University of Paris 6 at TRECVID 2005: High-Level Feature Extraction

Christophe Marsala
Université Pierre et Marie Curie-Paris6
CNRS UMR 7606, LIP6,
8 rue du Capitaine Scott,
Paris, F-75015, France
Christophe.Marsala@lip6.fr

Marcin Detyniecki
Université Pierre et Marie Curie-Paris6
CNRS UMR 7606, LIP6,
8 rue du Capitaine Scott,
Paris, F-75015, France
Marcin.Detyniecki@lip6.fr

Abstract

In this paper, we present the methodology we use in the NIST TRECVID'2005 evaluation. We have participated in the High-level Feature Extraction task. Our approach is founded on Fuzzy Decision Trees through the Salammbô software.

1 Structured Abstract - Summary

Here we present the contribution of the University of Paris 6 at TRECVID 2005 [1]. It concerns only the High-Level Feature Extraction task. The approach focuses on the use of Fuzzy Decision Trees (FDT) and is based on a rather simple image description.

In the following, we start with a short summary of the used method and up from Section 3, our approach is detailed. First, we describe the particularities of our image descriptors. Then we explain how we performed the training (Section 4) and classification (Section 5). Before concluding, the submitted runs are discussed in details(Section 6).

1.1 Brief Description of the Runs

Here is the general information about all the runs (more information can be found in the rest of this paper):

The task:	High-Level Feature Extraction.
The feature:	#40. Map: Segment contains video of a map.
Type:	A - system trained on TRECVID development collection data, and common annotation of such data.
Data used:	- XML files that describes the cutting into shot of each video (Master shot references by [5]), - All of the image files representing keyframes, - Annotations files for devel keyframes.
Pre-treatment:	- Each keyframe was cut in 5 regions (see 3.1), - HSV histogram was computed for each piece of a keyframe (see Section 3.1), - Temporal information about each shot was extracted from the XML files. (see Section 3.2).
Training:	The Fuzzy Decision Tree (FDT) learning method was used in each run (see Section 4).

Table 1 shortly differentiates the submitted runs.
The columns of the table describe:

Run Id	#KF	RC id	Operators
A_FuzzyDTzadV1_1	1840	rc1	Zadeh
A_FuzzyDTzadV2_2	1800	rc2	Zadeh
A_FuzzyDTzadV3_3	1840	rc3	Zadeh
A_FuzzyDTzadV4_4	1800	rc4	Zadeh
A_FuzzyDTluka_5	1800	rc4	Łukasiewicz
A_FuzzyDTstrict_6	1800	rc4	Strict

Table 1: Summary of each LIP6 run

- **Run Id**: System Run Id.
- **#KF**: number of keyframes that composed the training set to construct the FDT (see Section 4).
- **RC id**: random choice was used several times to select devel keyframes to compose the training set. RC id enables us to retain the performed random selection, so that we can find the devel keyframes later on (see Section 6).
- **Operators**: internally used operators' family (t-norm and t-conorm, see Section 5.1) used to classify and to rank the test keyframes.

1.2 Comments on the Runs

1.2.1 Differences among the runs

The differences among the runs are:

- Training difference: each run is associated with the same group of keyframes with the Map Feature, but a different random selection of keyframes without the Map Feature is used.
- Training difference: size of the training set.
- Classification difference: choice of the family of operators to aggregate the membership degrees when classifying a new case.

1.2.2 Relative Contribution of each Component

Visual Information Descriptors are crucial since they are at the basis of the learning. We choose to cut the keyframes into a set of rectangular regions and work on their color description. The choice of the number of parts on which we cut a keyframe and the number of bins for the histogram have still to be optimized. Moreover, more visual descriptors should be added in order to enhance the possibilities for the learning algorithm (FDT) to base its decisions.

Video Information Descriptors are also as important. We chose to include the temporal information brought by the position in the video of a shot. It came out that it was a fundamental information.

Training (Fuzzy Decision Tree) is the heart of our approach. The use of decision trees enables us not only to automatically discover the discriminating features, but also it provides an explanation (under the form of rules) of how the classification is performed. The fuzzy logic theory enables a more robust treatment of numerical values of the descriptors. In fact, we have smooth decisions avoiding any threshold effects. Moreover, fuzzy values enables us to have a more general information about numerical values.

