

Video Cataloguing and Browsing

Jeff E. Tandianus¹, Andrias Chandra², Jesse S. Jin³

^{1,2,3} School of Computer Science and Engineering, University of New South Wales

³ Department of Computer Science, University of Sydney

{jtan, andriasc, jesse}@cse.unsw.edu.au

Abstract

Videos contain rich information. Until recently, information within a video had been largely left under-utilized, with fast forward/rewind as the most popular method of accessing video content. This is no longer sufficient however, as more and more users demanded the flexibility and ability to access video content selectively. Unlike textual information however, video bits do not convey same level of meaning as text do. Therefore, we have to use metadata to describe the structured information within videos.

One project, ViMeta-Vu or Video Metadata Viewer, was undertaken to provide users with such content-based video access. It represented video using fixed three-level hierarchical structure. Users can access and retrieve specific video information through browsing of the metadata

This paper will discuss various ways of improving the existing system. It proposes a more generic hierarchical structure using XML to represent the metadata, which will facilitate additional query methods. This paper also proposes additional query methods utilizing visual features of key frames in the video. The queries implemented include *structural browsing*, *query by feature*, *query by selection* and *query by keyword*.

Keywords: video cataloguing, content-based video access, visual feature extraction.

1. Introduction

Interests in digital video had been increasing steadily as newer technologies become readily available to the general population at a much affordable prices. More and more technologies discovered on how to make a better video, such as the new favourite DVD-technology; makes video to be one of the most popular and accessible media of entertainment.

The increase of availability of digital video had also lead to the needs for better methods of utilizing content of the video data. Unlike other information media such as book (which has textual Information), video data contains much richer information. Unfortunately, this richness in information does not correlate to the ease of utilising the content of the video. Since unlike textual information, raw video pixel bits do not contain meaningful information by itself. Therefore,

information within a video has to be described through its semantic meaning.

Unlike video data, textual information can be easily organised through long established practices. For example: let us consider a book, words are usually grouped together into paragraph, paragraphs into section, sections into chapter, chapters into book and books into volume. There is also a table of content or indexes to facilitate searches for a particular topic or keyword. Video data on the other hand still relies on crude methods of fast forward/rewind in searching for a region of interest. With the immense information stored within a video, these query methods are unacceptable. A more versatile and efficient organisational structure should be available to facilitate more natural ways of browsing or access.

In the past several years, there have been several projects involved with the creation of metadata. Several of such tools will be discussed here. *Corona* is a system used to segment video into shots. This system is a very important first step towards generating video metadata. The shots produced will act as the building blocks upon which larger segments can be created when creating the video hierarchy.

Vimix, or Video Metadata in XML, is a video metadata annotation tool; Vimix receives input as raw video data and segmentation result from *Corona*. In addition, Vimix requires XML schema file, which will determine the type of elements allowed in the video metadata XML file. Moreover, the XML schema file also describes the structure, relationship and constrains of the elements in the XML file.

Supplied with these inputs, Vimix will annotate the video data. Vimix divides the video data into various segments, analogous to video shots in ViMetaVU. However unlike its ViMetaVU counterpart, Vimix allows video shots to be overlap each other and video shots may be grouped together to form a much bigger video segment.

SegLib or Segmentation Library is the utility program used to perform Image Segmentation. The library supports I/O of images file in a number of formats such as BMP, GIF, PNM, BWC and Z-image. Supplied with an image, the system can segment the input image into several regions using Unseeded Region Growing algorithm.

This thesis will use the segmentation algorithm to segments the key frame images into regions, where the visual features from each of this region will be kept in features library.

The major aim of this paper is to improve upon existing system ViMeta-Vu. ViMeta-Vu is a video cataloguing system providing content-based video access using pre-generated metadata.

