

# Automatic Discovery of Categories in Video

---

*Project Proposal*

*Chris Reynia and Michael Branton  
Stetson University*

**Abstract:** As we move further in to the information age, the problem is no longer the lack of availability of information, but rather the ability to find the right piece of information in the overabundance of information available. This is made even more difficult when the piece of information being sought is not text, but a more complicated form of media such as an image or video. We propose a method to help find video based off similarity to other video, as well as automatically classify video in to a genre. We intend to do this by leveraging algorithms for finding interesting frames in video, and then using those frames as still images for image comparison algorithms.

# Contents

- Introduction ..... 3
- Previous Work..... 3
  - Automatic Video Categorization ..... 3
  - Media Categorization..... 4
  - Images from Video ..... 5
  - Image Comparison ..... 6
- Work Done ..... 10
- Proposal ..... 9
- Conclusion..... 12
- Works Cited..... 12

## Introduction

We have reached a point in time where information is both extremely valuable, but also overabundant. It is becoming exceedingly difficult to sift through the large amounts of information to find what you are looking for, especially when dealing with rich media such as images, video, or sound. The complexity of the problem only increases when you have no natural language context or annotation to associate with the media. Because of this, we find particularly interesting the subset of this problem that deals with the comparison and categorization of media using the innate features of that media.

While it is not a trivial problem, the task of comparing images is well identified in the area of Computer Vision and many solutions have been proposed. However, there has been little work in the area of comparing and categorizing video, especially using methods unaided by humans. We think it is especially important to focus on solutions that do not require human input after training, or else the applications of the solution will be greatly limited.

With this in mind, we will propose in this document a solution to computationally categorize a library of video. We will first give an overview of related work in the field. Next we will discuss the current status of work on the project. Then we will explain the proposed plans for continuing the project. Lastly, we will conclude with some final thoughts.

## Previous Work

### Automatic Video Categorization

As mentioned previously, we found very little work already completed in the area of automatic video categorization, however there has been some. In (1), Truong and Dorai extend a study they did on editing effects, motion, and colors used in video to automatically categorize videos. The analysis of these features was inspired by the common use of the features within a genre as a technique to cause certain

emotions within viewers. They then analyzed the trends of each of the features in various genres to help understand similarities, dissimilarities, and confusions between genres.

In (2), Pan and Faloutsos create a tool which analyzes video, still frames, and audio to create a vocabulary describing a video that is similar to a natural language vocabulary. A genre, or class, of video can then be described by a vocabulary, or set of basis functions, found when doing Independent Component Analysis on the features of videos. This is done through a compression method, finding the bases of a class that can be best used to reconstruct a video clip with the smallest error. The videos can then be mined through the vocabulary of each video or through the classes.

## Media Categorization

In a different paper, (3), Pan, et al. describe a generalized method for comparing any type of media. This method creates a “Multi-Media Graph” (MMG) to provide a framework for media categorization abstracted from the actual method of comparison. We will go over in detail how this graph is created and structured later in the paper. The graph can then be traversed using any number of graph traversal algorithms to find coefficients for how related two pieces of media are to each other.

We were immediately attracted to the generalized nature of the solution presented by Pan, et al., as well as its direct applicability to video. One of the attributes of this method that we find to be the most powerful is the independence of the actual attribute comparison from the framework; any metric we implement for measuring a difference between videos can be implemented in to the MMG. With the MMG in mind, it becomes very easy to separate our problem in to two distinct parts: finding descriptive images from video, and then comparing the images found. While neither of these two problems is trivial, they both have a notable research record.

## Images from Video

Finding useful images from a video is a very interesting problem in its own right and has many applications completely separate from the problem we are trying to solve. Because of this, we found two primary motivations for work in this area: finding useful frames for intelligently traversing through video, and creating a descriptive summary of a video using still frames.