Classification (Fuzzy Decision Tree) . Here again, the fuzzy logic theory implies a certain robustness when handling numerical values. Moreover, it enables us to obtain a degree of *mapness* for each keyframe. Without such a degree, it will be impossible to have a good ranking of the keyframe (see the inferior result of A_FuzzyDTstrict_6 which does not use such a degree).

1.2.3 Overall Analysis

We obtained encouraging results (usually among the first fifty of hundred) using extremely simple visual description and out of the box fuzzy decision tree software. The use of this type of algorithm is a novelty on this kind of application. This approach provides as result classification rules which are human understandable, thus allowing further developments. The presented runs are an underestimation of what could be easily obtained. In fact, a lot of shot possessed the same membership degree and were therefore ranked *alphabetically* in a second sub-ranking, masking in this way some good shot by not-so-good ones. In fact, the FDT optimizes the classification of all the examples and not the ranking of the results. Some further developments on the adaptation of fuzzy decision trees to ranking problems (instead of just classification) should be done.

2 Introduction

The method we have used in the NIST TRECVID'2005 evaluation task is based on the use of Fuzzy Decision Trees (FDT). More precisely, we used the Salammbô software, which developed in our team at Computer Science Department of the University of Paris 6: LIP6.

In a first step, before the construction and the use of a Fuzzy Decision Tree, the preliminary work consisted on transforming the data (devel and test set of shots extracted from the video) in order to be processed by the Salammbô software.

The following description of the approach is decomposed as follows: in Section 3, the generation of vectors of descriptors from the keyframes and the XML files is presented. In Section 4, the training process, i.e. the constitution of training sets that should be process by the Salammbô software to construct FDT, is presented. In Section 5, the method of processing FDT to classify keyframes is presented. In particular, we focus on the process that enables us to rank the test keyframes. In Section 6, each of the performed runs is detailed. Finally, we conclude on our experiment of the TRECVID 2005 Challenge.

3 Extraction of Image Descriptors

3.1 Visual Information Descriptors

The *Visual Information Descriptors* are obtained directly from the keyframes.

To obtain visual spatial-related information from the keyframe, we cut the image into 5 pieces (see Figure 1). Each piece corresponds to a spatial part of the keyframe: top, bottom, left, right, and middle. The five regions do not have the same size in order to reflect the importance of the contained information based on its position.

Afterwards, for each region we computed the associated histogram in the HSV space. Depending on the area of the region, the histogram is more or less precise (based on the number of bins): 6x3x3 for Middle, Top, and Bottom, 4x2x2 for Left, and Right.

At the end, we obtain a first set of numerical values (each one ranging from 0 to 1) that characterizes every keyframe. We call this set the Visual Information Descriptors.

3.2 Video Information Descriptors

The *Video Information Descriptors* are obtained from the information associated with the video and given by means of the shot detection process. They correspond to the temporal information associated with the

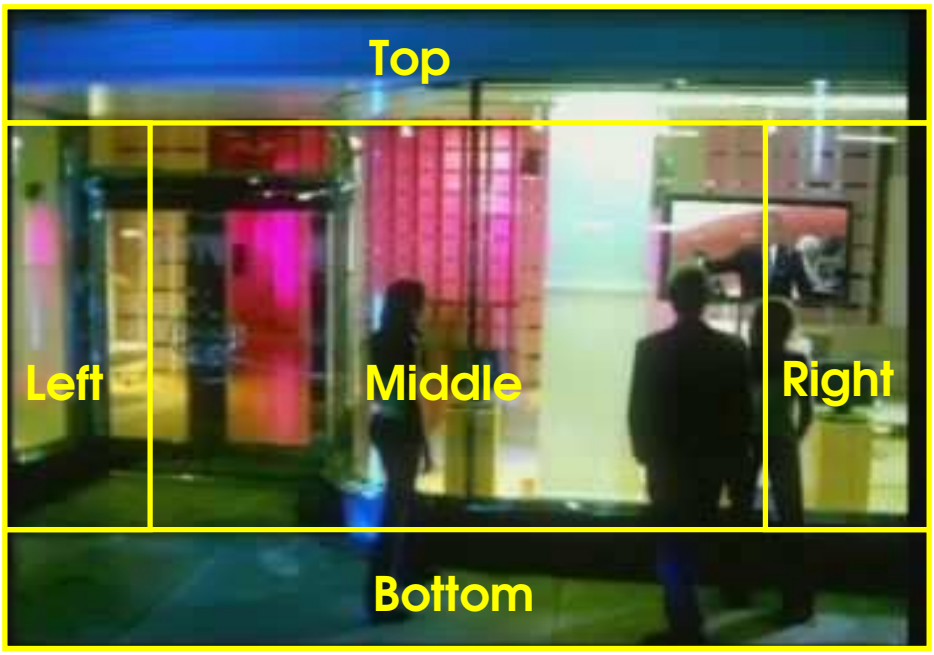

Figure 1: Spatial Decomposition of a Keyframe

shot from which the keyframe was extracted.