The first implementation of ViMeta-VU describes the content of a video using three levels hierarchical structure with text file. Such arrangements are sufficient to describe the relationships of low-level information of the video data. But when it comes to a much more complex relationship, the existing hierarchical structure proves to be insufficient. Therefore, a better scheme is needed to represent information within metadata.

Note that it is not the aim of this thesis to develop new ways of creating video metadata. Rather, this paper will use existing metadata creation process. A very important work affecting this thesis is ViMix (Video Metadata in XML), a first attempt by a UNSW student to describe the content of a video data using XML format.

Using existing segLib library, this thesis will implement a query by feature through the use of texture analysis. This method will start by creating a library of visual features using key frames from all the video segments. Here, key frame refers to the image frame with visual features most closely resembling the video segment it represents. Then, clustering ISODATA algorithm will be used to create an indexing tree out of the feature library. Each branch of the tree will only contain features with similar features. Hence, users have the liberty of restricting his/her search to particular branch of the tree characterised by certain visual features.

2. Video Structure and Organization

Raw video data is organized as a linear sequence of frames. Video information itself is composed of visual information delivered as the sequence of frames accompanied by its audio or sound information. In general, visual information can be generalized into spatial and temporal domain.

- *Spatial domain* expresses information contained within a single frame. The information includes any objects or entities identified in the frame and characteristics of such objects/entities including its locations, presence and textural features.
- *Temporal domain* expresses information through a sequence of frames. Such information is usually derived from activity of objects/entities identified in the spatial domain. Examples of such information include walking man, moving car or singing bird.

Most information in video came from the temporal domain. The existing system attempted to encompass information from both domains using following structure.

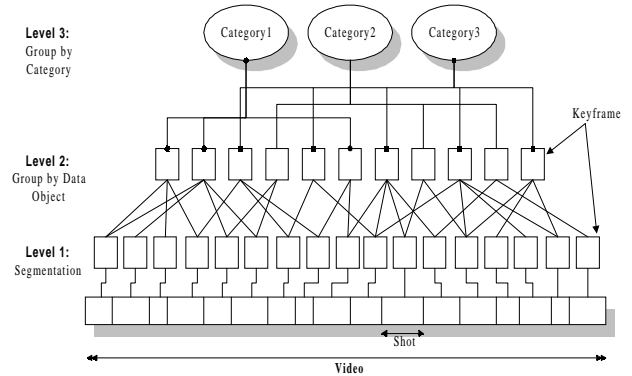


Figure 1: Existing Video metadata structure

The existing video structure and organization is quite basic. The entire video structure was organised in the first level of the metadata structure depicted in Figure 1 above. Here, linear sequences of frames from raw video are being segmented into non-overlapping video shots before they were being referenced by second level entities.

Such organization though simple and elegant has several noticeable drawbacks. First of all, no overlapping is allowed in the video structure. This drawback can be demonstrated using the following example:

The video has two perceptual entities A and B. entity A appears at shot 1 to 3, while entity B appears in shot 1 to 4. Under existing metadata, each entity has to separately reference the video shot that it appeared in.

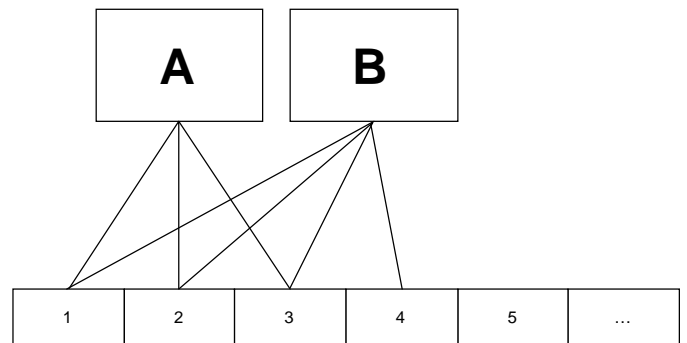


Figure 2: Non-overlapping shots referenced by perceptual entities

A better way is allow shots to overlap and form higher-level video segment. In this example, two new video segments are created, with entity A and B referencing only the higher-level segment. Such organization is desirable when video shots do not have any significant meaning existing on its own. There are cases when it makes more sense to group video shots into bigger segments.

