# FAST SCENE SEGMENTATION USING MULTI-LEVEL FEATURE SELECTION

*Yan Liu and John R. Kender*

Department of Computer Science
Columbia University
New York, NY 10027
{liuyan, jrk}@cs.columbia.edu

## ABSTRACT

High time cost is the bottle-neck of video scene segmentation. In this paper we use a heuristic method called Sort-Merge feature selection to construct automatically a hierarchy of small subsets of features that are progressively more useful for segmentation. A novel combination of Fastmap for dimensionality reduction and Mahalanobis distance for likelihood determination is used as induction algorithm. Because these induced feature sets form a hierarchy with increasing classification accuracy, video segments can be segmented and categorized simultaneously in a coarse-fine manner that efficiently and progressively detects and refines their temporal boundaries. We analyze the performance of these methods, and demonstrate them in the domain of long (75 minute) instructional videos.

## 1. INTRODUCTION

The rapid growth and wide application of digital video has led to a significant need for efficient video data management. Temporal video segmentation is an important topic in video understanding.

To reach semantic goal, some machine learning methods such as classification and boosting are introduced for video scene segmentation. But due to the huge volume of video data, the high time complexity is always the bottle-neck for efficient video analysis. Researchers therefore work on speeding up their algorithms; one way is by seeking efficient ways of reducing the dimensionality of the data prior to segmentation.

Vailaya et al [1] and Smeulders et al [2] discuss this problem from the view of image processing and computer vision. They assume that some features, such as color histograms or texture energies, are more sensitive than others, based on the researchers' intuition. They provide theoretical analyses and empirical validations for their choices, but this approach is difficult to extend to other domains where the relationships between features and categories are unclear and changeable.

The heart of this paper is a novel feature selection algorithm, which focuses on selecting representative features in the massive and complex dataset automatically; no manual definition or construction of features is required. This form of learning has received significant attention in the AI literature recently and has been applied to moderately large data sets in applications like text categorization and genomic microarray analysis. Learning research is not often carried out in video domain--although Lew et al [3] used a feature selection method to refine features for stereo image matching. This is because the sheer magnitude of video data has limited the choice and application of existing feature selection algorithm, which have been designed for smaller databases and which run inordinately long even on those.

This paper is organized as follows. Some related work in feature selection is introduced in section 2. Section 3 proposes a fast scene segmentation algorithm using feature selection. Section 4 presents empirical validation of the accuracy and efficiency of algorithm when applied to the particular genre of instructional videos, and validates the algorithm in a second domain. We close the paper with discussion in section 5.

## 2. RELATED WORK

### 2.1. Filter methods and wrapper methods

There appears to be two major approaches to the feature selection problem. The first emphasizes the discovery of any relevant relationship between the features and the concept, whereas the second explicitly seeks a feature subset that minimizes prediction error of the concept. The first is referred to as a filter method, and the second approach is referred to as a wrapper method. In general, wrapper methods attempt to optimize directly the predictor performance so that they can perform better than filter algorithms, but they require more computation time. Seen in this context, this paper proposes a wrapper feature selection method with low time cost.

### 2.2. Feature selection algorithm design and evaluation

Feature selection methods are typically designed and evaluated with respect to the accuracy and cost of their three components: their search algorithm, their statistical relationship method (in the case of filter methods) or their induction algorithm (in the case of wrapper methods), and their evaluation metric (which is simply prediction error in the case of wrapper methods). The dominating cost of any method, however, is that of the search algorithm, since feature selection is fundamentally a question of choosing one specific subset of features from the power set of features. So far, three general kinds of heuristic search algorithms have been used: forward selection, backward elimination, and genetic algorithms.

### 2.3. Sort-Merge feature selection algorithm

Liu and Kender in [4] proposed a Sort-Merge feature selection algorithm, which can exploits several properties unique to video data to induce appropriate but small feature sets in low time cost.