A method motivated by the previous is described in (4), in which a continuous still “tapestry” is produced that describes a variable timeline of the video based off a temporal scale (Figure 1). This solution is very interesting as a means of video traversal, but we are primarily interested in their means for selecting frames to include in the tapestry. This is achieved by minimizing a function outlined by Simakov, et al:

$$d_{BDS}(S, T) = \frac{1}{N_S} \sum_{s \in S} \min_{t \in T} D(s, t) + \frac{1}{N_T} \sum_{t \in T} \min_{s \in S} D(s, t)$$

where  $S$  is the source, or original image,  $T$  is the target image, small rectangular image patches  $s$  and  $t$  are sampled from the source and target images, and the number of source and target patches are  $N_S$  and  $N_T$ , respectively. The patches are of fixed size: we use  $7 \times 7$  patches in our implementation. The distance  $D(s; t)$  is the distance in color space between these square patches: we use  $L^2$  distance [Euclidean distance] in RGB space. When retargeting, the source image  $S$  will have different dimensions than the retargeted image  $T$

As a first level optimization, Barnes, et al. use entire frames for the image patches, creating a set of interesting frames. The function is then minimized again, using the set of frames as the source and the patches as the target to create a set of interesting content from the frames.

Figure 1



A different method, (5), which was motivated by creating descriptive summaries that look similar to stained glass windows (Figure 2), takes a slightly different approach to finding interesting portions of video. First, the video is segmented into clips of similar frames using techniques such as color histograms or variations of pixel-wise differences. Next, regions of high importance are found. This is done by arranging the frames in a 3D space (x-y-time) and then creating bounding 3D rectangular boxes around pixels of high velocity in the 3D space. The velocity of a pixel is its change in luminance between



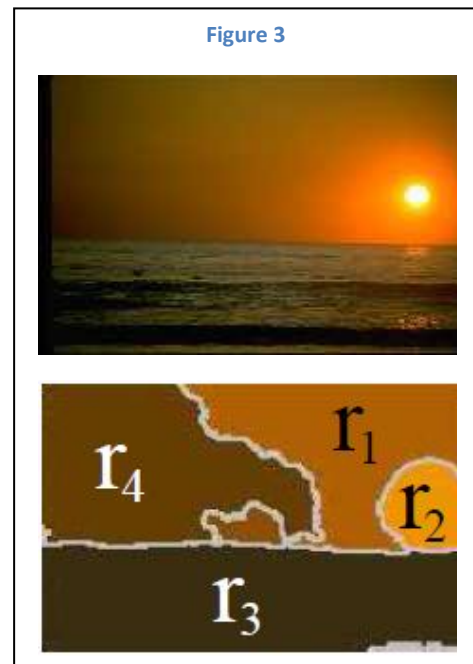
video frames. These bounding boxes can then be used to describe interesting portions of a series of frames, which can then be used to create a descriptive still image summary of a video.

## Image Comparison

Now that we have described some methods for finding useful still frames in a video, we will explore some methods for comparing and categorizing images, and which can be used to compare our useful frames. There is a wide variety of image comparison algorithms, but in our search for different methods we noticed a strong prevalence of semantically based algorithms that tie some sort of vocabulary to images, portions of images, or sets of images. Obviously many low level measures exist for quantifying pixel data between images, but they seem to be trivial enough that there is not very much in

depth research on those methods. Below is a series of interesting methods for image comparison to help set the context for the state of the field as well as present possible method that can be implemented as measures in our solution.

Chen and Wang present a method of image categorization, (6), in which images are represented as bags containing instances. A bag is considered positive if it contains at least one instance, otherwise it is negative. Each image is segmented (Figure 3) and each segment, represented by an instance, is described by a feature vector. The algorithm learns a series of instance prototypes and creates a mapping from bags to points in the bag feature space. Vector machines are then trained in the bag feature space to detect the instance of regions in images. When vector machines are trained, they construct a series of binary separations in a high dimensional space. These separations essentially segment the space, allowing for each segment to be considered an instance in our application. Lastly, these vector machines are used to detect the instance of new regions, by using the feature vectors to see in which instance space the region falls.



Shotton, et al. introduce a new low level feature analysis called texton forests (7). Texton forests are decision trees that act on the pixels of an image to represent semantic textons. Semantic textons are square patches of an image, centered at each pixel with one of four functions applied to it. The semantic textons can then be used to generate both local and global information about the image. The global information is used for categorization, and the local information is used for segmentation.

Li, et al. demonstrate a method of categorizing images through statistical modeling (8). In their approach, a set of training images for each category is used to generate a two dimensional multiresolution hidden Markov model. Images being used for training or categorization are used to generate a set of feature vectors extracted at different resolutions and arranged on a pyramid grid. In categorization, an image's pyramid is run through each of the categorical models, which output a likelihood of a match. The likelihood values are used to find statistical significance of each category for that image, and thus identifying the image as a member of any category that is significant.