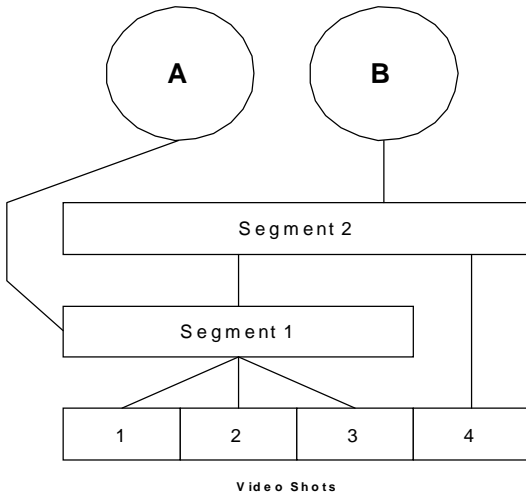


Figure 3: More meaningful organization

2.1 Video Hierarchy

Video shots and its larger segment will from now on referred to as video segment. The video hierarchy presented below is essentially a Directed Acyclic Graph (DAG). Each node below represents a segment. Additionally, the structure will allow overlapping.

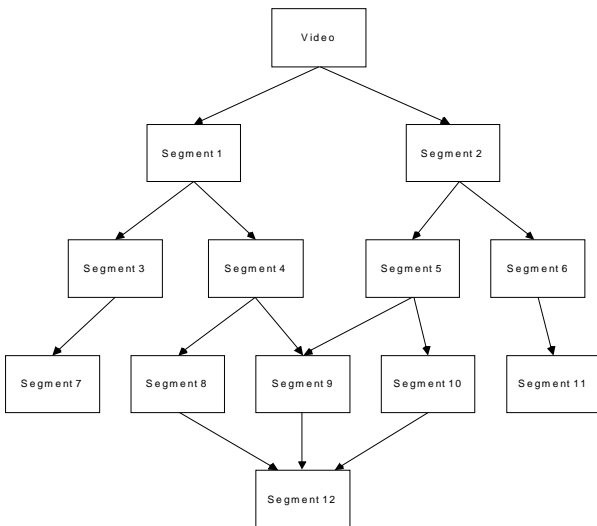


Figure 4: Video Hierarchy

Here, the arrow represents the relationship ‘is parent of’ and the arrow pointed to the child. Here, parent segment may have multiple child segments. Child segments may have multiple parent segments as well. The segments at the lowest level will correspond to the video shots obtained from the shot segmentation algorithm.

There is however one constraint to be observed here. Any child segments must have its boundary defined within the range of its parent segment. This can be modelled by the following relationship:

$$\text{Parent}_{\text{start}} \leq \text{Child}_{\text{start}} \leq \text{Child}_{\text{end}} \leq \text{Parent}_{\text{end}}$$

2.2 Video Entity Structure

After modelling the video structure, the next step is to model entities that may be found in various levels of the cinematic codifications. Here, entity refers to a single or a grouping of entities, objects or concepts that have semantic meaning in the video. Similar to video structure, each entity may have several child entities with the lowest level in the structure corresponds to video key. A *video key* is a collection of attributes or collection of other keys that when put together will uniquely identify a concept or entity that can be found in the video.

