Brook Road
Reading, Massachusetts 01867
sguler@northropgrumman.com

IAN PUSHEE

Northrop Grumman Information Technology / TASC
55 Walkers Brook Road
Reading, Massachusetts 01867
ipushee@northropgrumman.com

ABSTRACT

In this paper we introduce a unified framework for a comprehensive video description schema and an intuitive browsing and manipulation tool “VideoViews” database navigation tool for video data mining. The description schema and the navigation tool are designed and developed as part of a video analysis and content extraction framework devised under U.S. Government ARDA /VACE project. The proposed description schema is based on the structure and the semantics of the video and incorporates scene, camera, object and behavior information pertaining to a large class of video data. The database navigator, VideoViews is designed to exploit both the hierarchical structure of video data, the clips, shots and objects, as well as the semantic structure, such as scene geometry the object behaviors. VideoViews provides means for intuitive presentation and navigation, interactive manipulation, ability to annotate and correlate the data in the video database. While also supporting conventional database queries this hierarchically and semantically structured browsing tool enables users to freely navigate up and down within the video database to visualize the information and data from a number of perspectives.

KEYWORDS

Video description schema, video database, video data mining, intelligent browsing, video analysis framework.

1. Introduction

Recent advances in digital video technology such as streaming video over IP networks, relatively low cost network cameras and digital video surveillance systems, and wireless video systems are giving rise to a new problem: increasingly larger volumes of video data that has to be browsed, reviewed, qualified and retrieved by video analysts or operators in order to enable decision

making. Hence, methods and tools to assist this process have been of particular interest [10, 7 and references therein].

The promise of content based access for digital video or any other multimedia data type is to enable users to browse, locate, access, interpret, manipulate and analyse the data that is not otherwise reachable by conventional means. This is of great importance, particularly for video data, as only a small percentage of video data collected contains relevant information for a typical user.

Earlier video mining methods are developed as extensions of image mining methods and are based on still image features like color histograms, shape, texture and the spatial composition of the scene, without taking the valuable temporal information into account [9]. Until the middle of last decade “content-based” video representation and browsing research mainly focused on using keyframes for summarizing the temporal information inherent in video and analysing still imagery features of keyframes [1,11,12,13]. One or more key frames that summarize the scene are used for browsing, image similarity assessment and retrieval. The efficiency of such methods depend heavily on how well the keyframes represent the corresponding video segment. These approaches range from simply taking the first frame of each shot as the keyframe [1,12], to detecting visual content changes [13], to analyzing motion characteristics of shots [11]. More recent research focuses on the temporal hierarchal structure of video data based on clips, segments and shots and on the video contents to the level of objects [3,5,10].

The *video description schema* which governs the representation and storage structure of the video data, and *video access (browsing and retrieval)* which deals with locating and accessing the video data, are dual problems and therefore will be best solved using a unified approach. Our approach is based on a video analysis

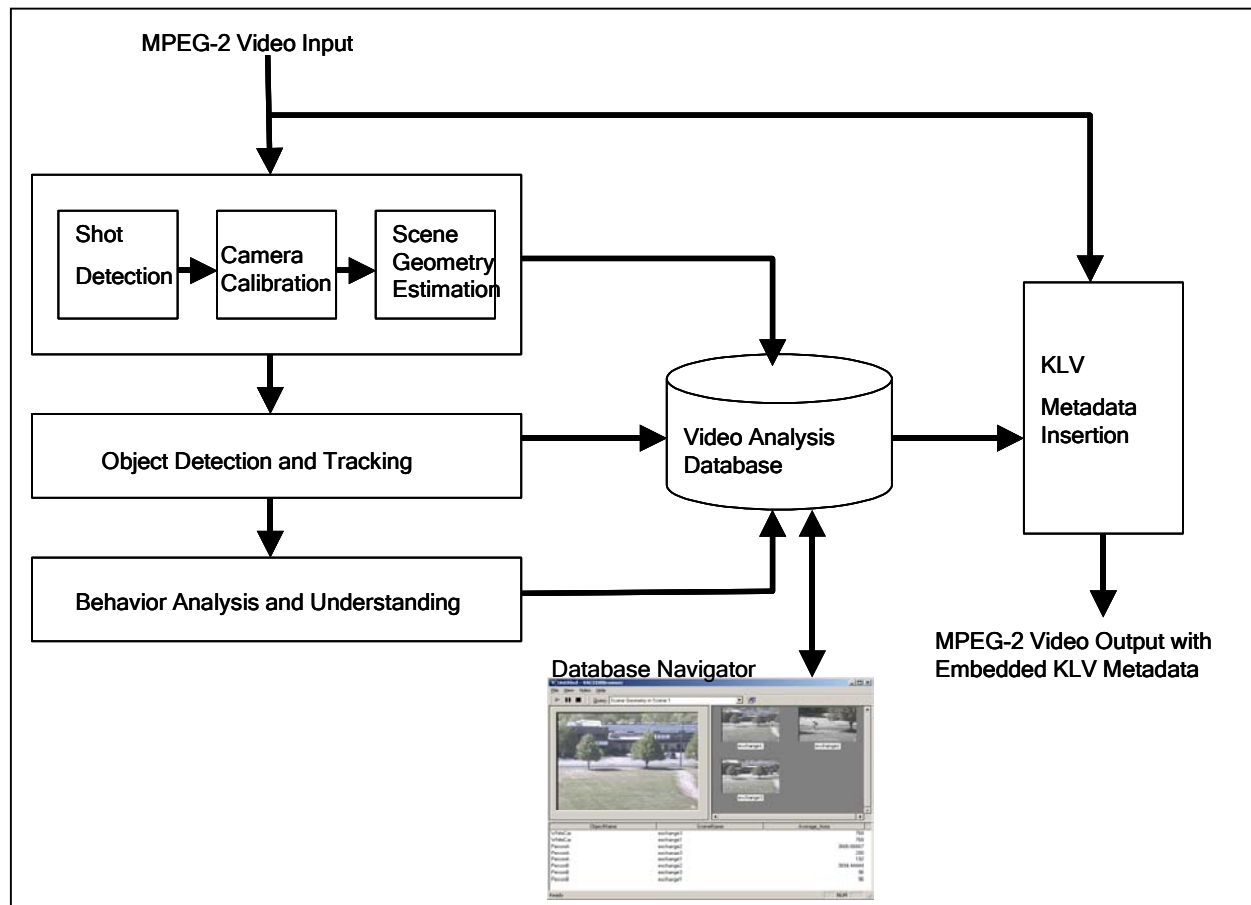
framework for both representation and access components. We exploit the hierarchical and semantic structure of video data as well as the true content, not based on few keyframes, but based on objects' behaviors in time.

The video description schema and the browsing tool "VideoViews" presented here are designed as part of a comprehensive video analysis and content extraction framework developed for U.S. Government ARDA /VACE project. A full discussion of the ARDA/VACE project can be found in [6] and is outside the scope of this paper, however for sake of completeness a high level architectural overview of the video analysis and content extraction framework is described in Section 2. The remainder of this paper is organized as follows: In Section 3, we introduce the video terminology and the video description schema and in Section 4 we describe the database browser VideoViews. Finally some concluding remarks are made and future directions are discussed in Section 5.

2. Video Analysis Framework

A high level architectural diagram of the ARDA VACE project video analysis and content extraction framework is given in Figure 1. In this framework, the video analysis starts with automatic detection of shot-changes, including camera operations such as zoom, pan, tilts and scene cuts. For each new shot, camera calibration is performed using measurements for available parameters and sample image point real world coordinates. Based on the estimated and measured camera parameters, the scene geometry is estimated and used to determine the absolute positions for each detected object. Objects in the video scenes are detected using a combined adaptive background subtraction and edge detection method and tracked over consecutive frames. Objects are detected and tracked in a way to identify the key split and merge behaviors where one object splits into two or more objects and two or more objects merge into one object. These behaviors serve as the key behavior components for several higher-level activities such package drop-off,

Figure 1. High-level overview of the framework



exchange between people, people getting out of cars or forming crowds etc. The discussion above mentioned methods and split and merge based behavior analysis and detection can be found in [6]. In this framework, after the processing the analysis results are stored into the video database using the proposed description schema and can be manipulated using the special database tool VideoViews which will be discussed in detail in Sections 3 and 4 of this paper. The results of the analysis are also encoded as SMPTE KLV metadata and inserted into the video stream in a frame accurate manner, resulting in a self-contained video stream or file that carries its own analysis results. The description of the metadata creation and insertion is outside scope of our discussion of this paper and can be found in [4,8].

3. Video Description Schema

The video description schema is concerned with and should well represent the structure and the semantics of the video data. We propose a schema that matches the inherent structure of video data and describes all aspects of video content and processing results. Before we discuss the details of the description schema we will introduce terminology we adopted for video structure and contents:

Video Asset/Clip/Stream: is a video data file or streaming video data input to be processed, analyzed, interpreted, manipulated and stored in the database which

will be browsed and retrieved.

Video Shot: is a portion of the video clip (a sequence of video frames) produced using a single camera operation such as zoom, pan, tilt or scene cuts with a consistent background. A video clip may contain several shots.

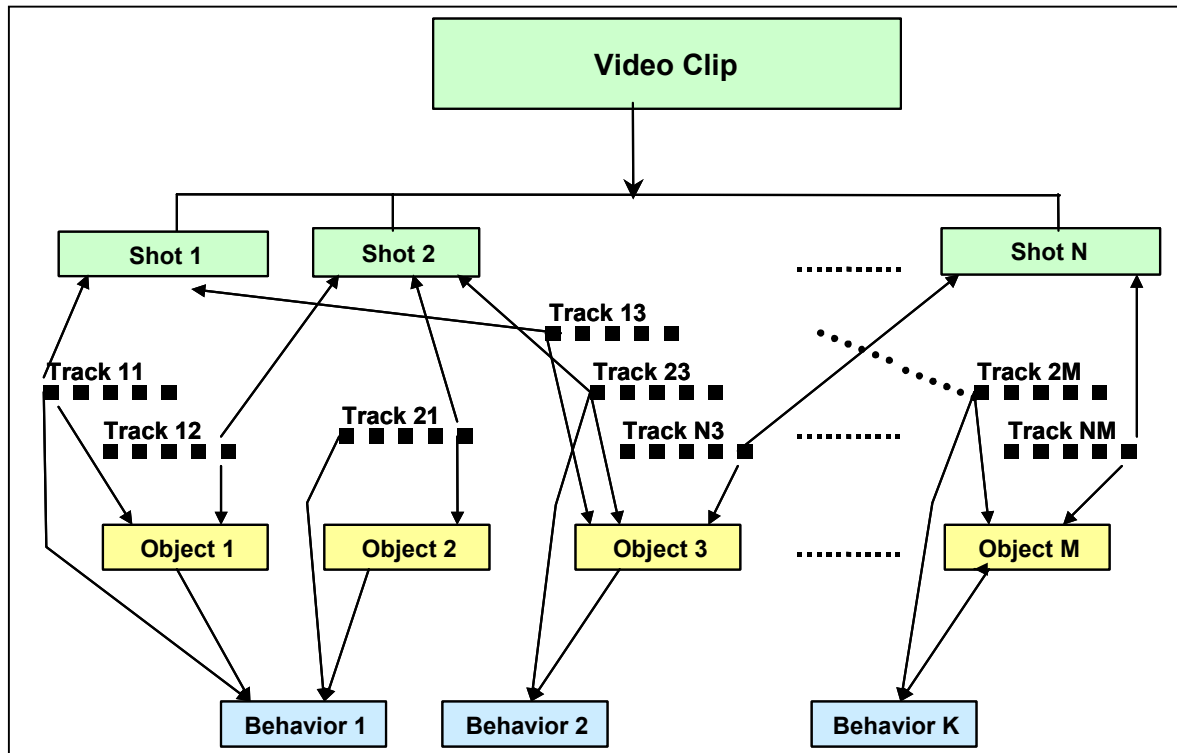
Video Object: is a moving (or stationary) object which is not a part of the shot background. A video shot may contain multiple objects, and conversely each object may appear in multiple video shots and even in multiple video clips.

Object Track: is a collection of coordinates that show the object center positions for detected and tracked objects in each frame. Each uniquely identified object has only one track in a particular video shot. A single point in an object track is called an **Object Track Point**.

Object Behavior: is a semantic interpretation of the actions of detected and tracked objects. Object behaviors represent the semantic story of the video. Each object may exhibit behaviors along each track in a particular video shot and conversely a behavior may be exhibited by several objects in shots of different video clips.

A graphic representation of video clips, shots, objects, their tracks and behaviors are depicted in Figure 2.

Figure 2. Video description hierarchy



These five components introduced above constitute the main elements of the proposed description schema, which is designed to entail all information pertaining to video structure, content and processing results and the relationships and dependencies among all the components.

Each video asset is represented by a name and a description. Video shots are represented by start and end frames within the video clip and identified by the video clip of which they belong to. In addition to these identifying data, each video shot has attributes for camera parameters, such as latitude, longitude and altitude, elevation, azimuth, and tilt angles that maybe available or estimated. Camera parameters are used to aid in post-processing to calculate real world positions, speeds and directions of objects in the video.

Objects (people, animals, cars, etc.) are automatically detected and tracked in a video clip and assigned an ID number by the detection process. Objects are represented by their size and identified type. Since the same object might appear in any number of shots and/or clips, the representation for objects contains only those attributes that remain constant throughout the entire set of video assets, such as the approximate real-world coordinates, and the ‘type’ (person, automobile, package, etc.) of the object. An analyst may also manually supply an object name to ease later understanding of the data.