Barnard, et al. present a number of different categorization models in (9). They introduce Shi and Malik's normalized cut segmentation algorithm, (10), and explain a set of 40 features generated for each image used in the models. The first model they present is for annotating images and is the "Multi-Modal Hierarchical Aspect Model". In this model, a tree is created from a training set and then traversed based off feature values. The tree is built such that the greater the depth traversed in to the tree, the more specific the annotation. They also present a "Discrete Data Translation" model, which is essentially a one to one translation of features to annotations. A feature space is created and trained using an annotated training set, and regions on the feature space are mapped to particular annotations. An image is then matched to an annotation if its feature vector falls within the region. They go on to explain a few other methods of annotation, as well as discuss the details of properly evaluating automatic annotations.

Wenyin, et al., (10), describe a system for semi-automatically annotating images. In their system a user searches for an image through keywords. The system then returns an image using a search algorithm. The user can give feedback as to the correctness of the keywords' correlation to the returned image. If the user gives positive feedback, annotations that were not already assigned to the image are

added. The system is designed so that any feature based search algorithm can be used, but the initial implementation just used a simple low level feature vector matching algorithm.

## Proposal

Now we will propose a process for the rest of this project. We intend to use MMGs as a framework for solving our problem. We think it will be the most efficient way to allow us to implement a wide range of existing algorithms for frame picking and image comparison, while still providing the flexibility to easily test our own methods as well as combine various methods. With that said, we plan to generate an MMG of videos to be compared as follows. First, we will create object nodes for each piece of video being compared. We will then find important frames from each of these videos and create object nodes of each of the frames as segmentations of the video node. Next, we will run an image segmentation algorithm on each frame, generating a third tier of segmentation nodes, each connected to their respective frames. Lastly we can run applicable processing algorithms on any node in the graph and generate feature nodes for each of these values.

Once we have this graph, we can traverse it to find the correlation between each video node. We can generate a function for the correlation value that causes highly correlated videos to output a lower number and less correlated videos to output higher number. We could then think of this value as an absolute Euclidean distance and the further away one node is from the other, the less related they are. One possible method of utilizing these distances to find the category, or genre, of a video would be to compare the video to archetypal videos, a known representative of each category. This would also allow us to present each video as having some variable relationship with each category; for example, a romantic comedy with some action could potentially be very close to the romance and category archetypes, and a bit further from the action archetype, but far away from all other archetypes. We

could also use this distance to present extremely similar or extremely different videos from one being compared.

Once we have this framework in place, and can produce viable results using simple algorithms for finding feature values, we would like to implement more existing features, as well as begin to experiment with creating our own features.

## Work Done

We will now outline the current status of our work. Because of the generalized nature of (3), we decided this would not only make a good starting point, but could become a framework for implementing other solutions to our problem. Our first goal is to implement an MMG as described by Pan, et al. and verify correctness by finding the correlations in the same COREL image set, using the same feature vectors. We have since implemented an MMG in Java and are in the process of implementing the feature comparisons used to generate feature vectors. To help with this, we have used the matlabcontrol utility, (11), to allow communication between the Java MMG code and any feature analysis that we may choose to implement in MATLAB. Thus far, we utilized this bridge to employ Shi and Malik's normalized cut segmentation MATLAB implementation (12) in our MMG implementation in order to segment images and generate a feature vector made up of the average red, green, and blue values.

As we have started working with the MMGs it has become apparent that the graph based approach provides a very intuitive medium for separating the processing applied to our media. We will briefly outline how a MMG is generated to both demonstrate this intuitiveness as well provide insight in to our implementation. The graph will consist of a series of object nodes and feature nodes. Object nodes represent a piece of media, while feature nodes represent the quantitative value outputted when

a feature function is run on a piece of media. For any given object node, the nodes for features that are evaluated for that piece of media are connected by an undirected edge to the object node. Also, if an