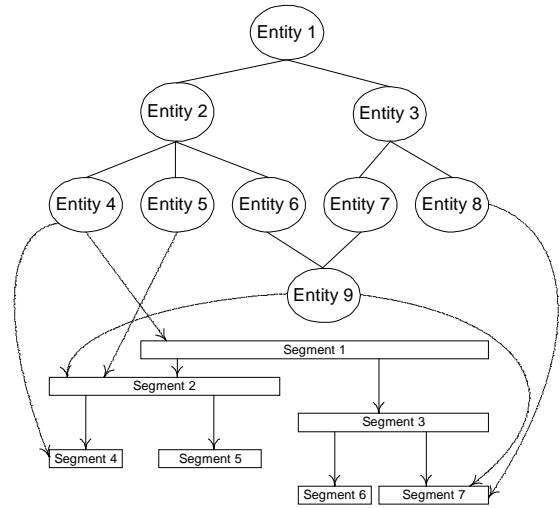


Figure 5: Entity-Segment Associations

Only video keys may be annotated with any video segments. This constraint is here because video key will act like second level objects in metadata depicted in Figure 1. It does not make sense to annotate any higher-level objects with video segments because any associations to the video hierarchy should have been taken care of by the associations made by the video keys.

3. Visual Feature Analysis

This section will explain the visual feature analysis employed in implementing the additional query methods. The additional query methods: Query by Feature, Query by Selection and Structural Browsing are implemented using two analyses. The first analysis, Image Texture analysis describes the texture of image regions through the use of a modified Gabor filter design. The second analysis is based on HSV colour space. The second analysis will calculate the colour histogram for every region in image. Each region will be represented with the components with maximum histogram value. The last section will explain the clustering algorithm used.

3.1 Image Segmentation

Prior to image feature analysis, the input image has to be segmented into several significant regions. Here, each region will represent neighbourhood of pixels, which share same characteristics. In the following example, the input image has been segmented into several regions that are of interests to us. Here, we can see regions representing the face, background and body.

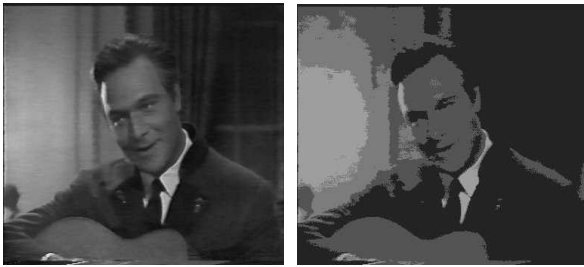


Figure 6: Input image and its segmentation result

For the purpose of this thesis, Unseeded Region Growing algorithm will be used to perform the image segmentation. The algorithm had already been implemented and it is part of the segLib C library that was discussed in section 1.

3.2 Texture Analysis

The texture analysis for this thesis is going to be implemented using Gabor filter approach. Under this scheme, FFT (Fast Fourier Transform) will be applied on the source image before convolving it with a set of Gabor filters with user specified parameters. Image features will then be extracted from the filter output using some statistical methods. One common approach is to represent the features as mean or standard deviation of the filtered image. For this implementation, statistical mean will be applied.

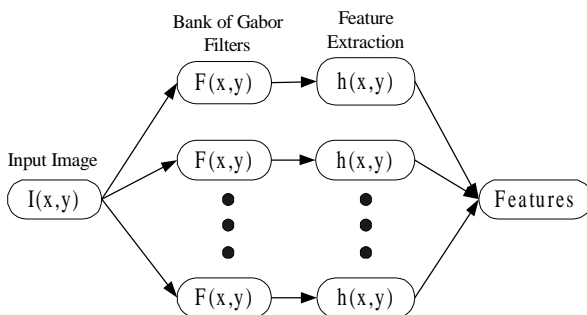


Figure 7: A paradigm of using Gabor filter for image feature extraction

This approach requires applying FFT (Fast Fourier Transform) on the source image and the Gabor filter before convolving them together. However, we know that FFT and convolution operations are time consuming and they will only get worse as the size of the input image increases. Hence, a better approach is needed to reduce the time complexity.

For the requirements of the thesis, we require a design that is faster compared to the traditional approach. This is crucial because each image will be segmented into regions and selected regions from each image will have to be analysed separately. This prompted us to look for an alternative approach that will yield the same results, while minimizing the time complexity factor.