If a newly detected object is discovered to be the same as a previously detected object on further processing the description schema allows for the detection process to associate an object and its entire track with another object through updating all the track points. Associations can be removed at any time to leave the objects in their pre-associated (separate) states. Recognizing that automatic object detection and tracking is subject to error, the navigation tool allows for users to form new associations between objects and break the automatically generated object associations. After such operations, the new state of object associations are updated and propagated / back-propagated in the database.

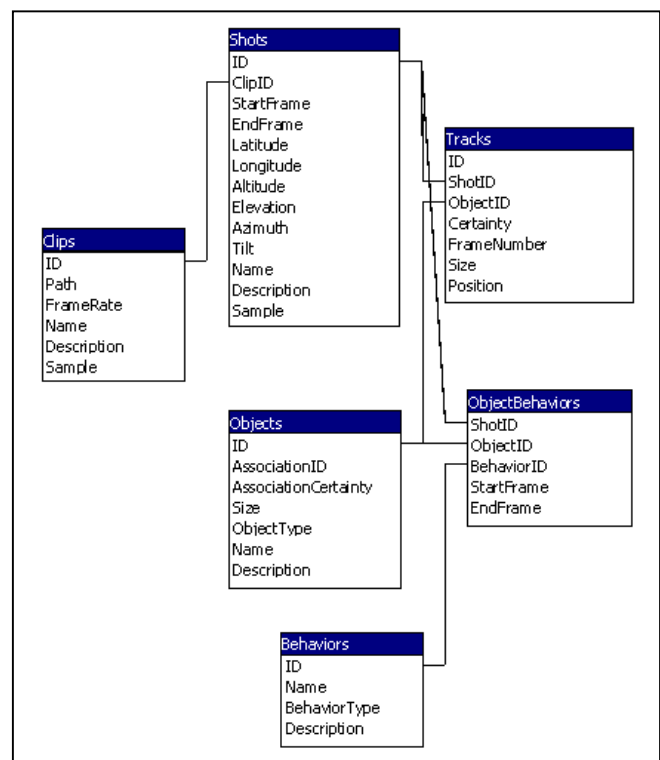
A track point represents an object’s position at that given frame in a particular video shot. Each track constitutes the link between an object and a video shot. The attributes for the track points are the object size, position and certainty at each track point denoting level of confidence in this track being actually associated with the object it is being attributed to.

Object behaviors are higher-level interpretation of the video content and represented by the object that exhibits the behavior, the video shot this behavior takes place and the start and end frames of the behavior in the video clip. Behaviors are estimated and inferred based on the low-level attributes such as object’s type, size, position, motion characteristics and the video shot properties such

as location, time etc., as well as other objects in the shot. Behaviors can involve any number of objects across any amount of time (even across multiple video clips).

This description schema is implemented as the database for our video content extraction framework. In the database each of the five components of the description schema is realized as a database table appropriately linked to other tables with attributes described above (Figure 3). In addition to those attributes mentioned above an ID and a name and description of each video clip, shot or object and behavior are also added to the corresponding tables. Note that, in the implementation of the schema the generic behaviors are represented in a table with a behavior type and description. Generic behaviors are related to objects through the Object Behaviors table, which stores the database object, shot, and start/end frames for each actor in the behavior.

Figure 3 – The representation of the Database Schema



4. VideoViews Database Browser

For accessing a temporal data type such as video, both browsing and retrieval steps are equally important. Since retrieval is dependent on locating a specific portion of the data, efficient browsing helps the user to quickly assess the relevancy of the data. The database browser VideoViews described here is designed to best exploit the

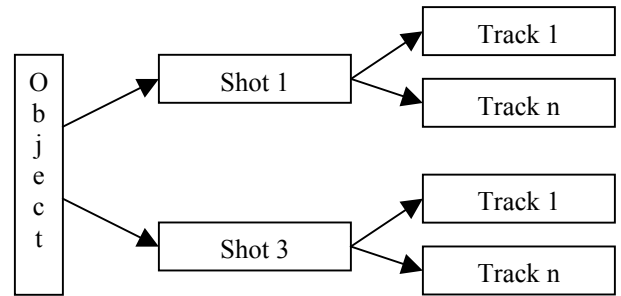
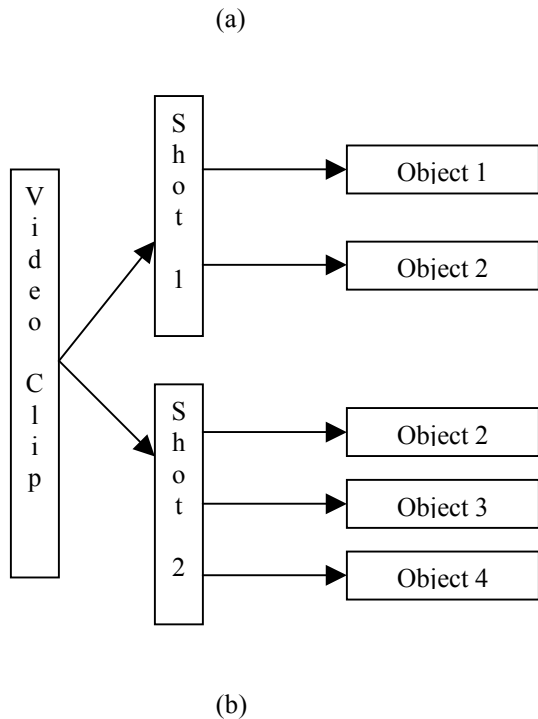
video description schema described in the previous section.

VideoViews provides multiple methods for displaying and analyzing the stored data about the video clips. These methods include database hierarchies (based on the hierarchical and semantic structure of the video), database table lists, generalized SQL queries and metadata displays.

VideoViews is hierarchically and semantically structured to enable users to freely navigate up and down within the database to visualize the raw data and the processed information from a number of perspectives for efficient data assessment while also supporting conventional database queries for retrieval. The structure can be used to logically navigate the information, and to select the items on which to perform further investigation. VideoViews facilitates browsing of the database using any of the following three structural hierarchical views;

- i) ClipsView: The video clip centric view Figure 4(a),
- ii) ObjectView: The object-centric view, Figure 4(b), and
- iii) BehaviorView The behavior-centric view Figure 4(c).

Figure 4. Schematic representation of video clip-centric(a), object-centric(b), and behavior-centric(c) navigator views



In the following we will describe the structure and semantics of each of these views and different browsing methods through an example video clip. The *Jay2Paul Exchange* clip depicts package exchange between two individuals. The clip has four shots defined by camera operations. In the first shot, a person carrying a bag is walking towards a road, a car comes to a stop on the roadside, the driver gets out of the car and walks toward the first person, they meet and the camera zooms in. The second shot is the zoomed in view showing the exchange of the bag, after the exchange the driver walks back to the car and the first person starts walking to the roadside while camera zooms out. During the third shot, as the first person starts getting out of the field of view, the camera pans to the left to follow him. The last shot shows a car approaching and picking up the person.

The view shown in Figure 5 is the *ClipsView*, this view facilitates top-down analysis and places the video assets at the highest level, followed by shots, objects, and finally tracks. The *Jay2PaulExchange* video clip is opened up to four shots (scenes), namely *First Scene*, *Zoomed-in* and *Zoomed-out Scenes* and *Panned -left*

Scene. Each scene can be opened to show objects in that scene. The *First Scene* has four objects: *Grey Car*, *Jay*, *Paul* and *White Car*. Under each object the tracks can be viewed as shown for object *Jay* in Figure 5.

The *ObjectView* illustrated in Figure 6, provides an object-centric look into the database thus, supporting bottom-up analysis. This view places objects at the highest level, followed by the shots in which an object is detected, and the tracks that object follows through the shot. Figure 6 shows several objects that are in the database, following the example video clip if we select the *Grey Car* object we see that it only exists in the *First Scene*, whereas *Jay* object exists in three scenes. The track points for each object can be viewed once the object is selected.

The *BehaviorView* (Figure 7) displays the behavior information as the first layer, followed by the shots across which the behaviors take place, the object that performs the behavior during each shot, and the specific tracks encompassed. Using the same example clip, examining the *Paul Enters Car* behavior, we see that it took place in the *Panned-left Scene*, with objects *Blue Car* and *Paul* involved, the tracks for these objects can be displayed for the duration of the behavior.

TableLists display in Figure 8 provides more information about the elements selected in a hierarchical view. There is a separate list for each type table of data,

and the list includes the useful/relevant information for that data type. Data is selected for viewing in the table list by selecting items in the hierarchical view. Selecting an item also selects its parents in the hierarchy.

The most conventional display method is the generalized SQL query as depicted in Figure 9. The user may enter a free-form SQL SELECT query into the *VideoViews* browser, and the resulting columns will be displayed in the *TableLists* view. Each selected column is given its own column in the list view. This method allows for any information to be retrieved from the database.

The final display method does not extract data from the database, but rather uses information stored in a video file, i.e. metadata. When a video clip is played in the *VideoViews*, the information extracted from its metadata is displayed in a separate specially designed and developed metadata window[8]. The metadata information (such as camera geometry, objects and tracks, behaviors, etc.) is updated in a frame accurate manner as the video plays. In addition to metadata window display, detected and tracked objects are marked through the video by a small marker overlain on the detected center. Double-clicking on this marker selects the object from the database, and displays all information about that object, thus linking the relatively concise metadata back to the wealth of information stored in the database. A frame from *Jay2PaulExchange* video clip *First Scene* is displayed along with the metadata window in Figure 10.