For a given keyframe, these descriptors are extracted from the XML file associated with a video and obtained from shot detection process. The XML tags associated with each shot enables us to obtain the following information for every keyframe:

- the name of the keyframe and its kind (RKF or NRKF)
- the timecode of the keyframe in the video
- the timecode of the beginning of the shot containing the keyframe
- the duration of the shot containing the keyframe

At the end, we obtain a second set of numerical values that characterize the keyframe and the shot to which it belongs. We call this second set of information, the Video Information Descriptors.

3.3 Class Descriptor

The *Class Descriptor* is obtained from the indexation of the video. It corresponds to the feature(s) that should be associated with a shot.

The Class Descriptor is extracted from the file obtained from the (human) indexation process of the devel video.

A keyframe can be associated with more than one feature depending on the result of the indexation process. The set of experiments we conducted this year focus just on the feature #40 (*Map: segment contains video of a map*).

4 Training with devel keyframes

The training with the devel keyframes enables us to obtain a classifier that will be used afterwards to classify and rank the test keyframes (see Section 5).

4.1 Building a training set

In order to use the FDT learning method, which is a supervised learning method, we must use a training set in which there are cases with the feature to be recognized and cases that do not possess that feature.

Moreover, decision tree construction methods are based on the hypothesis that the value for the class is equally distributed. This hypothesis is not valid when considering the TRECVID'05 data set. For instance, for the Map feature #40, in the whole devel set of indexed keyframes, there are 940 keyframes with the Map feature and 61273 keyframes without the Map feature. Thus, to have a valid training set for the construction of a fuzzy decision tree, we have to balance the number of keyframes of each class.

For instance, we can choose to select 900 keyframes with each class (with the Map feature, or without the Map feature) in order to build a training set with 1800 keyframes.

4.2 Construction of a Fuzzy Decision Trees

Inductive learning raises from the *particular* to the *general*. Let us assume that a set of classes $C = \{c_1, ..., c_K\}$ is considered, representing a physical or a conceptual phenomenon and that this phenomenon is described by means of a set of attributes $\mathcal{A} = \{A_1, ..., A_N\}$.

In that case, a *description* is a N-tuple of attribute-value pairs (A_j, v_{jl}). Each description is linked with a particular class c_k from C to make up an *instance* (or *example*, or *case*) e_i of the phenomenon. Finally, the inductive learning is the process that generalizes from a *training set* $\mathcal{E} = \{e_1, ..., e_n\}$ of examples to a general law to bring out relations between descriptions and classes in C.

In our case, each attribute A_j can take a (fuzzy, numerical, or symbolic) value v_{jl} in a set $\{v_{j1}, ..., v_{jm_j}\}$ of possible values. We suppose that v_{jl} is associated with a membership function $\mu_{v_{jl}}$. Similarly, each c_k is supposed to be associated with a membership function μ_{c_k}.

4.2.1 Selection of attributes

Most algorithms designed for constructing decision trees proceed in the same way: the so-called *Top Down Induction of Decision Tree* (TDIDT) method. They build a tree from the root to the leaves, by successive partitioning the training set into subsets. Each partition is done by means of a test on an attribute and leads to the definition of a node of the tree. An attribute is selected thanks to a *measure of discrimination* H. Such a measure enables us to order the attributes according to an increasing accuracy when splitting the training set. The discriminating power of each attribute in $\mathcal{A}$ is valued with regard to the classes. The attribute with the highest discriminating power is selected to construct a node in the decision tree.

4.2.2 Construction of Fuzzy Partitions

The process of construction of FDT is based on the knowledge of a fuzzy partition for each numerical attribute. However, it is rare to know, a priori, such a fuzzy partition. Thus an automatic method of construction such a partition from a set of precise values was implemented. In this way we obtain a set of fuzzy values for each numerical attribute [4].