One problem affecting the performance is the size of the input image. As input image size increases, more computations have to be performed in the FFT and convolution operations. For the purpose of our thesis however, we will only be performing the analysis on sub-regions of the image. Such regions will usually be much smaller than the input image. Using this fact, one solution is to resize each region to standard size before performing the analysis on. This standard size is chosen such that it is not too small that it still retains visual information from its original size and not too large that it will give too large a computation time.

Another crucial factor to the time complexity is the need to perform the FFT of the Gabor functions. To reduce the time complexity associated with the Gabor filter FFT operation; we propose to emulate the Gabor function in frequency domain. Here, instead of performing FFT operation for each set of parameter, an approximate rectangular area is used instead. We have found that such approach yields expected result at the fraction of the time.

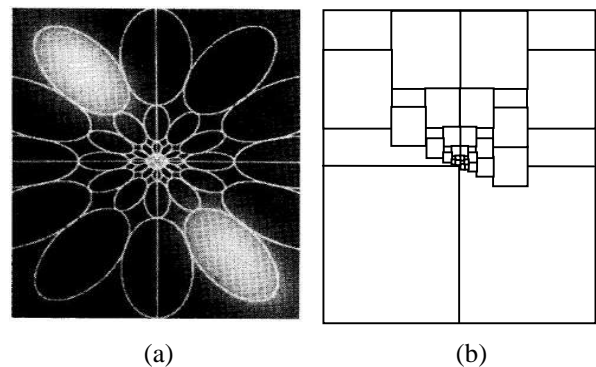


Figure 8: Gabor filters in frequency domain tuned to different frequency and orientation. (a) Traditional Gabor filter. (b) Proposed model.

Image feature will be extracted from regions in the segmented image. For this thesis, we are going to use Gabor filter with template size = 128. Under such size, there are going to be 42 rectangles as shown on Figure 8. Note that each of the rectangles is a Gabor filter with certain user parameter. Using the paradigm shown on Figure 7, the bank of Gabor filters will consist of 42 filters. Then for every filter in the bank, a convolution will be performed between the filter and the Fourier transform of the region. Out of each convolution, the statistical mean of the filtered output will be stored. At the end of the operation, we are going to have a vector

with 42 elements, which will be the image feature for the region.

3.2 Colour Analysis

Colour analysis will also perform image segmentation similar to the texture analysis. As with texture analysis, only features from a limited number of resulting regions will be extracted. For colour analysis, every pixel in the region is converted from its RGB (Red, Green and Blue) colour space to HSV (Hue, Saturation and Value) colour space before the analysis is performed.

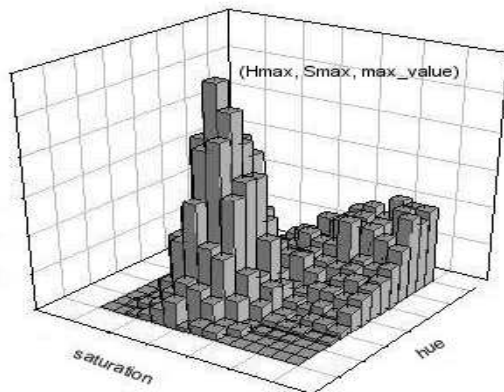


Figure 9: Colour histogram of a region with (Hmax, Smax) producing highest value

In HSV colour space, the histogram of the region will be calculated. From the histogram, hue and saturation component producing highest histogram count value will be used as the image feature of the region. Note that only the hue and saturation component is included in the colour histogram. Value is not included for several reasons. First of all, including all three components will require a three dimensional array that will consume a fair bit of memory. Secondly, enough information is conveyed by the hue and saturation component that it's not necessary to include the value component. In other word, the colour feature extracted will be the colour that is being used the most. This is illustrated in the Figure 9.