Figure 5 – Database Browser: ClipsView

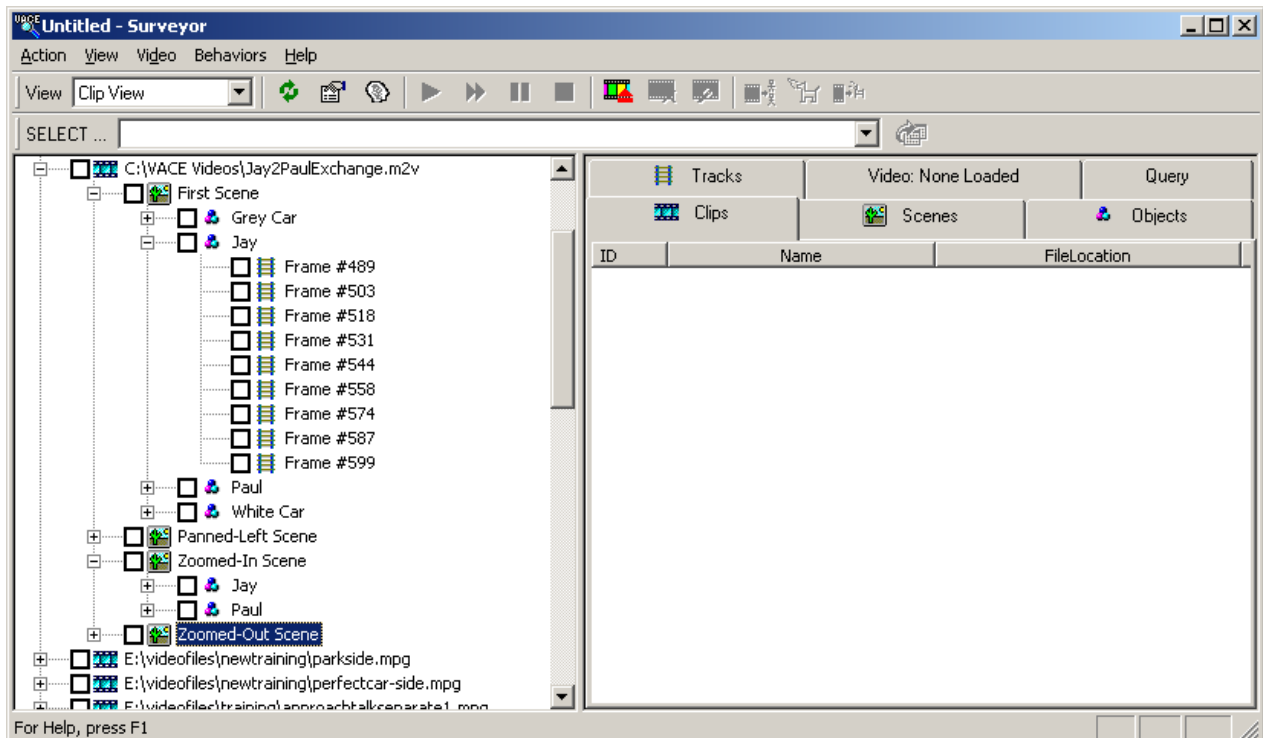


Figure 6 - Database Browser: ObjectView

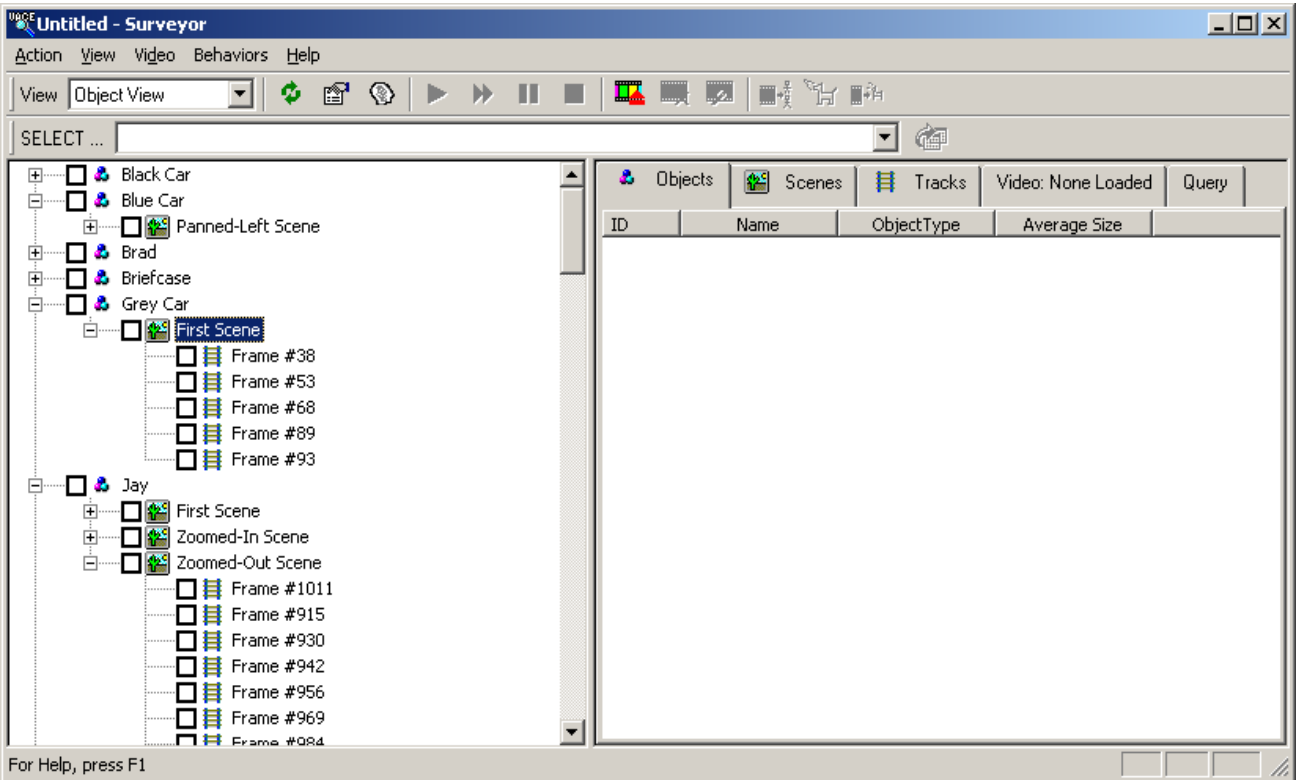


Figure 7 - Database Browser: BehaviorView

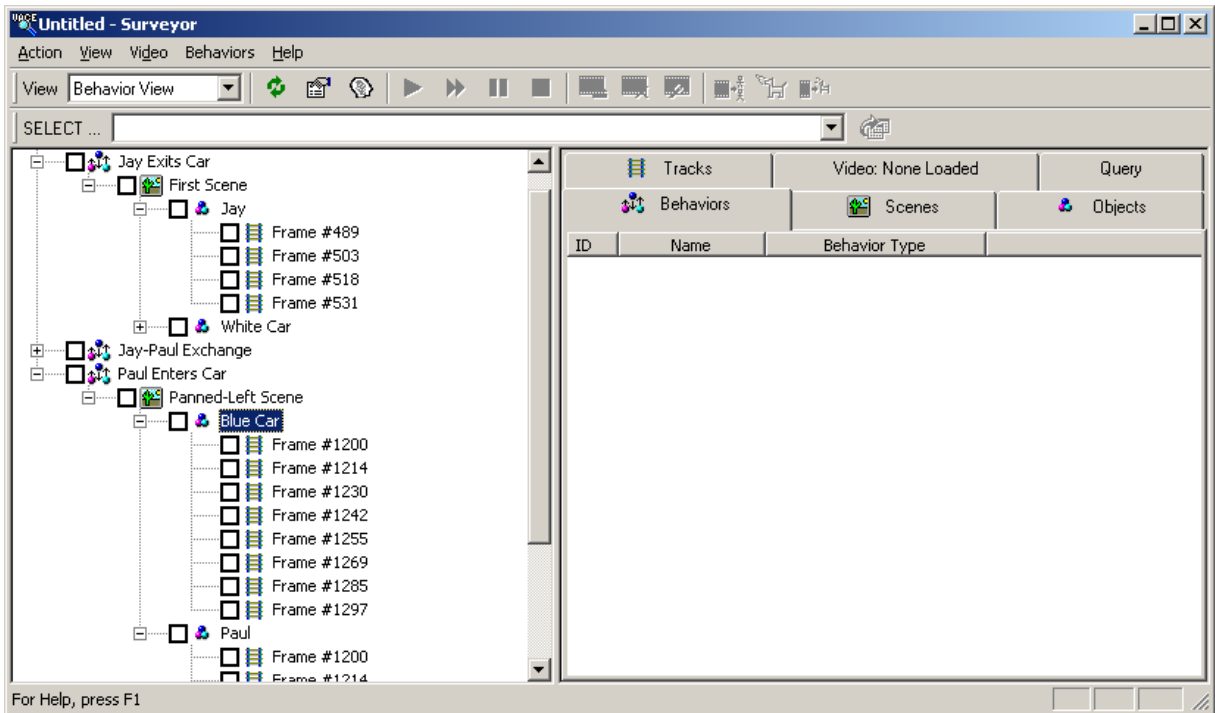


Figure 8 - Database Browser: TableLists display

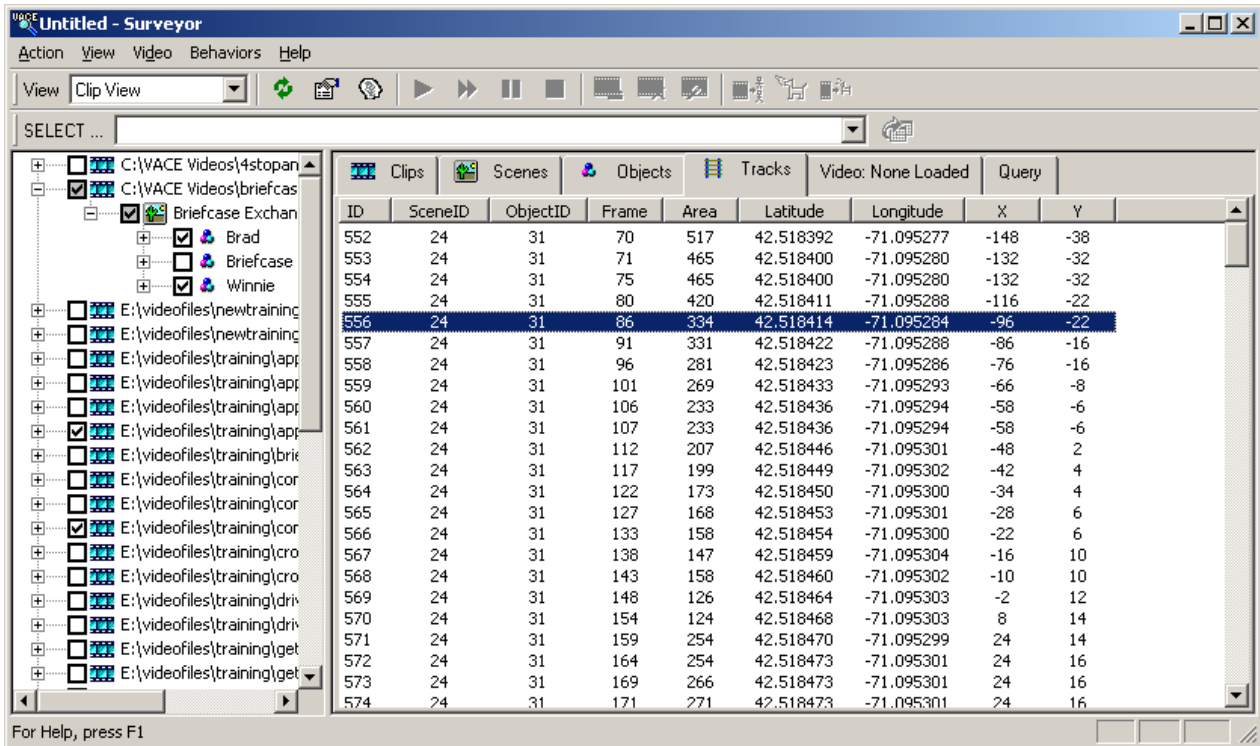


Figure 9 - Database Browser: General SQL Query Display

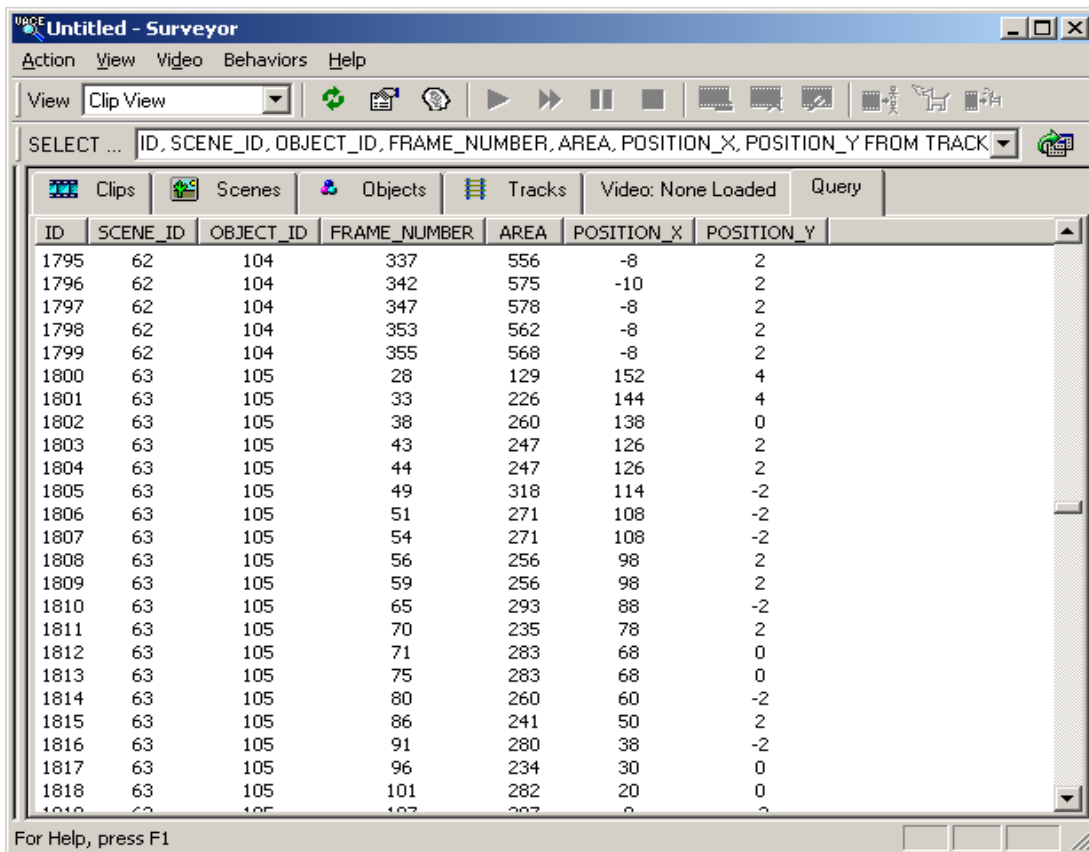
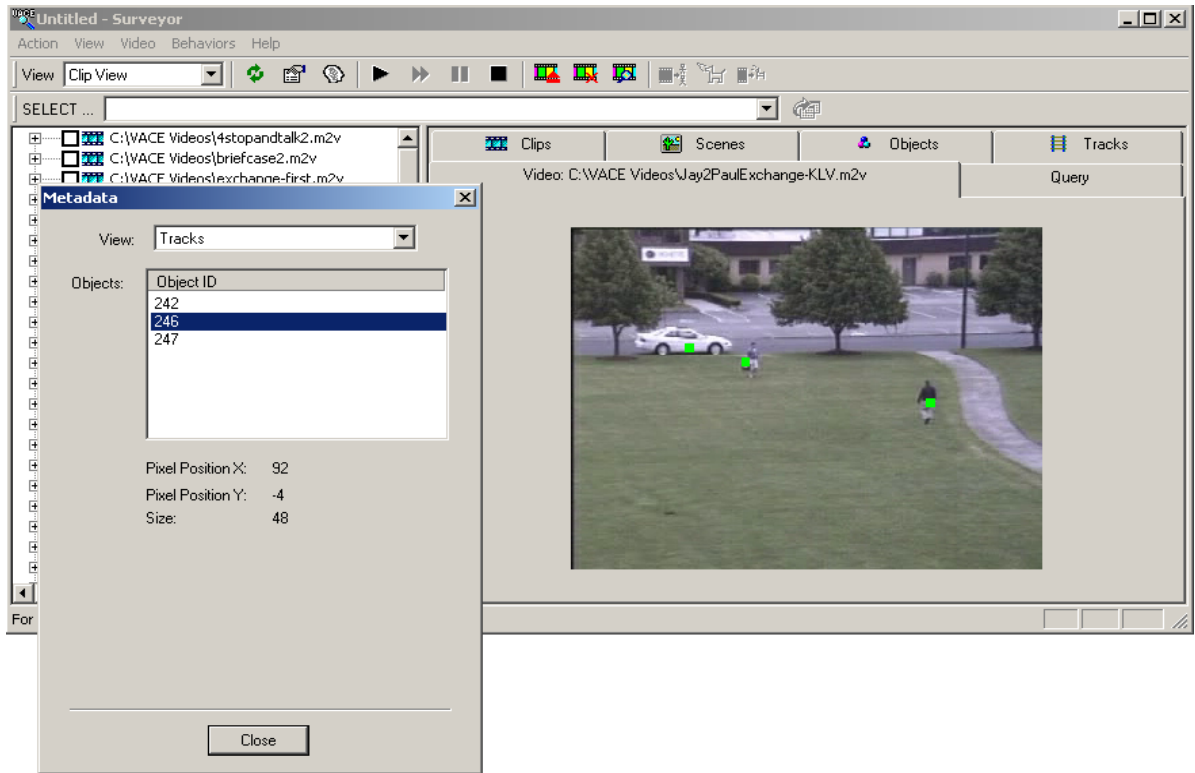


Figure 10 – Database Browser: Metadata Display



5. Conclusions

We presented our work in progress for a video analysis framework based description schema and browsing tool. This work presented here is part of a video analysis framework we developed under the ARDA/VACE program. The analysis framework includes scene geometry estimation, object detection and tracking and high level event understanding to develop a capability to automatically detect key events from video typical of that found in area security and surveillance environments. However, the proposed description schema is generic and can be applied to most video applications and the VideoViews database browser can be used to effectively store, browse, manipulate, annotate and retrieve video data. VideoViews combines multiple methods for displaying and analysing the stored information about the video asset set. These methods include database hierarchies, table lists, generalized SQL query, and video metadata displays. Each method has its own individual strengths, and combined to cover almost any scenario.