The method is based on the utilization of the mathematical morphology theory. Kernels of concordant values of a numerical attribute related to the values of the class can be found. Fuzzy values induced from a set of numerical values of an attribute are linked with the repartition of the values of the class related to the numerical attribute. Thus a contextual partitioning of an attribute is done that enables us to obtain the best partition related to the attribute with respect to the class.

4.2.3 The Salammbô Software

The construction and the use of the FDT was done by means of the Salammbô software.

This software was developed for building FDT efficiently and it enables us to test several kinds of parameters of the FDT [2, 3]. Moreover, the automatic method to build a fuzzy partition on the set of values

of a numerical attribute, mentioned above, was implemented [4] enabling us to avoid the prior definition of fuzzy values of attributes.

Various parameters (t-norms, t-conorms) can be set in the Salammbô software and have been tested in the process of classification on different kinds of databases.

4.3 Evaluation of Fuzzy Decision Trees

In order to quantify the performance of the FDT, we use a cross validation. It enables us to find the more efficient FDT for the classification of the keyframes.

The cross validation was conducted as follows:

Step 1 The training set is composed by 900 keyframes with the Map feature and 900 keyframes without the Map feature. Each of these keyframes is randomly selected in the corresponding set of keyframes,

Step 2 An evaluation set is composed using the rest of the keyframes (40 with the Map feature, and 60373 without the Map feature).

Step 3 A FDT is constructed by means of the training set, and is used to classify the evaluation set (the presence of the feature for a keyframe is predicted by means of the FDT, and the result is compared with the indexation of this keyframe.

These 3 steps are renewed 3 times in order to obtain several results, which are then averaged.

4.4 Global Fuzzy Decision Trees

The final FDT (submitted to the TRECVID'05 Challenge), was obtained by reproducing the process presented in the last sections, but this time using the whole set of keyframes with the Map feature.

Step 4 The training set is composed by 940 keyframes with the Map feature and 940 keyframes without the Map feature. The keyframes without the Map feature are randomly selected in the whole set of keyframes without the Map feature,

Step 2 A FDT is constructed by means of the training set, and is used to classify the whole test set of keyframes from video 1 to video 140.

As the random selection of a subset of 940 keyframes without the Map feature from the set of 61273 ones enables us to obtain several subsets, we renewed these 2 steps 5 times in order to obtain several runs. Each random selection of a set of keyframes is identified by a *RC id*.

5 Classification and ranking of test shots

5.1 Classifying with a Fuzzy Decision Tree

It is well-known that the path of a decision tree is equivalent to a production rule [3]. The premises for such a rule r are composed by tests on values of attributes, and the conclusion is the value of the class that labels the leaf of the path:

$$\texttt{if } A_{l_1} = v_{l_1} \texttt{ and ...and } A_{l_p} = v_{l_p} \texttt{ then } C = c_k$$

Here, the value of the class can be either *possess the Map feature* or *do not possess the Map feature*. In a FDT, a leaf can be labelled by a set of values $\{c_1, ..., c_K\}$ for the class, each value c_j associated with a weight computed during the learning phase. Thus, a path of a fuzzy decision tree is equivalent to the following rule:

$$\begin{array}{c}\texttt{if } A_{l_1} = v_{l_1} \texttt{ and ...and } A_{l_p} = v_{l_p} \texttt{ then} \\ C = c_1 \texttt{ with the degree } P^*(c_1|(v_{l_1}, v_{l_2}, ...v_{l_p})) \texttt{ and ...} \\ \texttt{and } C = c_K \texttt{ with the degree } P^*(c_K|(v_{l_1}, v_{l_2}, ...v_{l_p}))\end{array}$$

In a FDT, each value v_i can be either precise or fuzzy, and is described by means of a membership function μ_{v_i}.

Now, when a keyframe e, described by means of a set of values $\{A_1 = w_1; ...; A_n = w_n\}$, is to be classified, this description is compared with the premises of the rule r to value the degree with which the observed value w is near the edge value v. This proximity is valued as a degree $\texttt{Deg}(w, v)$. In our case, the value w is a precise value and we have $\texttt{Deg}(w, v) = \mu_v(w)$.