*2.2.1. Sort-Merge search algorithm for feature selection*

Sort-Merge feature selection algorithm combines the features of forward selection, backward elimination, and genetic algorithms. To avoid irrevocable adding or subtracting, it always operates on some representation of the original feature space, so that at each step every feature has an opportunity to impact the selection. To avoid heuristic randomness, at each step a greedy algorithm is used to govern subset formation. Further, the recursive nature of our method provides an additional advantage over existing methods, in that it enables the straightforward creation of near-optimal feature subsets of any or all given cardinalities or accuracies, with little additional work.

The Sort-Merge algorithm can be divided into two parts: the creation of a tree of feature subsets, and the manipulation of the tree to create a feature subset of desired cardinality or accuracy. Each part uses a heuristic greedy method.

Table 1 shows the algorithm of setting up the tree. Figure 1 illustrates a tree with N=256. Table 2 shows the algorithm of cutting the tree based on the application requirement, for example, to create a feature space with exactly r features.

*2.2.2. Induction algorithm for feature selection*

The performance of a wrapper feature selection algorithm not only depends on the search method, but also on the induction algorithm. For our induction method during the course of the learning, we use a novel combination of Fastmap for dimensionality reduction and Mahalanobis maximum likelihood for classification. We refer readers to the literature for a detailed explanation of these methods, but summarize their significance here.

```
Initialize level = 1
        N singleton feature subsets.
While level <  $\log_2 N$
        Induce on every feature subset.
        Sort subsets based on their
        classification accuracy.
        Combine, pair-wise, feature subsets.
```

**Table 1.** Sort-Merge feature selection basic algorithm

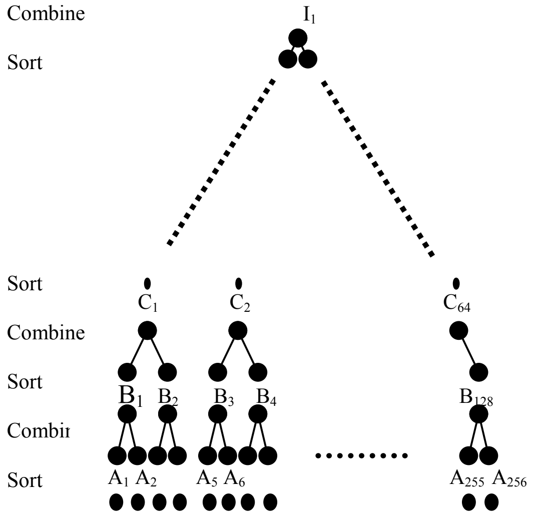


**Figure 1.** Setup of the Sort-Merge feature selection tree

```
Select the leftmost branch of size $2^{\lceil \log 2r \rceil}$.
Initialize cutout = $2^{\lceil \log 2 r \rceil}$ - r.
While cutout >0
        Let branch-size = $2^{\lfloor \log 2 \text{ cutout} \rfloor}$.
        For all remaining branches of this
        size, evaluate the induction result of
        removing those branches individually.
        Remove the branch with best result.
        Let cutout = cutout – branch-size.
```

**Table 2.** Algorithm to select exactly r features from the tree of feature subsets.

The fastmap method proposed in [5] approximates Principal Component Analysis (PCA), with only linear cost in the number of reduced dimensions sought, c, and in the number of features, N. In brief, as defined in statistical texts Duda et al. [6], or in the documentation of

Matlab, the Mahalanobis distance computes the likelihood that a point belongs to a distribution that is modeled as a multidimensional Gaussian with arbitrary covariance.

## 3. SCENE CATEGORIZATION USING MULTI-LEVEL FEATURE SELECTION

We illustrate the framework of fast video segmentation using MPEG-1 instructional videos.

### 3.1. Framework of applying feature selection algorithms

We down-sample the video temporally using only every other I frame (that is, one I frame per second), and we spatially subsample by only using the DC terms of each macroblock of the I frame (consisting of six terms: four luminance DC terms, one from each block and two chrominance DC terms). We therefore do not have to decompress the video. This gives us, for each second of video, 300 macroblocks (15 by 20) of 6 bytes (4 plus 2) of data: 1800 initial dimensions.