Other parts of the VACE project include detection, analysis and identification of components such as audio, faces and scene text from video and our intention is to

extend our description schema and the browser to include those elements into the framework.

Acknowledgements:

This work was supported in full by the Advanced Research and Development Activity (ARDA). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the US Government.

REFERENCES

1. J. R. Bach, C. Fuller, and A. Gupta, "The VIRAGE Image Search Engine: An open Framework for Image Management", Proc. SPIE '96, *Storage and Retrieval for Still Image and Video Database IV*, pp. 170-179, Feb.1996.
2. J. Fan, Y. Ji, and L. Wu, "Automatic Moving Object Extraction Toward Content-Based Video Representation and Indexing," *Journal of Visual Communications and Image Representation*, Vol. 12, No. 3, pp. 217-239, Sept. 2001.

3. A. M. Ferman, B. Gonsel and A. M. Tekalp, "Object-Based Indexing of MPEG-4 Compressed Video", *Proc. VCIP'97*, Vol. SPIE-3024, pp. 953-963, San Jose CA, Feb. 1997.
4. Forthcoming SMPTE 336M, Television – Data Encoding Protocol Using Key-Length_Value.
5. S. Guler, M. Rizkalla and M. Vetter "An Object Behavior And Event Based Index/Browse/Retrieve Framework And Tool For Video Data", in *Proc. 1st European Workshop on Content Based Multimedia Indexing*, Toulouse France, Oct. 1999.
6. S. Guler, "Scene and Content Analysis From Multiple Video Streams", in *Proc. 30th AIPR*, Washington D.C., Oct 1-12, 2001.
7. F. Idris and S. Panchanathan, "Review of Image and Video Indexing Techniques", *Jour. Of Vis. Comm. And Image Repr.* Vol. 8 No 2, pp. 146-166, June 1997.
8. W. H. Liang, "Mapping KLV Packets into Synchronous MPEG-2 Program Streams," *Proc. 36th SMPTE Advanced Motion Imaging Conference*, Dallas, TX, Feb. 2002, 36-13-TX.pdf
9. W. Niblack, R. Barber, W. Equitz, M. Glasman, D. Petkovic. P. Yanker, C. Faloutsos and G. Taubin, "The QBIC Project: Querying Images by Content Using Color Texture and Shape", *Storage Ret. Image Video Databases* No.1908, pp. 173-187, Feb 1993.
10. Y. Rui, and T. Huang, "Unified Framework for Video Browsing and Retrieval," *Handbook of Image & Video Processing*, Academic Press, pp. 705-715, 2000.
11. W. Wolf, "Key Frame Selection by Motion Analysis," in *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing* IEEE, New York, 1996.
12. H. Zang, C.Y. Low, S. W. Smoliar, and D. Zhong, "Video parsing, retrieval and browsing: An Integrated And Content-Based Solution," *Proceedings of the ACM Conference on MultiMedia*, ACM, New York, 1995.
13. Y. Zhuang, Y. Rui, T.S. Huang, and S. Mehrotra, "Adaptive Key Frame Extraction Using Unsupervised Clustering," in *Proceedings of the IEEE International Conference on Image Processing*, IEEE, New York, 1988.

Subjective interpretation of complex data: Requirements for supporting kansei mining process

Nadia Bianchi-Berthouze
University of Aizu
Aizu Wakamatsu, Fukushima-Ken, Japan
tel: +81 242 372790, fax: +81 242 372753
email: nadia@u-aizu.ac.jp

Tomofumi Hayashi
Japan Advanced Institute of Science and Technology
Nomi Gun, Ishikawa-Ken, Japan
email: thayashi@jaist.ac.jp

ABSTRACT

Information retrieval by subjective content has been recently addressed by the Kansei engineering community in Japan. Such information retrieval systems aim to include subjective aspects of the users, such as their sensitivity, in the querying criteria. While many techniques have been proposed to model such users' aspects, little attention has been placed on analyzing the multi-interpretation of the information involved in the process. We propose a data warehouse as a support for the mining process of such information. A unique characteristic of our data warehouse lays in its ability to store multiple hierarchical descriptions of the multimedia data. Such characteristic is necessary to allow the mining of complex data, not only at different levels of abstraction, but also according to multiple interpretation of the content. The framework we propose can be generalized to support the analysis of any type of complex data that relates to subjective cognitive processes and hence whose interpretation is greatly variable.

Keywords

Multimedia data mining, kansei user modeling, data-warehouse

1. INTRODUCTION

Information retrieval by subjective interpretation has been recently addressed by the kansei engineering community in Japan. Various web search engines [1], art appreciation systems [2,3,4], and design support systems [5,6] have been proposed to allow the retrieval of information on the basis of the subjective impression (kansei in Japanese) they convey to a human. An airline advertiser could, for example, query such search engines to "retrieve romantic images of airplanes". Figure 1 shows an example of query interface for such systems [1].

These systems query the web database using models of impression words. These models, called Kansei User

Models, are mathematical functions that map low level features, characterizing a multimedia information, into the word used to label the resulting subjective impression. These models are generally tailored to the subjectivity of each person or to groups of persons sharing a similar profile. As shown in Figure 1, the tailoring is based upon relevance feedback [7] entered by the person to assess the system's output.

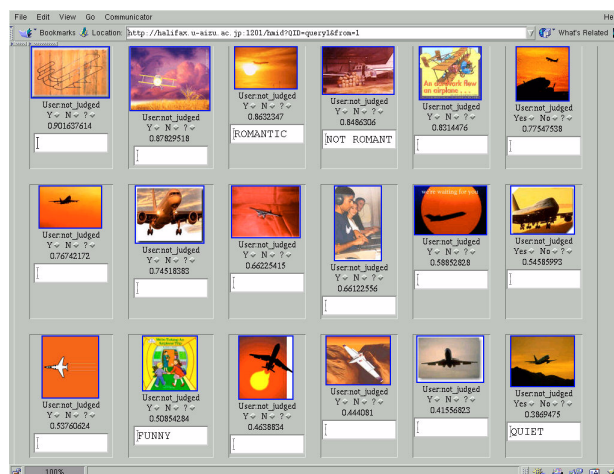


Figure 1. K-DIME (Distributed Environment for Kansei Information Management), a web-based search engine. It retrieves images from the web by entering objective keywords (e.g. "airplane") and subjective visual impression words (e.g. "romantic"). The screenshot displays the best 18 hits for "romantic" after the filtering of 452 images of airplanes. The user can input relevance feedback by writing impression words under each retrieved image.

While techniques for creating and adapting Kansei User Models have been widely explored, the analysis of the multi-interpretation and multi-description of such information has been largely ignored. Images, for example, allow for a multiple interpretation of their

content due to the attention and selection mechanisms [8] that our brain uses in filtering information (see Figure 2). These mechanisms are triggered by external factors such as mood, experience, goals, etc.

By avoiding such analysis, the resulting approach is inappropriate to account for the complexity and variability of users' subjective impressions. As a consequence, the results obtained so far have not been very encouraging.

In order to improve the performance of Kansei User Models, such information must be analyzed according to:

- multiple interpretation, i.e. subjective perceptions of the information contained in multimedia data;
- the dynamic selection of the salient (to a certain subjective impression) features describing the information;
- the fuzziness and limits in the meaning of the words used to label a given subjective impression.

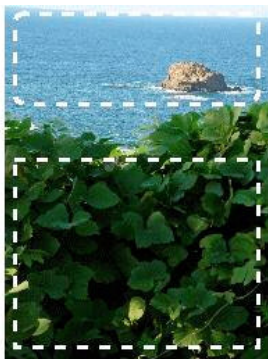


Figure 2. Our brain uses attention and selection mechanisms to interpret images. The impression conveyed by these images changes according to the focus of attention of the user. (See section 4 for more details).

We believe that a computational environment able to manage multiple interpretation should facilitate such analysis. In this research we review the kansei user modeling methodology proposing the use of KITE, Kansei management Environment, an environment that

allows to store complex data and to handle multi-signature of such data taken at different levels of descriptions. Moreover, it enables the handling of signatures whose structure is not predefined. This is a necessary requirement in order not to limit the possible emergent perspectives on these data.

We describe KITE in the context of the modeling of the mapping between image features and subjective visual impression. The paper is organized as follows. After briefly describing the architecture of KITE and introducing the requirement for the kansei modeling process, we propose a hierarchical data model to store and access multiple hierarchical descriptions of images. Finally, we present the interface library that allows the access to the data in a transparent way from the DBMS language and data model. We conclude with an example of use of KITE in the framework of kansei data mining.

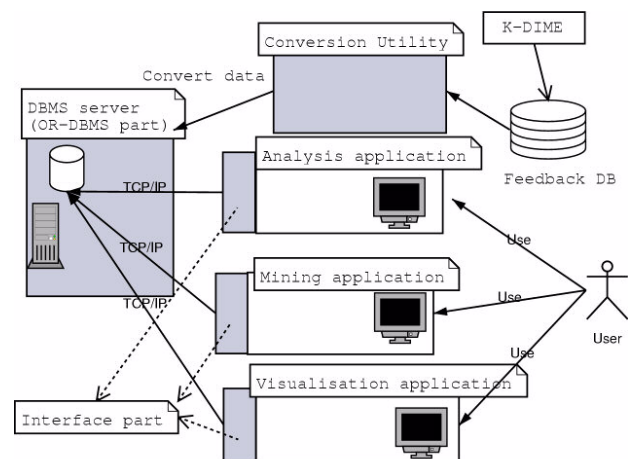


Figure 3. KITE's architecture

2. KITE: A KANSEI MANAGEMENT ENVIRONMENT

KITE, Kansei management Environment, has been developed to support a deeper analysis of the user feedback and improve image retrieval performance. User feedback is collected using a kansei search engine (K-DIME[1], see figure 1) upon a user evaluation of the system output.

Figure 3 shows the architecture of KITE. As highlighted in gray in the figure, 3 main modules form its architecture: a DBMS server, a software interface and a set of data conversion utilities. The DBMS server serves as a data warehouse to store the users' profile and the user feedback collected over time. The software interface creates a bridge between the kansei mining applications and the DBMS server offering a set of functions to facilitate the access to the data.

User profiles and the user feedback are loaded into the DBMS server using the data conversion utilities. These utilities integrate and convert the external data into the DBMS server data model.

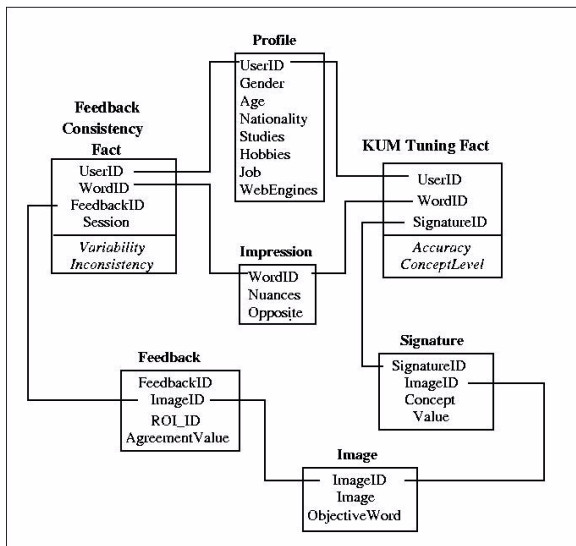


Figure 4: An example of fact constellation schema to support kansei data mining

3. DBMS SERVER AND DATA MODELING

In this section, we present a scenario used as case study: the mining process of user feedback to create Kansei User Models for image retrieval. The fact constellation schema shown in Figure 4 represents the set of data involved in the analysis of the feedback and in the creation and adaptation of Kansei User Models (KUMs). It presents 2 fact tables: FeedbackConsistency fact table and KUM-Tuning fact table.

The FeedbackConsistency fact table is described in terms of userID, WordID, and the feedbackID collected during image retrieval sessions. The user profile simply contains information that can affect the user's subjectivity, such as nationality, age, studies, gender, goal, etc. A user feedback consists of an image retrieved by the search engine for the wordID, some user selected portions (ROIs) of the image to highlight his/her focus of attention, and finally his/her dis/agreement with the search engine on the evaluation of the image.