For each premise, the degree $\texttt{Deg}(w_{l_i}, v_{l_i})$ is valued for the corresponding value w_{l_i}. Finally, given the rule r, the keyframe e is associated with the class c_j with a *final degree* $\texttt{Fdeg}_r(c_j)$. This final degree is the aggregation of all the degrees $\texttt{Deg}(w_{l_i}, v_{l_i})$ by means of a t-norm $\top$ (for instance, the *minimum*):

$$\texttt{Fdeg}_r(c_j) = \top_{i\,=\,1...p} \texttt{Deg}(w_{l_i}, v_{l_i}).P^*(c_j|(v_{l_1}, v_{l_2}, ...v_{l_p}))$$

Final degrees computed from all the rules are aggregated by means of a t-conorm $\perp$ (for instance, the *maximum*) to obtain a single degree of satisfiability $\texttt{Fdeg}(c_j)$. If n_ρ is the number of rules given by the fuzzy decision tree:

$$\texttt{Fdeg}(c_j) = \perp_{r\,=\,1...n_\rho} \texttt{Fdeg}_r(c_j)$$

For each value of the class, the description e is associated with such a membership degree $\texttt{Fdeg}(c_j)$, from $[0, 1]$, for each class c_j computed from the whole set of rules. The class c_e associated with e can be chosen as the class with the higher membership degree:

$$\texttt{FDeg}(c_e) = \max_{j=1...K} \texttt{FDeg}(c_j)$$

We used this process of aggregation in order to have meaningful values of degrees for each class.

In this process of classification, there are 2 operators that should be stated: the t-norm $\top$ and the t-conorm $\perp$. Such t-operators can be dual and it has been proved that it is better to select a coherent pair. For the TRECVID 2005 challenge, we used 2 families of triangular operators (see Table 2).

	Zadeh operators	Łukasiewicz operators
$\top(x, y)$	$\min(x, y)$	$\max(x + y - 1, 0)$
$\perp(x, y)$	$\max(x, y)$	$\min(x + y, 1)$

Table 2: Families of triangular operators

5.2 Classifying keyframes

After the construction of the FDT as explained in Section 4, each FDT is used to classify the whole test set of keyframes.

First of all, Visual Information Descriptors and Video Information Descriptors are extracted for all the keyframes from the test set. This enables us to obtain vectors of numerical data that can be classified with the FDT.

By means of the classification, each keyframe e from the test set is associated with a membership degree $\texttt{FDeg}(c_e)$ to the Map feature.

5.3 Ranking shots

At the end, each shot from the test video set is associated with a membership degree $\texttt{FDeg}(c_e)$ of its keyframe e. All the test shots can thus be ranked by means of these membership degrees. We assumed that the higher the membership degree, the more confident the FDT is of the presence of the feature in the shot. This ranking method is the one used for all runs submitted to TRECVID 2005.

6 Submitted runs

The choice of the operators (see Section 5.1) when combining the membership degrees leads to different runs for TRECVID 2005.

The detail for each run is:

Run 1 *(FuzzyDTzadV1)*: Zadeh operators are used to combine the membership degrees when classifying a keyframe. The training set is composed by all the 940 keyframes with the Map feature, and 940 keyframes randomly chosen from the set of keyframes without the Map feature. The RC id is 1.

Run 2 *(FuzzyDTzadV2)*: Zadeh operators are used to combine the membership degrees when classifying a keyframe. The training set is composed by a random selection of 900 keyframes with the Map feature, and 900 keyframes randomly chosen from the set of keyframes without the Map feature. The RC id is 2.

Run 3 *(FuzzyDTzadV3)*: Zadeh operators are used to combine the membership degrees when classifying a keyframe. The training set is composed by all the 940 keyframes with the Map feature, and 940 keyframes randomly chosen from the set of keyframes without the Map feature. The RC id is 3.

Run 4 *(FuzzyDTzadV4)*: Zadeh operators are used to combine the membership degrees when classifying a keyframe. The training set is composed by a random selection of 900 keyframes with the Map feature, and 900 keyframes randomly chosen from the set of keyframes without the Map feature. The RC id is 4 (same set as for Run 5 and 6).

Run 5 *(FuzzyDTluka)*: Łukasiewicz operators are used to combine the membership degrees when classifying a keyframe. The training set is composed by a random selection of 900 keyframes with the Map feature, and 900 keyframes randomly chosen from the set of keyframes without the Map feature. The RC id is 4 (same set as for Run 4 and 6).