3.3 Clustering Algorithm

The next step in the visual feature analysis is to develop ways of organising the visual patterns obtained from both the analysis described above. Visual features obtained above have to be organised in such a way that allow easy access by users. The most common pattern classification techniques are those that are based on distance measurement. Distance measurements are popular because it is the simplest way of establishing similarity between pattern vectors, which can also be considered as points in Euclidean space, is by determining their proximity.

For the purpose of this thesis, we are going to use the ISODATA (Iterative Self-Organising Data Analysis Techniques A) algorithm. ISODATA algorithm

provides pattern classifications when the numbers of pattern classes or clusters are unknown. The algorithm provides mechanisms of splitting and merging existing clusters to obtain the final clusters.

Another advantage of using ISODATA algorithm is because it is an instance of unsupervised classification algorithm. This means that no prior knowledge is required to produce the classification map.

The pseudo-code for the ISODATA algorithm:

For a set of N samples $\{x_1, x_2, \dots, x_N\}$

1. Identify following parameters
 - K = Number of clusters desired
 - θ_N = Threshold on the number of samples in clustering
 - θ_S = Standard deviation parameter
 - θ_C = lumping parameter
 - L = Maximum number of pairs of cluster centres which can be lumped
 - I = maximum number of iteration.

2. Distribute the N samples among existing clusters using following relation:
 - $x \in S_j$ if $\|x - z_j\| < \|x - z_i\|, I = 1, 2, \dots, N_c; i \neq j.$
3. Discard sample subsets with fewer than θ_N members. Go back to step 2 if discard operation was performed. Otherwise, continue.
4. Update cluster centre by setting it equal to the sample mean of corresponding cluster.
5. Compute average distance D_j for every cluster using the relation

$$D_j = 1/N_j \sum \|x - z_j\|, j = 1, 2, \dots, N_c$$

6. Compute overall average distance of samples from their respective cluster.
 - $D = 1/N \sum (N_j * D_j), j = 1, 2, \dots, N_c$
7. Performs merging if following condition is fulfilled. Otherwise, continue.

This is an even iteration or clusters.size() $\geq 2 * K$

Merging operation can be described as following:

- Compute pairwise distance D_{ij} between all cluster centres. Arrange the L smallest distance, which is less than θ_C in ascending order.
 - For each of the pairing, if neither of the cluster centres had been used in lumping in this iteration, merge the two clusters into one cluster.
8. Performs splitting if following condition is fulfilled. Otherwise, continue

$$\text{Cluster's size} < K/2$$

Splitting operation can be described as following:

- Calculate standard deviation vector for $\sigma_j = (\sigma_1, \sigma_2, \dots, \sigma_n)$ for each cluster.

- Find the maximum component of each σ_j and denote it as σ_{jmax} .
- For each cluster S_j , split it into two new clusters if any of the conditions are true:
 - $\sigma_{jmax} > \theta_S$ and $D_j > D$ and $N_j > 2(\theta_N+1)$
 - $\sigma_{jmax} > \theta_S$ and $N_c \leq K/2$

9. If this is the last iteration, exit. Else go to step 2.

4. Browsing Video Metadata

The user interface had been modified to support the changes made to the video metadata. The most noticeable change on the main interface is the replacement of the catalogue windows with hierarchy tree and its corresponding entity window. Here, catalogues are arranged in hierarchical manner with the entity window used to display the content of the catalogue. Similarly, the video segment windows are used to display all video segments referenced by the entities in the catalogue.