The FeedbackConsistency fact table supports the analysis of the variability of the user subjectivity, i.e. the variability of his/her image evaluation. This analysis is

directed to detect different patterns of inconsistencies. We identify three types of inconsistencies:

- intrinsic or derived from a different evaluation of the same selected information (temporal evolution in the observer),
- true or derived from the misuse of a word in conveying a subjective experience and
- attentional or derived from a different reading of the information by the selection mechanism.

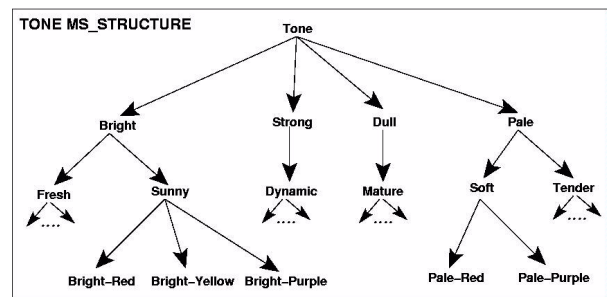


Figure 5. Hierarchical aggregation of color features into higher-level tonality concepts.

The detection of inconsistencies is necessary to drive the mining process and the tuning of the KUMs. The KUM-Tuning fact table is described in terms of userID, wordID, SignatureID. A Signature (S) describes the content of the image or a portion of it (ROI) according to its color, texture and shape characteristics [9]. This fact table aims at detecting the set of low level features that are the partial cause of the visual impression the images conveyed to the user. A description of the tools to perform such analysis and create the KUMs can be found in [10].

4. MULTISIGNATURE: A HIERARCHICAL META SCHEMA

The information present in an image is filtered and aggregated [11] in various ways by our brain. Our state of mind, goal and/or past experience direct our attention on some aspects of an image and at different levels of abstraction [8]. For example, different areas of the sea landscape shown in Figure 2 (up) convey different impressions: the blue sea might convey an impression of freedom while the dark wall of leaves may convey the opposite impression. In the second picture of the same figure, a romantic impression is caused by degrading nuances of the red-yellow hues on a dark background (we refer to the color version of this image), while a peaceful impression is caused by the slight light up of dark regions without referring to its hues (colors).

Thus, one of the aims of the mining process will be to determine the focus of attention in an image and the filtered features and concepts that caused the associated impression in a feedback. In order to analyze and capture the relevant features, we propose to see the low-level features as leaves of hierarchical meta-schemas. These meta-schemas allow the creation and management of various interpretations of an image content.

Mining Task: evaluation of the consistency of user's feedback for the word "quiet" according to 2 different hierarchical signatures:

```
USE database KanseiFeedback
MINE COMPARISON AS "ToneSignature"
IN RELEVANCE to agreement, features-concepts
WHERE S.type='Tone'
VERSUS "ColorSignature"
WHERE S.type='Color'
ANALYZE count%
FROM signature S
WHERE S.userId='Akko' AND
S.objectiveLabel='landscape' AND
S.kanseiLabel='quiet'
DISPLAY AS table
```

Figure 6. An example of mining activity on the set of signatures of images classified as "quiet" by the user "Akko".

Figure 5 shows an example of aggregation of low-level features. This structure shows a hierarchical set of concepts for the tone signature of images. In this particular structure, the color dimensions (leaves) are aggregated into higher level concepts that combine the feeling conveyed by tonality aspects of color with its hues. Using this hierarchy, regions with bright red, purple and yellow colors can be labeled with the concept "sunny regions" while regions with pale tonality of the same hues are labeled as "soft regions".

Figure 6 shows an example of mining activity aimed at comparing the signatures of "quiet" images according to 2 different hierarchical structure levels. In the first case, we used the tonality information of the image contents without differentiating between hues. In the second case, the color content of the image is described according to the second level of the hierarchical structure of figure 5, thus combining hue and tonality information of the color.

We chose the 2 signatures by analyzing the variability of the values of the color features in the training sets of nuances of the word "quiet". Such training sets were collected through experiments in which users were asked to group images, previously judged as "quiet", by nuances of such impression.

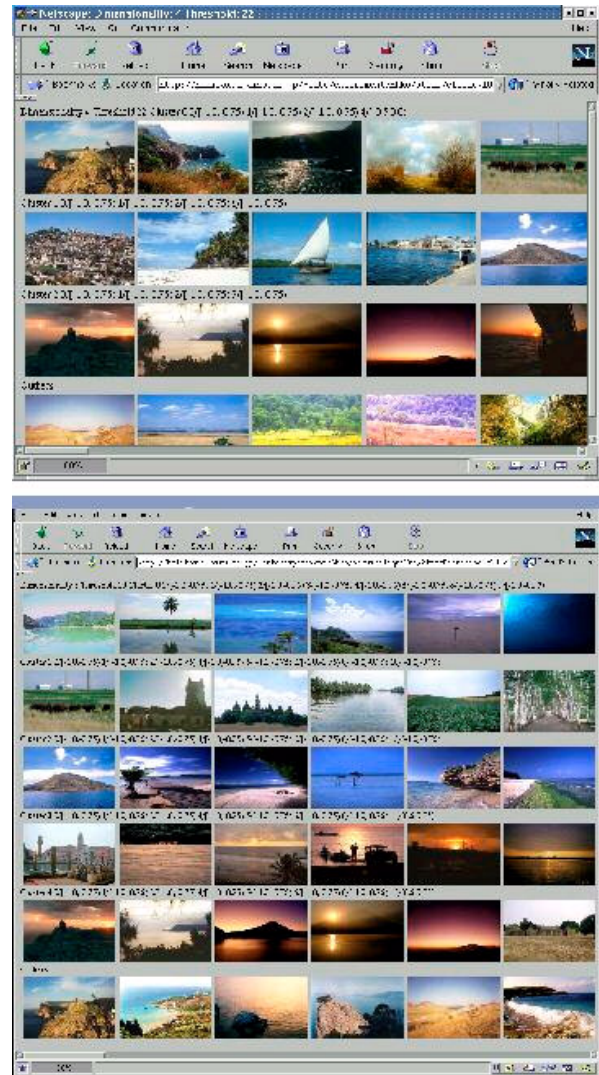


Figure 7. Results of the clustering performed on 2 different signature structures of the same set of "quiet" images. The clustering aims at identifying groups of images that can be associated with nuances of the impression word "quiet".

Using such signatures, we performed an automatic clustering of the set of images. Figure 7 shows the results of such clustering. The clustering was used to realize a stratified sampling of the total training set for the impression word "quiet". The selected samples were used to create the Kansei User Model for that impression word. Table 1 shows the learning error for 5 different impression words, after having applied the same sampling process for the tonality signature and for the color signature.

5. DBMS SERVER AND META-SCHEMA IMPLEMENTATION

In this section, we describe in more details the implementation of the model to store multiple hierarchical signatures (MS_Structure). We will skip here the straightforward implementation of the data model for the other data involved, e.g. user profile.

Words:	Tone Signature		Color Signature	
	Training Set	Test. Set	Training Set	Test. Set
Atsui (warm)	0.0200	0.0452	0.0188	0.0398
Bukimina scary)	0.0124	0.0329	0.0177	0.0427
Kakkooii (good feeling)	0.0331	0.0513	0.0230	0.0402
Miwakuteki (enchanting)	0.0270	0.0504	0.0204	0.0361
Shizuka (quiet)	0.0240	0.0462	0.0327	0.0567

Table 1: Comparison of the training and testing error obtained using 2 different signatures for the same set of images. The training is performed using neural networks trained by back-propagation algorithm.

```
CREATE TABLE MS_Structure (
    ImageID varchar NOT NULL,
    Depth int NOT NULL,
    NodePath varchar NOT NULL,
    ValueType varchar NOT NULL
);
CREATE TABLE floatValue (
    Value double precision
) INHERITS (MS_Structure);
CREATE TABLE textValue (
    Value text
) INHERITS (MS_Structure);
....
```

Figure 8: SQL expressions for the definition of the tables containing the hierarchical structure of the signature and the signature values.

We implemented our data model in SQL on PostgreSQL [12], an Object Relational Database Management System (OR-DBMS). In order to allow the storage of multiple hierarchical signatures, we adopt the following structure: each node of a hierarchy is characterized by a name, a depth and a value. The name indicates not only the

concept but also the parent names of the concept. For example, the node "soft" will be labeled as "Tone.Pale.Soft", where "Tone" identifies the type of hierarchical structure. This structure enables also the storing of hierarchical structure whose depth is not predefined.

Signatures can have values of different type. For example, the signature describing the percentage of a color in an image will have values of type float, while shape signatures describing the main shapes detected in an image can use polygonal type (a set of points).

In order to allow the storage of signature values of different types, we use the inheritance mechanism offered by PostgreSQL. We created 2 types of tables to store the signatures: MS_Structure table and <type>Value tables. The MS_Structure table and the different <type>Value tables are created by the SQL expression given in Figure 8.

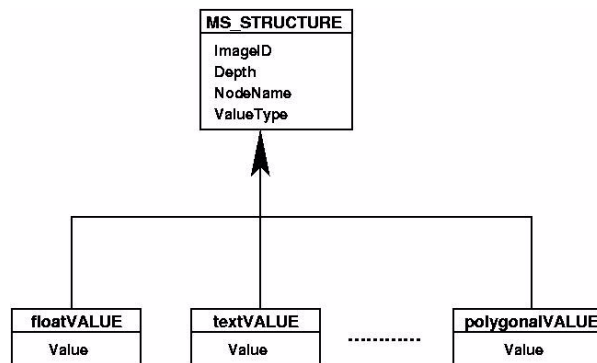


Figure 9: Inheritance model for storing multi-type signatures.

The data model of the multi signature is shown in Figure 9. The signature values are stored in a <type>Value table according to the type of its values. Hence, each <type>Value table has an attribute "value", to store the value of a concept, and also all the attributes of the MS_Structure table. Table 2 shows an example of <type>Value table containing the color signatures of images. Each row of this table corresponds to a node in the hierarchical structure of Figure 5 for the signature of the image "img456". The second attribute of the table indicates the abstraction level of the concept in the hierarchical structure used. The third attribute indicates the complete path in the hierarchy. The attribute "value" indicates the percentage of pixels in the image that reflects the correspondent color concept.

This data model does not require the predefinition of all the types of hierarchical structures that can be stored in the DBMS server. In fact, it allows an introduction at run-

time of any new hierarchical structure and the related image signatures. This is an important requirement for our environment. The visual data mining process could very well request the analysis of new types of aggregation of low-level features of images.

Image	Depth	NodePath	Value
Img456	4	Tone.Bright.Sunny.Bright Red	0.09
Img456	4	Tone.Bright.Sunny.Bright Yelloww	0.01
Img456	4	Tone.Bright.Sunny.Bright Purple	0.02
Img456	3	Tone.Bright.Sunny	0.4
Img456	4	Tone.Bright.Fresh.Bright Red	0.01
Img456	4	Tone.Bright.Fresh.Bright Yelloww	0.1
Img456	4	Tone.Bright.Fresh.Bright Purple	0.03
Img456	3	Tone.Bright.Fresh	0.15
Img456	2	Tone.Bright	0.4
Img456

Table 2. Example of <double>Value table. Values in the Value-column indicate the percentage of the correspondent color features present in the image. The attribute valueType has been left out for readability.

6. THE SOFTWARE INTERFACE

In order to allow an easy access to the data, KITE contains a software interface. The software interface lets external applications access the DBMS servers independently of the DBMS's model and the DBMS's language. The library consists of two packages. One contains DBMS-access tools, the second one is a collection of data-access tools.

The software interface module is installed on each client machine and communicates with the DBMS server through TCP/IP, allowing distributed computation. The DBMS-access tools provide the functions to create the TCP/IP connection between a client application and DBMS module. It also contains the data-conversion functions between DBMS primitive data structures and Java primitive data structures. The Data-Access tools supply the functions for accessing the data in the DBMS server. In particular, it enables the access to data at different abstraction levels by exploiting the hierarchical descriptions of the multi-signatures.

The interface module is implemented in Java in order to offer architecture independence. Figure 10 shows an example of software application (written in Java) to mine the data in the DBMS server. This application uses the library functions to connect to the DBMS server and to

extract the signatures of the images that the user "Akko" has classified as "quiet".

The retrieved signatures describe such images according to the color concepts in the third level of the hierarchical structure shown in Figure 5. A clustering algorithm is applied to these signatures in order to detect patterns/clusters that relate to nuances of impression word "quiet" (e.g. peaceful, silent, etc.) (see figure 7). Their detection would facilitate the creation of KUMs for such impression word [10].

```

.....
pgConnection con;

int main ( ) {

//setting for the selection of the data to mine
String username="Akko";
String word="quiet";
String MS_Structure="Tone";
int depth=2;

//open connection to the DBMS server of KITE
con= new pgConnection
    ("jdbc:postgresql://hali/kitedb?user=hayashi");
con.createConnection();

//selection of the data to mine
clUser usr = getUserByID (con, username);

//retrieving the image signature aggregated according to
the abstraction level 2 of the Tone MS_Structure (Figure
5)
String[] images= usr.getImageID (con, word);
double[][] values=usr.getSignatureTS (con, images,
    MS_Structure,
    depth);

//clustering of the image signatures
Cluster cl =new Cluster(values);
boolean check=cl.performClustering();

```

Figure 10: An example of software application to mine the data stored in the DBMS server of KITE. The data are retrieved using the software interface functions of KITE.