Run 6 *(FuzzyDTstrict)*: The FDT is considered as a non fuzzy decision trees: the fuzzy values that label the tree are unfuzzified before the classification. At the end, the (F)DT is used as a classical decision tree. The training set is composed by a random selection of 900 keyframes with the Map feature, and 900 keyframes randomly chosen from the set of keyframes without the Map feature. The RC id is 4 (same set as for Run 4 and 5).

The results are the following:

Run sent by LIP6	Average precision	#Hits in 100	#Hits in 1000	#Hits in 2000
FuzzyDTzadV1	0.117	51	343	581
FuzzyDTzadV2	0.147	69	411	586
FuzzyDTzadV3	0.145	68	406	483
FuzzyDTzadV4	0.163	57	377	683
FuzzyDTluka	0.099	33	303	657
FuzzyDTstrict	0.021	37	115	189
Results from TRECVID 2005:				
- Best method	0.524	100	876	1136
- Mean of the results	0.24	81.4	491.8	651.3
- Median of the results	0.185	91	410	557

Table 3: Results for feature #40 (Map)

The best, the mean and the median were extracted from the 103 results (by all the methods) that were sent for evaluation to TRECVID.

7 Conclusion

Although, this work is still at a preliminary stage, we obtained encouraging results. In fact our runs were ranked among the first half. One of the main drawbacks of our method is that it is based on very simple and generic visual descriptions. Unfortunately, the development of these descriptors was not in our research scope.

However, we should notice two main contributions to the description part. It seems that in general decomposing the keyframes into regions improves the quality of the classification. Also adding extra features, as for instance the temporal position of the keyframe inside the video program (news), strongly improves the results. Further developments will focus on how to obtain more of these non-visual features.

Moreover, as far as we know, this is the first time that Fuzzy Decision Trees (and more generally Decision Tree algorithms) are applied to this type of problems. We notice its relatively good performance, taking into account the fact that they were not tweaked to the problem.

For us, it is the first time we manage the whole process from the segmented video (in shots) to the ranking of the shots after the classification. Many drawbacks of our method were discovered during the developments. In particular we noticed that the FDT (as all classification algorithms) do not necessarily optimize the ranking, but rather look for a compromise in order to classify as many as possible keyframes as possible. In other words, the degree provided by the FDT does not necessarily discriminate the very good results from others less good. In fact, the goal of a FDT is to discriminate between any non MAP and a MAP keyframe and not to order from the most "map" one to the less one. This implies that our approach performs worse if compared for the first results (e.g. rank 100) rather than at a larger scale (e.g. the whole test set).

It is also to notice, that we were short in time since a lot of technical problems arisen not only when extracting the descriptors but also when the runs were performed. In any case, for this our first complete participation and we are proud to have fulfilled the challenge in time and in spite of our limited means.

We plan to participate in the next challenge and to exploit all the fruitful experience we acquire this year.

Acknowledgement

The authors would like to thank Maria Rifqi for her help during the TRECVID 2005 challenge.

References

[1] Guidelines for the TRECVID 2005 evaluation - National Institute of Standards and Technology, 2005. http://www-nlpir.nist.gov/projects/tv2005/tv2005.html.

[2] B. Bouchon-Meunier, C. Marsala, and M. Ramdani. Learning from imperfect data. In D. Dubois, H. Prade, and R. R. Yager, editors, *Fuzzy Information Engineering: a Guided Tour of Applications*, pages 139–148. John Wileys and Sons, 1997.

[3] C. Marsala. *Apprentissage inductif en présence de données imprécises : construction et utilisation d'arbres de décision flous.* Thèse de doctorat, Université Pierre et Marie Curie, Paris, France, Janvier 1998. Rapport LIP6 n^o 1998/014.

[4] C. Marsala and B. Bouchon-Meunier. Fuzzy partioning using mathematical morphology in a learning scheme. In *Proceedings of the 5th IEEE Int. Conf. on Fuzzy Systems*, volume 2, pages 1512–1517, New Orleans, USA, September 1996.

[5] C. Petersohn. Fraunhofer HHI at TRECVID 2004: Shot boundary detection system. Technical report, TREC Video Retrieval Evaluation Online Proceedings, TRECVID, 2004. URL: www-nlpir.nist.gov/projects/tvpubs/tvpapers04/fraunhofer.pdf.