Each six-dimensional feature is first placed into its own subset to initialize the Sort-Merge process. Next, using Fastmap, the dimensionality of each feature subset is reduced to a pre-specified small number, c, of dimensions. Then, for each feature subset at this level, using the reduced dimensionality representation, the training frames of the video train the Mahalanobis classifier to classify the test frames of the video.

Next, the classification accuracy of each feature subset is measured. If any subset achieves the user's pre-specified desired accuracy, or if the cardinality of each subset achieves the user's pre-specified desired cardinality, the process stops, and that subset is the desired feature subset. Otherwise, the feature subsets are sorted by accuracy, and the next level of the feature subset hierarchy is formed by merging these subsets pair-wise and in order (see Figure 1).

Lastly, the process repeats again, starting at the Fastmap step. It is clear that at most O(log N) iterations of this Sort-Merge algorithm are necessary to setup the whole tree.

### 3.2. Video segment boundary refinement using multi-level feature selection

We now show how the feature subset hierarchy can be exploited to efficiently refine the boundaries of contiguous video segments with identical labels. The hierarchy enables less work to be done on the segment interiors, and permits a multi-level refinement strategy using more accurate but more costly feature subsets at segment edges.

To illustrate, we select the best 2-feature subset from the 300 features using Sort-Merge feature selection algorithm, and classify each frame of the video into for example four different categories. This is shown as the first line in Figure 2 as $C_2$, $C_1$, $C_4$, etc. The classification tends to have more errors at segment transitions, whether they are abrupt (cuts) or gradual (fades and dissolves). So we devise a multi-level (coarse-to-fine) strategy to more carefully investigate the video wherever a neighborhood of frames shows a lack of consistency of labeling. Note that this will occasionally occur even within the interior of a well-defined segment.

This strategy is governed by several parameters, which vary depending on the number of the successive iterations of refinement. We therefore define a feature subset size $R_i$, which increases with i and therefore increases the classification accuracy, and a neighborhood parameter $L_i$, which remains constant or decreases with i and therefore focuses the attention of the more costly classifier. Further, we define a decision threshold $S_i$, according to:

$$S_i = Pr_{mahal}(C_j) - \sum Pr_{mahal}(C_k) \qquad k = 1, 2 \ldots n \text{ and } k \neq j$$

where Prmahal($C_j$) is the maximum Mahalanobis likelihood among all categories using this feature subset. This threshold ensures that classification is correct and unambiguous.

Figure 2 illustrates three typical cases. Most of the refinements result in the first case: a clarification of the location of the boundary developed by the intial classification of frames. However, in a second case, shown at the transition between $C_1$ and $C_3$, it is possible that an intervening segment of a completely different label is refined such as $C_2$. In the third case, refinement proceeds to using all available features with resolving the labeling of an individual frame sufficiently confidently: this frame is the exact center of a dissolve between two classes.

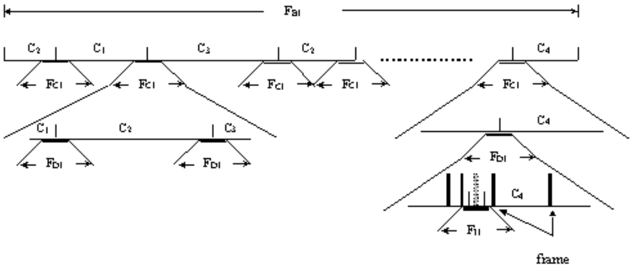
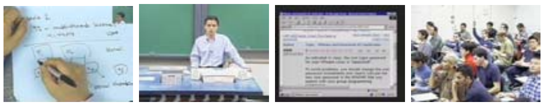


**Figure 2.** Video segment boundary refinement by multi-level feature selection.

## 4. EXPERIMENT

In this section, we illustrate scene categorization of four categories in Figure 3: handwriting, announcement, demo, and discussion on the extended instructional video of 75 minutes duration in MPEG-1 format. Other kinds of videos are in progress. After down-sampling the video temporally and spatially as mentioned in section 3.1, we use about 4700 frames of 300 six-dimensional features as test data and 400 frames distributed over the video and across these four classes as training data. We evaluate the accuracy performance using classification error rate, which is defined as the number of incorrect labeled data compared with the number of all test data.