7. CONCLUSIONS

The framework we propose supports the management of multi-descriptions of complex data. Multi-descriptions are necessary to investigate the various interpretations that humans can perform on complex data. This is an important issue when interpreting data that relate to human subjective cognitive processes. Complex data do not limit to multimedia data. Another field of applications

for such framework could be the analysis of body language in multi-modal communication environment [13,14,15,16]. In this case a description of a body gesture through a hierarchical representation of the body parts could help the analysis of relevant features with respect to the message being set or interpreted. In summary, we believe that such framework should enable:

- to dynamically defining hierarchical structures for describing the content of the complex data,
- to maintain multiple descriptions of the same data,
- to support the dynamical selection of the abstraction level of interest for the mining process,
- to handle in a transparent way a multiple type value for such descriptions,
- an easy access to such descriptions independently of the language and the data model used to store the data.

REFERENCES

1. Inder, R., Bianchi Berthouze, N., and Kato, T. K-DIME: A Software Framework for Kansei Filtering of Internet Material. In *Proceedings of IEEE International Conference of Systems, Man and Cybernetics*, V.6, pp.358-363.
2. Yoshida, K., Kato, T., and Yanoru, T. A Study of Database Systems with Kansei Information, In *Proceedings of IEEE Intern. Conference on Systems Man and Cybernetic* 6, (1999), 253-256
3. Hattori, R., Fujiyoshi, M., and Iida, M. An Education System on WWW for Study Color Impression of Art Paintings Applied NetCatalog, , In *Proceedings of IEEE Intern. Conference on Systems Man and Cybernetic* 6, (1999), 218-223
4. Imai, T., Yamauchi, K., and Ishi, N. Color Coordination System on Case Based Reasoning System using Neural Networks. In *Proceedings of IEEE Intern. Conference on Systems Man and Cybernetic* 6, (1999), 224-229
5. Lee, S., and Harada, A. A Design Approach by Objective and Subjective Evaluation of Kansei Information. In *Proceedings of International Workshop on Robot and Human Communication*, (1998), IEEE Press, 327-332.
6. Shibata, T., and Kato, T. Kansei Image Retrieval System for Street Landscape. Discrimination and Graphical Parameters based on correlation of Two Images. In *Proceedings of IEEE International Conference on Systems Man and Cybernetics* 6, (1999), 247-252
7. Rui, Y., Huang, T.S., Ortega, M., and Mehrotra, S. Relevance Feedback: A power tool in interactive content-based image retrieval. *IEEE Transaction on Circuits and Systems for Video Technology* 8, 5, (1998) 644--655.
8. Pashler, H. Attention and Visual Perception: Analysing Divided Attention. *Visual Cognition* V.2, S.M Kosslyn, D.N. Osherson (eds), MIT Press, (1996), pp. 71-100
9. Bianchi-Berthouze, and N., Berthouze, L. Exploring Kansei in Multimedia Information. *International Journal on Kansei Engineering*, V2, N2, (2001), 1-10
10. Bianchi-Berthouze, N. Mining Multimedia Subjective Feedback. *International Journal of Information Systems*, Kluwer Academic Publishers, 2002
11. Kobayashi, S. Colorist: a practical handbook for personal and professional use. Kodansha
12. PostgreSQL: <http://www.postgresql.org/>
13. Bianchi-Berthouze, N., Lisetti, C. Modeling multimodal expression of users's affective subjective experience. *International Journal on User Modeling and User-Adapted Interaction: Special Issue on User Modeling and Adaptation in Affective Computing* 12, 1, (2002), 49-84
14. Nakata, T, Sato, T. and Mori, T. Expression of Emotion and Intention by Robot Body Movement. In *Proceedings of Conference of International Autonomous Systems* 5 (IAS-5), (1998).
15. Cardon, A. The approach of the concept of embodiment for autonomous robot: towards consciousness of its body. In *Proceedings of AEMAS Workshop on ACAI*, (Prague, 2001)
16. Camurri, A., Hashimoto, S. , Ricchetti, M. , Ricci, A. , Suzuki, K. , Trocca, R., and Volpe, G. EyesWeb - Toward Gesture and Affect Recognition in Dance/Music Interactive Systems. *Computer Music Journal*, 24, 1, (2000), 57-69, MIT Press

User Concept Pattern Discovery Using Relevance Feedback and Multiple Instance Learning for Content-Based Image Retrieval

Xin Huang
Distributed Multimedia
Information System
Laboratory
School of Computer Science
Florida International
University
Miami, FL 33199
USA
xhuan001@cs.fiu.edu

Shu-Ching Chen
Distributed Multimedia
Information System
Laboratory
School of Computer Science
Florida International
University
Miami, FL 33199
USA
chens@cs.fiu.edu

Mei-Ling Shyu
Department of Electrical and
Computer Engineering
University of Miami
Coral Gables, FL 33124
USA
shyu@miami.edu

Chengcui Zhang
Distributed Multimedia
Information System
Laboratory
School of Computer Science
Florida International
University
Miami, FL 33199
USA
czhang02@cs.fiu.edu

ABSTRACT

Understanding and learning the subjective aspect of humans in Content-Based Image Retrieval has been an active research field during the past few years. However, how to effectively discover users' concept patterns when there are multiple visual features existing in the retrieval system still remains a big issue. In this paper, we propose a multimedia data mining framework that incorporates Multiple Instance Learning into the user relevance feedback in a seamless way to discover the concept patterns of users, especially where the user's most interested region and how to map the local feature vector of that region to the high-level concept pattern of users. This underlying mapping can be progressively discovered through the feedback and learning procedure. The role user plays in the retrieval system is to guide the system mining process to his/her own focus of attention. The retrieval performance is tested under a couple of conditions.

Keywords

Multimedia Data Mining, Image Retrieval, Multiple Instance Learning, Relevance Feedback

1. INTRODUCTION

Recently, many efforts have been made to Content-Based Image Retrieval (CBIR) in order to personalize the retrieval engine. The subjectivity of human perception of visual content plays an important role in the CBIR systems. It is very often that the retrieval results are not very satisfactory especially when the level of satisfaction is closely related to user's subjectivity. For example, given a query image with a tiger lying on the grass, one user may want to retrieve those images with the tiger objects in them, while another user may find the green

grass background more interesting. User subjectivity in image retrieval is a very complex issue and difficult to explain. Therefore, a CBIR system needs to have the capability to discover the users' concept patterns and adapt to them.

In this paper, we propose a multimedia data mining framework that can dynamically discovering the concept patterns of a specific user to allow the retrieval of images by the user's most interested region. The discovering and adapting process aims to find out the mapping between the local low-level features of the images and the concept patterns of the user with respect to how he/she feels about the images. The proposed multimedia data mining framework seamlessly integrates several data mining techniques. First, it takes advantages of the user feedback during the retrieval process. The users interact with the system by choosing the positive and negative samples from the retrieved images based on their own concepts. The user feedback is then fed into the retrieval system and triggers the modification of the query criteria to best match the users' concepts [14]. Second, in order to identify the user's most interested region within the image, the Multiple Instance Learning [16, 18] and neural network techniques are integrated into the query refining process. The Multiple Instance Learning technique is originally used in categorization of molecules in the context of drug design. Each molecule (bag) is represented by a bag of possible conformations (instances). In image retrieval, each image is viewed as a bag of image regions (instances). In fact, the user feedback guides the system mining through the positive and negative examples, and tells the system to shift its focus of attention to the region of interest. Compared with other Multiple Instance Learning methods used in CBIR, our methodology has the following advantages: 1) Instead of manually dividing each picture into many overlapping regions [16], we adopt the image segmentation method in [5] to partition the images in a more natural way; 2) In

other Multiple Instance Learning based image retrieval systems such as [18], the users are usually asked to provide the positive and negative samples by looking through a huge amount of images in the database. While in our framework, user feedback is used in the image retrieval process, which makes the process more efficient and precise. It is more efficient since it is easy for the user to find some positive samples among the initial retrieved results. It is more precise since among the retrieved images, the user can select the negative samples based on his/her subjective perception. The reason is that the selected negative ones have similar features/contents with the query image but they have different focuses of attention from the user's point of view. By selecting them as negative samples, the system can better distinguish the real needs of the users from the "noisy" or unrelated information via Multiple Instance Learning. As a result, the system can discover which feature vector related to a region in each image best represents the user's concept, and furthermore, it can determine which dimensions of the feature vector are important by adaptively reweighing them through the neural network technique.

This paper is organized as follows. Section 2 briefly introduces the related work in Relevance Feedback and Multiple Instance Learning. Section 3 introduces the details of the Multiple Instance Learning and neural network techniques used in our framework. The proposed multimedia data mining framework for content-based image retrieval using user feedback and Multiple Instance Learning is described in Section 4. The experimental results are analyzed in Section 5. Section 6 gives the conclusion and future work.

2. RELATED WORK

2.1 Retrieval Using Relevance Feedback

While lots of research efforts establish the base of CBIR, most of them relatively ignore two distinct characteristics of the CBIR systems: (1) the gap between high-level concepts and low-level features, and (2) the subjectivity of human perception of visual content. To overcome these shortcomings, the concept of relevance feedback (RF) associated with CBIR was proposed in [13]. Relevance feedback is an interactive process in which the user judges the quality of the retrieval performed by the system by marking those images that the user perceives as truly relevant among the images retrieved by the system. This information is then used to refine the original query. This process iterates until a satisfactory result is obtained for the user.

In the past few years, the RF approach to image retrieval has been an active research field. This powerful technique has been proved successful in many application areas. Various ad hoc parameter estimation techniques have been proposed for the RF approaches. The method of RF

is based on the most popular vector model [4] used in information retrieval. The RF techniques do not require a user to provide accurate initial queries, but rather estimate the user's ideal query by using positive and negative examples (training samples) provided by the user. The fundamental goal of these techniques is to estimate the ideal query parameters (both the query vectors and the associated weights) accurately and robustly. Most of the previous RF researches [1][6] are based on the low-level image features such as color, texture and shape and can be classified into two approaches: query point movement and re-weighting techniques [8]. More recently, the new trend towards taking advantages of the semantic contents of the images in addition to the low-level features has appeared.

2.2 Multiple Instance Learning

Dietterich et al. [7] introduced the Multiple Instance Learning problem and presented Multiple Instance Learning algorithms for learning axis-parallel rectangles (APR). In [3], Auer et al. proposed MULTIINST algorithm for Multiple Instance Learning that is also an APR based method. In [10], Maron et al. introduced the concept of Diversity Density and applied a two-step gradient ascent with multiple starting points to find the maximum Diversity Density. Based on the Diversity Density, Qi Zhang et al. [17] proposed EM-DD algorithm. In their algorithm, it was assumed that each bag has a representative instance and treated it as a missed value, and then the EM (Expectation-Maximization) method and Quasi-Newton method were used to learn the representative instances and maximize the Diversity Density simultaneously. [12] also used the EM method to do Multiple Instance Regression. Jun Wang et al. [15] explored the lazy learning approaches in Multiple Instance Learning. They developed two kNN-based algorithms: Citation-kNN and Bayesian-kNN. In [19], Jean-Daniel Zucker et al. tried to solve the Multiple Instance Learning problem with decision trees and decision rules. Jan Ramon et al. [11] proposed the Multiple Instance Neural Network. Stuart Andrews et al. [2] utilized the Support Vector Machine in Multiple Instance Learning.

In this paper, one of the main goals is to map the original visual feature space into a space that better describes the user desired high-level concepts. In other words, we try to discover the specific concept patterns for an individual user via user feedback and Multiple Instance Learning. In our method, we assume the user searches for those images close to the query image and responds to a series of machine queries by declaring the positive and negative sample images among the displayed images. Efficiency can be measured by the average number of queries necessary to locate the desired images. For this purpose, we introduce a multiple instance feedback model that accounts for various concepts/responses of the user. Each

new query is chosen to achieve the user expectation more closely given the previous user responses. Compared with the traditional RF techniques, our method differs in the following two aspects:

1. It is based on such an assumption that the users are usually more interested in one specific region (blob object) than other regions of the query image. However, to our best knowledge, the recent efforts in the RF techniques are based on the global image properties of the query image. In order to produce a higher precision, we use the segmentation method proposed in [5] to segment an image into regions (segments) that roughly correspond to objects, which provides the possibility for the retrieval system to discover the most interested region for a specific user based on his feedback.
2. In many cases, what the user is really interested in is just a region (an object) of the query image (example). However, the user's feedback is on the whole image. How to effectively identify the user's most interested region (object) and to precisely capture the user's high-level concepts based on his/her feedback on the whole image have not received much attention yet. In this paper, we apply Multiple Instance Learning method to discover the user's interested region and then mine the user's high-level concepts. By doing so, not only the region-of-interest can be discovered, but also the ideal query point of that query image can be approached within several iterations.

3. THE PROPOSED MULTIPLE INSTANCE LEARNING FRAMEWORK

In a traditional supervised learning scenario, each object in the training set has a label associated with it. The supervised learning can be viewed as a search for a function that maps an object to its label with the best approximation to the real unknown mapping function, which can be described with the following:

Definition 1. Given an object space Ω , a label space Ψ , a set of objects $O = \{O_i | O_i \in \Omega\}$ and their associated labels $L = \{L_i | L_i \in \Psi\}$, the problem of supervised learning is to find a mapping function $\hat{f}: \Omega \rightarrow \Psi$ so that the function \hat{f} has the best approximation of the real unknown function f .