object is segmented so

that a feature can be

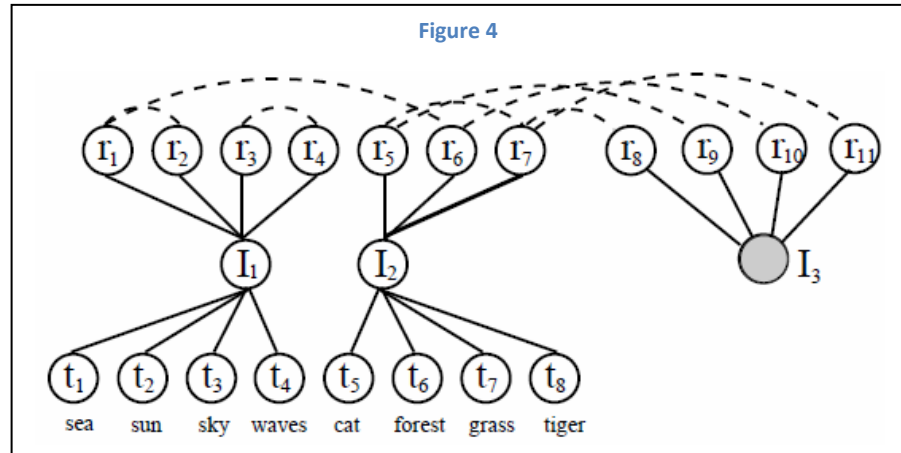
evaluated for the segment,

object nodes are created

for each segment and

attached by an undirected

edge to the object they



were derived from. For each of these segments, the nodes for the features evaluated on the segment

are attached to the segment's object node, not the original object node. Once all of the objects,

segments, and features are established in to a graph, directed edges are created between the feature

nodes that can be compared using some comparison function, depending on the value of the

comparison function run on each pair of feature nodes. Figure 4 gives an example of what an MMG may

look like; In this case,  $I_1$  through  $I_3$  are image nodes;  $r_1$  through  $r_{11}$  are various low level feature nodes,

and  $t_1$  through  $t_8$  are feature nodes describing natural language annotations. The graph is now complete

and can be traversed using a graph traversal algorithm to generate the correlation values between

different pieces of media. In the example outlined in Figure 4, the correlations could then be used to

assign annotations from  $I_1$  and  $I_2$  to  $I_3$ . It is worth noting that Pan, et al. suggest using random walks with

restarts (13) as a graph traversal algorithm. Because of this, we also implemented an algorithm for

generating the correlation values between two nodes that was outlined in (3).

## Conclusion

We are hopeful that this project will mature to be working in a viable form. The largest issue we foresee is the possibility that still frames from a video may not provide enough contexts to successfully identify genres of video. However, because of the abstract nature of the MMG framework, it would require relatively little effort, beyond implementing the algorithms, to add features that evaluate sound or dynamic aspects of the video. If we are able to get results with a high enough accuracy of identification there would be definite applications in recommendation systems, cataloging systems, and even as an end user application.

## Works Cited

1. *Automatic Genre Identification for Content-Based Video Categorization*. **Trong, Ba Tu and Dorai, C.** Barcelona, Spain : s.n., 2000.
2. *VideoCube: a novel tool for video mining and classification*. **Pan, Jia-Yu and Faloutsos, Christos.** Singapore : s.n., 2002.
3. *Automatic Multimedia Cross-modal Correlation Discovery*. **Pan, Jia-Yu, et al.** Seattle, WA : s.n., 2004.
4. *Video Tapestries with Continuous Temporal Zoom*. **Barnes, Connelly, et al.** 2010.
5. *Stained-Glass Visualization for Highly Condensed Video Summaries*. **Chiu, Patrick, Girgensohn, Andreas and Liu, Qiong.** Palo Alto, CA : s.n.
6. *Image Categorization by Learning and Reasoning with Regions*. **Chen, Yixin, Wang, James Z and Geman, Donald.** 2004.
7. *Semantic Texton Forests for Image Categorization and Segmentation*. **Shotton, Jamie, Johnson, Matthew and Cipolla, Roberto.** Anchorage, AK : s.n., 2008.
8. *Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach*. **Li, Jia and Wang, James Z.** 2003.
9. *Matching Words and Pictures*. **Barnard, Kobus, et al.** 2003.
10. **Shi, Jianbo and Malik, Jitendra.** Normalized Cuts and Image Segmentation. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*. 2000. Vol. 22, 8.

11. *Semi-Automatic Image Annotation*. **Wenyin, Liu, et al.** 2001.
12. matlabcontrol. *Google Code*. <http://code.google.com/p/matlabcontrol/>.
13. **Doyle, Pete G. and Snell, J. Laurie.** *Random walks and electric networks*. 2006.