Handwriting Announcement Demo Discussion

**Figure 3.** Four categories of video scenes

Figure 4 compares the scene categorization accuracy using 30 macro-blocks. For random feature selection, we ran 100 experiments in which 30 features were selected randomly and calculate the mean of error rate. The classification error rate of the Sort-Merge method is not only less than that of hand selection, but also appears to be very stable as the Fastmap dimension varies.

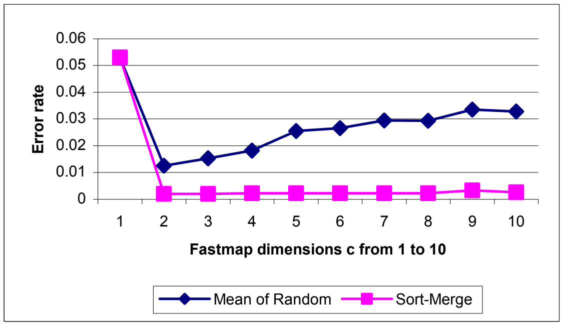


**Figure 4.** Scene segmentation error rate using random feature selection and Sort-Merge feature selection

| Clip | $R_1$=2<br>c=9 | $R_2$=4<br>c=7 | $R_3$=8<br>c=4 | $R_4$=16<br>c=4 | $R_5$=32<br>c=3 |
|---|---|---|---|---|---|
| Re-checked segments | 27 | 27 | 10 | 7 | 5 |
| Fraction of video frames | 100.00% | 6.31% | 7.14% | 2.31% | 1.19% |

**Table 4.** Classification of video clips in a coarse-fine manner using multi-level feature selection algorithm. At iteration i, Ri = size of feature subset, c = Fastmap dimension.

Table 4 summarizes the results of applying multi-level feature selection to the entire instructional video. The method begins by seleting the best 2-feature subset ($R_1$ = 2) for classification. We terminate the process at $R_5$ = 32, where we attain a classification error rate of 0.2%. We stop here for comparison reasons, as we already know that this error rate is equivalent to the error rate attained by applying the more expensive 30-feature Sort-Merge classifier of Figure 4 above to the full video. However, the accumulated work of this boundary refinement approach has been much less, as the bulk of the processing has been done with simpler classifiers; on average, only 3.6 features are used per frame.

## 5. CONCLUSION

We have presented a coarse-fine scene segmentation algorithm using multi-level feature selection. It relies on a low-cost Sort-Merge feature selection algorithm that are well-adapted to video domain, which is impractical with existing feature selection techniques, because of high time complexity. We have illustrated its performance a single, but long, video of an instructional type, but we plan to investigate its utility both across a library of videos of this kind, and also on other genres such as situation comedies which share a similar recurring category structure.

## 6. REFERNECE

[1] A. Vailaya, M. Figueiredo, A. K. Jain, and H.-J. Zhang, “Image Classification for Contnet-Based Indexing”, *IEEE Transactions on Image Processing, vol. 10, no. 1, January, 2001,* pp 117-130.

[2] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta and R. Jain, “Content-based image retrieval: the end of the early years”, *IEEE trans. PAMI, 22 - 12:1349 -- 1380, 2000.*

[3] Michael S. Lew, Thomas S. Huang, Kam W. Wong, “Learning and Feature Selection in Stereo Matching”, *IEEE Transactions on Pattern Analysis and Machine Intelligence on Learning in Computer Vision, September, 1994,* pp. 869-881.

[4] Yan Liu and John R. Kender. Video frame categorization using Sort-Merge feature selection. In *Proceedings IEEE Workshop on Motion and Video Computing, pages 72--77, 2002.*

[5] Christons Faloutsos and king-Ip (David) Lin. FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. *In Proceedings of ACM SIGMOD,*1995, pp 163-174.

[6] Richard O. Duda, Peter E. Hart and David G. Stork, *Pattern classification*, Wiley, New York, 2000.