Unlike the traditional supervised learning, in multiple instance learning, the label of an individual object is unknown. Instead, only the label of a set of objects is available. An individual object is called an instance and a set of instances with an associated label is called a bag. Specifically, in image retrieval there are only two kinds of

labels which are Positive and Negative respectively. A bag is labeled Positive if the bag has one or more than one Positive instance and is labeled negative if and only if all its instances are Negative. The Multiple Instance Learning problem is to learn a function mapping from an instance to a label (either *Positive* or *Negative*) with the best approximation to the unknown real mapping function, which can be defined as follows:

Definition 2. Given an instance space Φ , a label space $\Psi = \{1 \text{ (Positive)}, 0 \text{ (Negative)}\}$, a set of n bags $B = \{B_i | B_i \in P(\Phi), i = 1 \dots n\}$, where $P(\Phi)$ is the power set of Φ , and their associated labels $L = \{L_i | L_i \in \Psi\}$, the problem of Multiple Instance Learning is to find a mapping function $\hat{f}: \Phi \rightarrow \Psi$ so that the function \hat{f} has the best approximation of the real unknown function f .

3.1 Problem Definition

Let $T = \langle B, L \rangle$ denote a training set where $B = \{B_i\} (i = 1 \dots n)$ are the n bags in the training set; $L = \{L_i\} (i = 1 \dots n)$ are the set of labels of B and L_i is the label of B_i . A bag B_i contains m_i instances that are denoted by $I_{ij} (j = 1, \dots, m_i)$. The function f is the real unknown mapping function that maps an instance to its label, and the function f_{ML} denotes the function that maps a bag to its label. In Multiple Instance Learning, a bag is labeled *Positive* if at least one of its instances is *Positive*. Otherwise, it has *Negative* label. Hence, the relationship between the functions f and f_{ML} can be described in Figure 1.

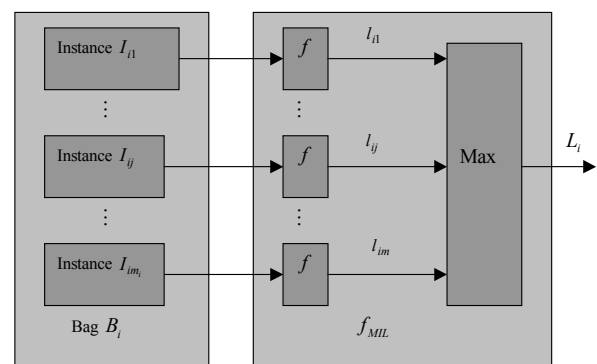


Figure 1. Relationship between functions f and f_{ML}

As can be seen from this figure, the function f maps each instance I_{ij} in bag B_i to its label l_{ij} . The label L_i of the bag B_i is the maximum of the labels of all its instances, which means $L_i = f_{ML}(B_i) = \text{MAX}\{l_{ij}\} = \text{MAX}\{f(I_{ij})\}$. The Multiple Instance Learning is to find a mapping function \hat{f} with best approximation to function f given a training set $B = \{B_i\}$ and their corresponding labels $L = \{L_i\}$

($i=1, \dots, n$). The corresponding approximation of f_{ML} is $\hat{f}_{ML}(B_i) = \text{MAX}_j \{\hat{f}(I_{ij})\}$.

In our framework, the Minimum Square Error (MSE) criterion is adopted, i.e., we try to find the function \hat{f} that minimizes

$$SE = \sum_{i=1}^n (L_i - \hat{f}_{ML}(B_i))^2 = \sum_{i=1}^n (L_i - \text{MAX}_j \{\hat{f}(I_{ij})\})^2 \quad (1)$$

Let $\gamma = \{\gamma_k\}$ ($k=1, \dots, N$) denote the N parameters of the function f (where N is the number of parameters), the Multiple Instance Learning problem is transformed to the following unconstrained optimization problem:

$$\hat{\gamma} = \arg \text{Min}_{\gamma} \sum_{i=1}^n (L_i - \text{MAX}_j \{\hat{f}(I_{ij})\})^2 \quad (2)$$

One class of the unconstrained optimization methods is the gradient search method such as steepest descent method, Newton method, Quasi-Newton method and Back-propagation (BP) learning method in the Multilayer Feed-Forward Neural Network. To apply those gradient-based methods, the differentiation of the target optimization function needs to be calculated. In our Multiple Instance Learning framework, we need to calculate the differentiation of the function $E = (L_i - \text{MAX}_j \{\hat{f}(I_{ij})\})^2$. In order to do that, the differentiation of the MAX function needs to be calculated first.

3.2 Differentiation of the MAX Function

As mentioned in [9], the differentiation of the MAX function results in a ‘pointer’ that specifies the source of the maximum. Let

$$y = \text{MAX}(x_1, x_2, \dots, x_n) = \sum_{i=1}^n x_i \prod_{j \neq i} U(x_i - x_j), \quad (3)$$

where $U(\cdot)$ is a unit step function, i.e., $U(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$

The differentiation of the MAX function can be written as:

$$\frac{\partial y}{\partial x_i} = \prod_{j \neq i} U(x_i - x_j) = \begin{cases} 1 & \text{if } x_i \text{ is maximum} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

3.3 Differentiation of the Target Optimization Function

Equation (4) provides a way to differentiate the MAX function. In order to use the gradient-based search method to solve Equation (2), we need to further calculate

the differentiation of the function $E = (L_i - \text{MAX}_j \{\hat{f}(I_{ij})\})^2$ on the parameters $\gamma = \{\gamma_k\}$ of function \hat{f} . The first partial derivative is as follows:

$$\begin{aligned} \frac{\partial E}{\partial \gamma_k} &= \frac{\partial (L_i - \text{MAX}_j \{\hat{f}(I_{ij})\})^2}{\partial \gamma_k} \\ &= 2 (\text{MAX}_j \{\hat{f}(I_{ij})\} - L_i) \times \frac{\partial \text{MAX}_j \{\hat{f}(I_{ij})\}}{\partial \gamma_k} \\ &= 2 (\text{MAX}_j \{\hat{f}(I_{ij})\} - L_i) \\ &\quad \times \sum_{j=1}^{m_j} \left(\frac{\partial \text{MAX}_j \{\hat{f}(I_{ij})\}}{\partial \hat{f}(I_{ij})} \times \frac{\partial \{\hat{f}(I_{ij})\}}{\partial \gamma_k} \right) \end{aligned} \quad (5)$$

Suppose the s^{th} instance of bag B_i has the maximum value, i.e., $\hat{f}(I_{is}) = \text{MAX}_j \{\hat{f}(I_{ij})\}$. According to Equation (4),

Equation (5) can be written as:

$$\begin{aligned} \frac{\partial E}{\partial \gamma_k} &= 2 (\hat{f}(I_{is}) - L_i) \times \sum_{j=1}^{m_j} \left(\frac{\partial \text{MAX}_j \{\hat{f}(I_{ij})\}}{\partial \hat{f}(I_{ij})} \times \frac{\partial \{\hat{f}(I_{ij})\}}{\partial \gamma_k} \right) \\ &= 2 (\hat{f}(I_{is}) - L_i) \times \frac{\partial \{\hat{f}(I_{is})\}}{\partial \gamma_k} = \frac{\partial (L_i - \hat{f}(I_{is}))^2}{\partial \gamma_k} \end{aligned} \quad (6)$$

Furthermore, the n^{th} derivative of the target optimization function E can be written as

$$\frac{\partial^n E}{\partial \gamma_k^n} = \frac{\partial^n (L_i - \text{MAX}_j \{\hat{f}(I_{ij})\})^2}{\partial \gamma_k^n} = \frac{\partial^n (L_i - \hat{f}(I_{is}))^2}{\partial \gamma_k^n} \quad (7)$$

and the mixed partial derivation of function E can be written as:

$$\begin{aligned} \frac{\partial^{(\sum_k n_k)} E}{\prod_k \partial \gamma_k^{n_k}} &= \frac{\partial^{(\sum_k n_k)} (L_i - \text{MAX}_j \{\hat{f}(I_{ij})\})^2}{\prod_k \partial \gamma_k^{n_k}} \\ &= \frac{\partial^{(\sum_k n_k)} (L_i - \hat{f}(I_{is}))^2}{\prod_k \partial \gamma_k^{n_k}} \end{aligned} \quad (8)$$

3.4 Multiple Instance Learning to Traditional Supervised Learning

Similar to the analysis on Multiple Instance Learning problem in Section 3.1, the traditional supervised learning problem can also be converted to an unconstrained optimization problem as shown in Equation (9).

$$\bar{\gamma} = \arg \text{Min}_{\gamma} \sum_{i=1}^n (L_i - \{\hat{f}(O_i)\})^2 \quad (9)$$

The partial derivative and mixed partial derivative of the function $(L_i - \hat{f}(O_i))^2$ are shown in Equations (10) and (11), respectively.

$$\frac{\partial^n (L_i - \hat{f}(O_i))^2}{\partial \gamma_k^n} \quad (10)$$

$$\frac{\partial^{(\sum_k n_k)} (L_i - \hat{f}(O_i))^2}{\prod_k \partial \gamma_k^{n_k}} \quad (11)$$

Notice that Equation (10) is the same as the right side of Equation (7), and Equation (11) is the same as the right side of Equation (8) except that O_i in Equations (10) and (11) represents an object while I_{is} in Equations (7) and (8) represents an instance with the maximum label in bag B_i . This similarity provides us an easy way to transform Multiple Instance Learning to the traditional supervised learning.

The steps of transformation are as follows:

1. For each bag $B_i (i=1, \dots, n)$ in the training set, calculate the label of each instance I_{ij} belonging to it.
2. Select the instance with maximum label in each bag B_i . Let I_{is} denote the instance with the maximum label in bag B_i .
3. Construct a set of objects $\{O_i\} (i=1, \dots, n)$ using all the instances I_{is} where $O_i = I_{is}$.
4. For each object O_i , construct a label L_{O_i} that is actually the label of bag B_i .
5. The Multiple Instance Learning problem with the input $\langle \{B_i\}, \{L_i\} \rangle$ is converted to the traditional supervised learning problem with the input $\langle \{O_i\}, \{L_{O_i}\} \rangle$.

After this transformation, the gradient-based search methods used in the traditional supervised learning such as the steepest descent method can be applied to Multiple Instance Learning.

Despite the above transformation from Multiple Instance Learning to the traditional supervised learning, there still exists a major difference between Multiple Instance Learning and traditional supervised learning. In the traditional supervised learning, the training set is static and usually does not change during the learning procedure. However, in the transformed version of Multiple Instance Learning, the training set may change

during the learning procedure. The reason is that the instance with the maximum label in each bag may change with the update of the approximated function \hat{f} during the learning procedure and therefore the training set constructed along with the aforementioned transformation may change during the learning procedure. In spite of such a dynamic characteristic of the training set, the fundamental learning method remains the same. The following is the pseudo code describing our Multiple Instance Learning framework.

MIL(B, L)

Input: $B = \{B_i\} (i=1, \dots, n)$ is the set of n bags in the training set.

$L = \{L_i\} (i=1, \dots, n)$ is the set of labels where L_i is the label of bag B_i .

Output: $\gamma = \{\gamma_k\} (k=1, \dots, N)$ is the set of parameters of the mapping function \hat{f} where N is the number of parameters.

- 1 Set initial values to parameters γ_k in γ .
- 2 If the stop criterion has not been met, go to step 3; else return the parameter set γ of function \hat{f} .
- /* The stop criterion can be based on MSE or the number of iterations. */
- 3 Transform Multiple Instance Learning to traditional supervised learning using the method described in this section.
- 4 Apply the gradient-based search method in traditional supervised learning to update the parameters in γ .
- 5 Go to Step 2.

Obviously, the convergence of our Multiple Instance Learning framework depends on what kind of gradient-based search method is applied at Step 4. Actually, it has the same convergence property as the gradient-based search method applied

4. IMAGE RETRIEVAL USING RELEVANCE FEEDBACK AND MULTIPLE INSTANCE LEARNING

In a CBIR system, the most common way is ‘Query-by-Example’ which means the user submits a query example (image) and the CBIR system retrieves the images that are most similar to the query image from the image database. However, in many cases, when a user submits a query image, what the user really interested in is just a region of the image. The image retrieval system proposed by [5] first segments each image into a couple of regions, and then allows the user to specify the region of interest on

the segmented query image. Unlike the Blobworld system, we use the user's feedback and Multiple Instance Learning to automatically capture the user-interested region during the query refining process. Another advantage of our method is that the underlying mapping between the local visual feature vector of that region and the user's high-level concept can be progressively discovered through the feedback and learning procedure.

In [18], Multiple Instance Learning is applied on CBIR. As a necessary step before actual image retrieval, the user has to first submit a set of images as the training examples that are used to learn the user's target concept. However, it is usually difficult for the user to provide such a training set. In our method, the first set of training examples are obtained from the user's feedback on the initial retrieval results. In addition, the user's target concept is refined iteratively during the interactive retrieval process.

It is assumed that user is only interested in one region of an image. In other words, there exists a function $f \in F: S \rightarrow \Psi$ that can roughly map a region of an image to the user's concept. S denotes the image feature vector space of the regions and $\Psi = \{1 \text{ (Positive)}, 0 \text{ (Negative)}\}$ where *Positive* means that the feature vector representing this region meets the user's concept and *Negative* means not. An image is *Positive* if there exists one or more regions in the image that can meet the user's concept. An image is *Negative* if none of the regions can meet the user's concept. Therefore, an image can be viewed as a bag and its regions are the instances of the bag in Multiple Instance Learning scenario. During the image retrieval procedure, the user's feedback can provide the labels (*Positive* or *Negative*) for the retrieved images and the labels are assigned to the individual images, not on individual regions. Thus, the image retrieval task can be viewed as a Multiple Instance Learning task aiming to discover the mapping function f and thus to mine the user's high-level concept from the low-level features.