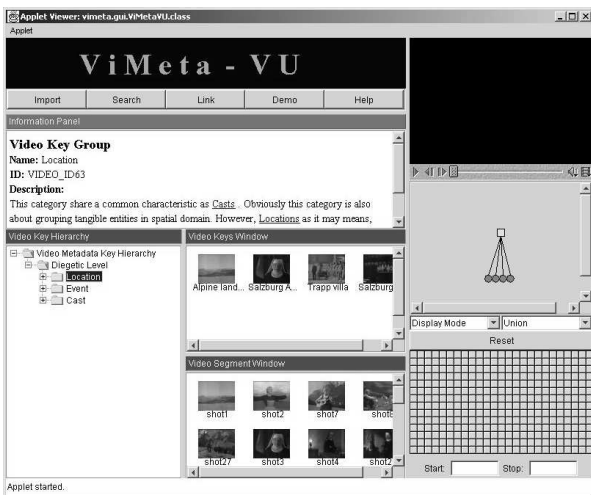


Figure 10: ViMeta-Vu user interface.

Another aim of this project is to improve on the query methods. The existing system can only provides query method based on text annotation of various entities or concepts in the video. However such accesses are deemed to be insufficient considering the wealth of information contained within the video. Though useful, text annotation is simply too subjective in describing the content of video. Such subjectivity is unreliable because two persons describing the same concept may come up with two completely different descriptions. Consequently, it highlighted the needs for more reliable and accurate query methods.

Examples of a much more reliable and accurate query methods are those that are based on visual features. This thesis proposes query methods based on library of such visual features. Using these methods, library of visual features will be created from the key frames used. Note that the library creation is limited only to

the key frames since library based on all frames in the video will simply be too huge and complex to be manageable.

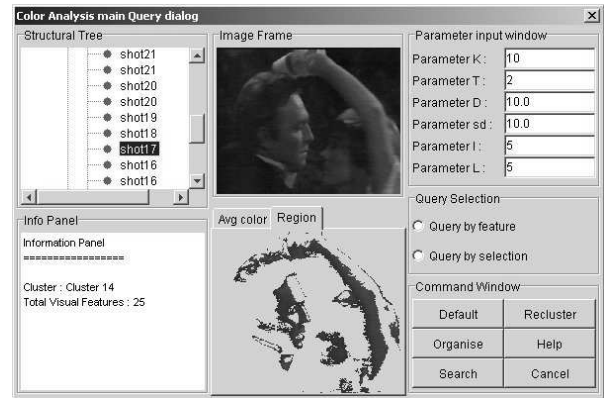


Figure 11: ViMeta-Vu Query interface

Unlike text description, visual features are much more difficult to describe and similar to text annotation, describing dominating visual features are also subjective. The query methods proposed have to address these issues. Based on this, we propose three new query methods in our system: *query by feature*, *query by selection* and *structural browsing*.

In *structural browsing*, the system will create a library of visual features using key frames from all the video segments. Then, clustering ISODATA algorithm will be used to create an indexing tree out of the features library. Using the tree hierarchical structure, users can restrict their selection to certain levels and are able to quickly eliminate branches that are not of interest. The structural tree is shown on top left hand corner of Figure 11.

Another query method is *query by feature*. Through the visual features library, Users will be able to select a visual feature from the features library and perform a query based on the selected visual feature. The elements in the resulting list will have similar characteristics to the input visual feature.

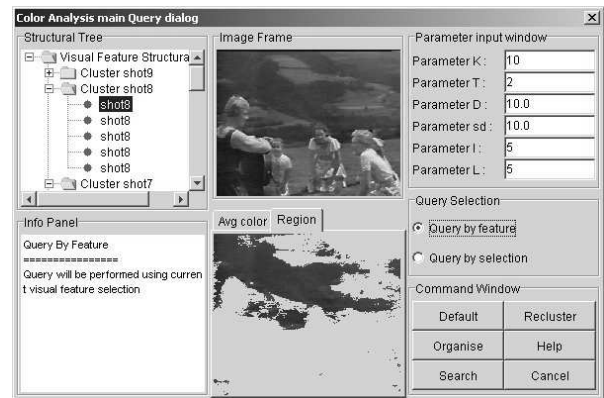


Figure 11: Query by Feature example



Figure 12: Query's results

Query by selection is different to both structural browsing and query by feature in the sense that it is not using the visual features library to base its query on. Rather, user can select a particular key frame image, select a region of interest and perform query based on the visual features from the selected region of interest only.