At the beginning of retrieval, the user only submits a query image, and there are no training examples available, which means the learning method is not applicable at the current stage. Hence, we use the following metric to measure the similarity of two images. Assume Image A consists of n regions and Image B consists of m regions, i.e., $A = \{A_i\}$ ($i = 1, \dots, n$) and $B = \{B_j\}$ ($j = 1, \dots, m$), where A_i is a region of Image A and B_j is a region of Image B . The distance (difference) between Images A and B is defined as:

$$D(A, B) = \underset{1 \leq i \leq n, 1 \leq j \leq m}{\text{Min}} \left\{ \|A_i - B_j\| \right\} \quad (12)$$

where $\|A_i - B_j\|$ is the Euclidean distance between two feature vectors of region A_i and B_j . The larger the $D(A, B)$, the less the similarity between Images A and B . This similarity metric implies that the similarity between two images is decided by the maximum similarity between any two regions of these two images.

Upon the first round of retrieving those "most similar" images, according to Equation (12), the users can give their feedbacks by labeling each retrieved image as *Positive* or *Negative*. Based on the user feedbacks, a set of training examples $\{B_+, B_-\}$ can be constructed where B_+ consists of all the Positive bags (i.e., the images the user assigns Positive labels) and B_- consists of all the Negative bags (i.e., the images the user assigns Negative labels). Given the training examples $\{B_+, B_-\}$, our Multiple Instance Learning framework can be applied to discover the mapping function f in a progressive way and thus can mine the user's high-level concept.

The feedback and learning are performed iteratively. Moreover, during the feedback and learning process, the capturing of user's high-level concept is refined until the user satisfies. At that time, the query process can be terminated by the user.

5. EXPERIMENTS AND RESULTS

In this section, the experimental setup and the experimental results are presented.

5.1 Image Repository

We created our own image repository using images from the Corel image library. There are 2,500 images collected from various categories for our testing purpose.

5.2 Image Processing Techniques

To apply Multiple Instance Learning on mining users' concept patterns, we assume that the user is only interested in a specific region of the query image. Therefore, we first need to perform image segmentation. The automatic segmentation method proposed in the Blobworld system [5] is used in our system. The joint distribution of the color, texture and location features is modeled using a mixture of Gaussian. The Expectation-Maximization (EM) method is used to estimate the parameters of the Gaussian Mixture model and Minimum Description Length (MDL) principle is used to select the best number of components in Gaussian Mixture model. The color, texture, shape and location characteristics of each region are extracted after image segmentation. Thus, each region is represented by a low-level feature vector. In our experiments, we used three texture features, three color features and two shape features as the representation of an image segment. Therefore, for each

bag (image), the number of its instances (regions) is the number of regions within that image, and each instance has eight features.

5.3 Neural Network Techniques

In our experiments, a three-layer Feed-Forward Neural Network is used as the function f to map an image region (including those eight low-level texture, color and shape features) into the user's high-level concept. By taking the three-layer Feed-Forward Neural Network as the mapping function \hat{f} and the back-propagation (BP) learning algorithm as the gradient-based search method in our Multiple Instance Learning framework, the neural network parameters such as the weights of all connections and biases of neurons are the parameters in γ that we want to learn (search). Specifically, the input layer has eight neurons with each of them corresponding to one low-level image feature. The output layer has only one neuron and its output indicates the extent to which an image segment meets the user's concept. The number of neurons at the hidden layer is experimentally set to eight. The biases to all the neurons are set to zero, and the used activation function in the neuron is Sigmoid Function. The BP learning method was applied with learning rate 0.1 and no momentum. The initial weights of the connections in the network are randomly set with relatively small values. The termination condition of the BP algorithm is based on $|MSE^{(k)} - MSE^{(k-1)}| < \alpha \times MSE^{(k-1)}$, where $MSE^{(k)}$ denotes the MSE at the k^{th} iteration and α is a small constant. In our experiments, α is set to 0.005.

5.4 CBIR System Description

Based on the proposed framework, we have constructed a content-based image retrieval system. Figure 2 shows the interface of this system. As can be seen from this figure, the query image is the image at the top-left corner. The user can press the 'Get' button to select the query image and press the 'Query' button to perform a query. The query results are listed from top left to bottom right in decreasing order of their similarities to the query image. The user can use the pull down list under an image to input his/her feedback on that image (Negative or Positive). After the feedback, the user can carry out the next query. The user's concept is then learned by the system in a progressive way through the user feedback, and the refined query will return a new collection of the matching images to the user.

5.5 Experimental Results

A number of experiments are conducted to test our proposed framework. Usually, it converges after 6 iterations of the user feedbacks. Also, in many cases, the user's most interested region of the query image can be

discovered, and therefore the query performance can be improved.

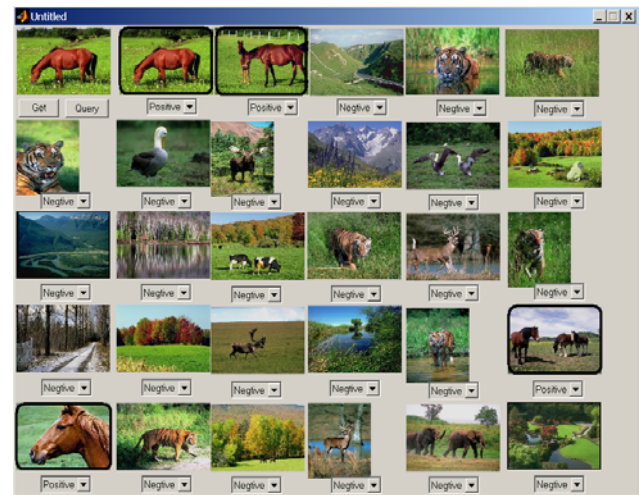


Figure 2. The interface of the proposed CBIR system and query results by using a simple distance-based metric of image similarity



Figure 3. The query results after 5 iterations of user feedback

As shown in Figure 2, there is one horse on the lawn in the query image. Assume the horse object (not the lawn) is what the user is really interested in. Figure 3 shows the initial retrieval results using a simple distance-based metric of image similarity according to Equation (12). As can be seen from this figure, many retrieved images contain lawns or green mountains without any animal object in them. The reason why they are considered more similar to the query image is that they have regions (e.g., lawn) very similar to the lawn region of the query image. However, what the user really needs are the images with the horse object in them. By integrating the user's feedback with Multiple Instance Learning, the proposed

CBIR system can solve the above problem since the user can provide his/her relevant feedback to the system by labeling each image as Positive or Negative. In Figure 2, those images with bounding boxes are labeled Positive, while the others are labeled Negative by the user. Such feedback information is then fed into the Multiple Instance Learning method to discover user's real interest and thus capture the user's high-level concept. Figure 3 shows the query results after 5 iterations of user feedback. The image repository includes eight images with the horse object in them. In addition to the query image, all the remaining seven images are successfully retrieved by the system. Especially, all of them have higher ranks than other retrieved images. Another interesting result is that some of the retrieved images, such as the sunset images, have been retrieved because of their similarity in color to the horse region of the query image. On the other hand, all the images with the pure lawn or the green mountain are filtered out during the feedback and learning procedure. Therefore, this example illustrates that our proposed framework is effective in identifying the user's specific intention and thus can mine the user's high-level concepts.

6. CONCLUSIONS

In this paper, we presented a multimedia data mining framework to discover user's high-level concepts from low-level image features using Relevance Feedback and Multiple Instance Learning. Relevant Feedback provides a way to obtain the subjectivity of the user's high-level vision concepts, and Multiple Instance Learning enables the automatic learning of the user's high-level concepts. Especially, Multiple Instance Learning can capture the user's specific interest in some region of an image and thus can discover user's high-level concepts more precisely. In order to test the performance of the proposed framework, a content-based image retrieval (CBIR) system using Relevant Feedback and Multiple Instance Learning was developed and several experiments were conducted. The experimental results demonstrate the effectiveness of our framework.

ACKNOWLEDGMENT

Shu-Ching Chen gratefully acknowledges the support received from the National Science Foundation through grant CDA-9711582 at Florida International University.

REFERENCES

1. Aksoy, S., and Haralick, R.M. A Weighted Distance Approach to Relevance Feedback. *Proceedings of the International Conference on Pattern Recognition (ICPR00)*.
2. Andrews, S., Hofmann, T., and Tsochantaridis, I. Multiple Instance Learning with Generalized Support Vector Machines. *The Learning Workshop*. (Snowbird, Utah, 2-5 Apr. 2002).
3. Auer, P. On Learning From Multi-instance Examples: Empirical Evaluation of a Theoretical Approach. *Proc. of 14th International Conference on Machine Learning*. (San Francisco, CA), 21-29.
4. Buckley, C., Singhal, A., Miltra, M. New Retrieval Approaches Using SMART: TREC4. *Text Retrieval Conference, Sponsored by National Institute of Standard and Technology and Advanced Research Projects Agency*. (Nov. 1995).
5. Carson, C., Belongie, S., Greenspan, H., and Malik, J. Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying. Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, submitted to *PAMI*. (Available at: <http://elib.cs.berkeley.edu/carson/papers/pami.html>).
6. Chang, C.-H. and Hsu, C.-C. Enabling Concept-Based Relevance Feedback for Information Retrieval on the WWW. *IEEE Transactions on Knowledge and Data Engineering*, 11(4), 595-609.
7. Dietterich, T.G., Lathrop, R. H., and Lozano-Perez, T. Solving the Multiple-Instance Problem with Axis-Parallel Rectangles. *Artificial Intelligence Journal*, 89(1-2), 31-71.
8. Lu, Y., Hu, C.H., Zhu, X.Q., Zhang, H.J., and Yang, Q. A Unified Framework for Semantics and Feature Based Relevance Feedback in Image Retrieval Systems. *ACM Multimedia*. (2000), 31-37.
9. Marks II, R.J., Oh, S., Arabshahi, P., Caudell, T.P., Choi, J.J., and Song, B.G. Steepest Descent Adaptation of Min-Max Fuzzy If-Then Rules. *In Proc. IEEE/INNS International Conference on Neural Networks*. (Beijing, China, Nov. 1992).
10. Maron, O., and Lozano-Perez, T.. Multiple-Instance A Framework for Multiple-Instance Learning. *In Advances in Neural Information Processing System 10*. Cambridge, MA, MIT Press, 1998.
11. Ramon, J., and De Raedt, L. Multi-Instance Neural Networks," *ICML 2000 Workshop on Attribute-value and Relational Learning*. (2000).
12. Ray, S., and Page, D. Multiple-Instance Regression. *Proc. Of 18th International Conference on Machine Learning*. (San Francisco, CA), 425-432.
13. Rui, Y., Huang, T.S., Mehrotra, S. Content-based image retrieval with relevance feedback in MARS. *Proceedings of the 1997 International Conference on Image Processing (ICIP '97)* (3-Volume Set).

14. Rui, Y., and Huang, T.S. Optimizing Learning In Image Retrieval. *Proc. of IEEE Intl. Conf on Computer Vision and Pattern Recognition (CVPR00)*. (Hilton Head, SC, Jun. 2000), 236-243.
15. Wang, J., and Zucker, J.-D. Solving the Multiple-Instance Learning Problem: A Lazy Learning Approach. *Proc. Of 17th International Conference on Machine Learning*. (San Francisco, CA), 1119-1125.
16. Yang, C., and Lozano-Pérez, T. Image Database Retrieval with Multiple-Instance Learning Techniques. *Proceedings of the 16th International Conference on Data Engineering*. (2000), 233-243.
17. Zhang, Q., and Goldman, S.A. EM-DD: An Improved Multiple-Instance Learning Technique. *Advances in Neural Information Processing Systems (NIPS 2002)*. To be published.
18. Zhang, Q., Goldman, S.A., Yu, W. and Fritts, J. Content-Based Image Retrieval Using Multiple-Instance Learning. *The Nineteenth International Conference on Machine Learning*. To be published, (Jul. 2002).
19. Zucker, J.-D., and Chevalere, Y. Solving Multiple-instance and Multiple-part Learning Problems with Decision Trees and Decision Rules. Application to the Mutagenesis Problem. *14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI 2001*. (Ottawa, Canada, 7-9 Jun. 2001), 204-214.

Author Index

Maria-Luiza Antonie	62
Babitha Bandi	1
Ana B. Benitez	39
Nadia Bianchi-Berthouze	93
Casey Breen	51
Shih-Fu Chang	39
Shu-Ching Chen	100
Mariana Ciucu	30
Alexandru Coman	62
Mihai Datcu	11, 30
David Feng	70
Anatole V. Gershman	76
Sadiye Guler	83
Tomofumi Hayashi	93
Patrick Heas	30
Xin Huang	100
William Jockheck	19
Latifur Khan	51
Junghwan Oh	1
Amal Perera	19
William Perrizo	19
Valery A. Petrushin	76
Ian Pushee	83
Dongmei Ren	19
Klaus Seidel	11
Mei-Ling Shyu	100
Pramod K. Singh.....	70
Simeon J. Simoff.....	70
James C. Tilton	30
Lei Wang	51
Gang Wei	76
Weihua Wu	19
Osmar R. Zaiane	62
Chengcui Zhang	100
Yi Zhang	19