Note that all of the proposed query methods avoid the need for users to describe the visual features directly. Such design will remove the subjectivity factor out of the system and provide a more accurate ways query methods. The other advantage is that by allowing users to browse, the search engine will adapt to the needs of individual users.

5. Conclusion and Results

The improvement started with the redesigning of the existing ViMeta-Vu system. Redesigning this system using various design patterns make ViMetaVU code more readable and future enhancements easier.

The additional query methods implemented had produced promising results. Colour analysis is significantly more accurate compared to its texture analysis counterpart. One drawback of texture analysis is the need to calculate bounding box of its region before the image feature can be extracted. The bounding box will only be accurate if all points within the region are connected and they are well distributed within the region. However, this is not often the case. When the bounding box is so big compared to the actual region, we are going to introduce error due to non-region pixels in the bounding box.

Colour analysis on the other hand does not rely on any bounding box. Its calculation will strictly be limited to region pixels. An example of query using colour analysis is demonstrated in Figure 11 and 12.

Testings and experiments show encouraging result in query methods based on visual features. However, further works are still needed in utilizing visual features of images.

REFERENCES

- [1] THARMAPALAN, J., ISA W.Y.H.B. W., JIN J. S., and LAMBERT, Tim (1998): Video Cataloguing in Video Archive and Information Retrieval. thesis. University of New South Wales, Sydney.
- [2] YAP, S., and JIN, J. S. (1999): Video Segmentation. thesis. University of New South Wales, Sydney.
- [3] YAO, A., and JIN, J. S. (2000): The Development of a Video Metadata Authoring & Browsing System in XML. thesis. University of New South Wales, Sydney.
- [4] JIN, J. S., LINDLEY, C. A., and FENG, D. D.: *Theory and Practice of Video Cataloguing*.
- [5] PARK, M. (2001): Fast Content-based Image Retrieval Using Hierarchical Visual Features. thesis. University of New South Wales, Sydney.
- [6] TOU, J. T., and GONZALES, R. C. (1974): *Pattern Recognition Principle*. Addison Wesley.
- [7] GONZALES, R. C., and WOODS, R. E. (1993): *Digital Image Processing*. Addison Wesley.
- [8] FOLEY J. D., and VAN DAM, A. (1997): *Computer Graphics Principles and Practice*. Addison Wesley.
- [9] AACH T., KAUP, A., and MESTER R. Local Energy Transforms versus Quadrature Filters, Signal Processing. In *Texture Analysis*.
- [10] DAUGMAN, J. G. (July 1998): Complete Discrete 2-D Gabor transforms by Neural Networks for Image analysis and Compression. *IEEE Transactions on Acoustics, Speech and signal processing, Vol. 36. No. 7*.
- [11] CHEN, C. C., and CHEN, D. C. (1996): Multi Resolucional Gabor Filter in Texture Analysis. *Pattern Recognition letter 17 1069-1076*.
- [12] W3C, Extensible Markup Language (XML) 1.0 recommendation (October 2001): <http://www.w3.org/TR/1998/REC-xml-19980210>.
- [13] W3C, XML Schema Part 0: Primer (October 2001): <http://www.w3.org/TR/xmlschema-0/>
- [14] W3C, XML Schema Part 1: Structures (October 2001): <http://www.w3.org/TR/xmlschema-1/>
- [15] W3C, XML Schema Part 2: Datatypes (October 2001): <http://www.w3.org/TR/xmlschema-2/>
- [16] W3C, Document Object Model (DOM) Level 2 Core specification (October 2001): <http://www.w3.org/TR/DOM-Level-2-Core/>
- [17] Apache XML project (October 2001): <http://xml.apache.org/>
- [18] JGuru (October 2001): <http://www.jguru.com/>
- [19] Java Sun (October 2001): <http://java.sun.com>