Proc. ACM Multimedia 2010, Events in MultiMedia Workshop (EiMM10), Firenze, Italy, October 2010.

# Automatic Event-Based Indexing of Multimedia Content Using a Joint Content-Event Model

Nikolaos Gkalelis
Informatics and Telematics Institute / CERTH
6th Km Charilaou-Thermi Road, Thermi 57001, Greece
gkalelis@iti.gr

Vasileios Mezaris
Informatics and Telematics Institute / CERTH
6th Km Charilaou-Thermi Road, Thermi 57001, Greece
bmezaris@iti.gr

Ioannis Kompatsiaris
Informatics and Telematics Institute / CERTH
6th Km Charilaou-Thermi Road, Thermi 57001, Greece
ikom@iti.gr

## ABSTRACT

In this paper a joint content-event model for the automatic indexing of multimedia content with events is proposed. This model treats events as first class entities and provides a referencing mechanism for automatically linking event elements (represented using the event part of the model) with content segments (described using the content part of the model). The emphasis of the paper is on this mechanism, which uses trained concept detectors to represent content segments with model vectors, and the subclass discriminant analysis algorithm to derive a discriminant subspace facilitating the indexing of content segments with event elements. The use of this referencing mechanism for associating multimedia content with five sport events is demonstrated on the MediaMill dataset.

## Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; I.2.10 [**Computing Methodologies**]: Vision and Scene Understanding

## General Terms

Algorithms, Theory

## 1. INTRODUCTION

With the rapid progress of hardware technology and the popularity of related multimedia devices, the quantity of multimedia data has surged into an unprecedented level. These data reside on local personal computers or global large scale repositories, and large amounts of them are daily consumed within the framework of networked media. For manipulating this information, sophisticated algorithms are needed for supporting the automatic indexing of multimedia. However, this is to some extent still beyond the capabilities of the current state-of-the-art in multimedia management. This is mainly due to the so-called semantic gap between the descriptions of multimedia data provided by automatic analysis tools and the meaning of the same multimedia data to humans. Recent studies in neuroscience have shown that humans remember real life using past experience structured in events [22]. For this reason, event-based indexing of multimedia content is expected to help reduce the semantic gap between human and machine interpretations.

The necessity of formal event models for describing real life events has been recently acknowledged, and a number of such models have been developed, e.g. [3, 21, 15, 9]. In [3], the IPTC G2 family of news exchange standards are provided, including EventML, the respective standard for describing events in a journalistic fashion. In [21], the event model E and a number of common event model requirements are presented. In [15], the event model F is proposed, which is based on the DOLCE foundational ontology [10] to provide formal semantics and representation of context. In [9], the video event representation language (VERL) is presented for the description of events in videos.

The models reviewed above present some drawbacks, including: a) they treat events as second class entities, i.e., the existence of the events depends on the content they describe [9], b) they provide little or no support for capturing the structure of multimedia content [3, 21, 15]. Most importantly, though, these models do not provide a mechanism for automatically associating the multimedia content with the events or event elements that the model represents. This is partly addressed in [5] where the temporal, casual and spatial aspects of the event model E are implemented and used for representing "Gunshot" and "Walkthrough" events in videos.

On the other hand, several researchers have proposed algorithms for automatically recognizing events in multimedia data, however, without explicitly providing a model for describing events, e.g. [20, 6, 7]. In [20, 7], HMMs are used to recognize ice hockey or wedding ceremony events in video. In [6], a video is represented as a sequence of Bag-of-Words (BoW) histograms and SVMs are used to learn the desired events. The algorithm is evaluated on a subset of the MediaMill video dataset and on a soccer video dataset collected by the authors.

In this paper, we propose a joint content-event model to address the limitations of the current event models addressed above, i.e., a model that treats events as first class entities, allows the description of multimedia content and offers a referencing mechanism for automatically linking event

*EiMM'10,* October 25, 2010, Firenze, Italy.

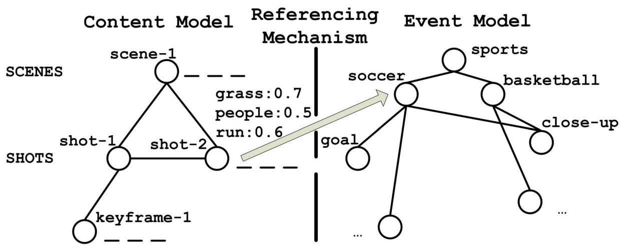


**Figure 1:** *Use of the proposed joint content-event model.*

elements with relevant content segments. This mechanism consists of a set of pre-trained concept detectors for semantically describing multimedia content, and an algorithm based on the subclass discriminant analysis (SDA) [23] for indexing content segments with events.

The rest of the paper is organized as follows. In section 2 the proposed joint content-event model is presented, while in section 3 the referencing mechanism of the proposed model is evaluated on the MediaMill dataset and several experimental results and comparisons are provided. Finally, concluding remarks are drawn in section 4.

## 2. JOINT CONTENT-EVENT MODEL

The joint content-event model has been designed to satisfy a number of requirements extracted after reviewing relevant literature, especially the work of [21]. It consists of a content part, an event part and a referencing mechanism linking the two parts. The content part of the model has a hierarchical graph structure consisting of nodes and edges, as shown on the left side of Figure 1. Content nodes are structurally alike, i.e., they all consist of the same set of properties, and each node is used to convey information for exactly one content segment. Edges between nodes are used to reflect content segments' compositional and temporal relations. The event part of the model has a more general graph structure, as shown on the right side of Figure 1. An event node corresponds to one real-life event element, e.g. a sub-event, and all event nodes have the same structure. Edges between nodes indicate a variety of relationships, e.g., temporal or causal. The properties of the event and content nodes are depicted in Figure 2. We observe that there is a number of properties that are common in both the content and the event node. There is also a number of event node properties that their values can be inferred from the values of the respective content node properties. This set of properties supports the referencing mechanism of the proposed model: the values of these properties can be used to associate one or more content nodes with an event node (based on, e.g., "absolute time" values of both the content and event nodes), in parallel to the process discussed in section 2.3, and, in the case of association, initialize several properties of the event node (Figure 1).

### 2.1 Content node properties

A type taxonomy similar to the Segment Description Scheme (DS) of MPEG-7 Multimedia DS (MDS) is deployed to characterize the type of content segments, e.g., video segment, moving region, or further specializations such as scene, shot, etc. Content segment type information is recorded in the contentType property. The ID property is filled with a URI to index content nodes in a uniform manner, while the technicalDetails property holds technical details of the multimedia data, e.g., frame rate, aspect ratio, etc. The properties hasParent, isChild, precedes and follows receive an URI to reflect the relative position of content nodes in the content graph. The mediaSpatialLocation property is used to describe a region of an image or frame, e.g., a rectangular area around a human body. The mediaTemporalLocation property records information regarding the temporal position of a content segment, e.g., it may record the start and the end frame of a shot. The properties creatorName, textAnnotation, absoluteLocation and absoluteTime hold information extracted from the metadata accompanying the content segment; the two latter are used to hold geospatial and time-related information respectively. Finally, concepts extracted from the content segments using trained concept detectors are described in the conceptIDs property.

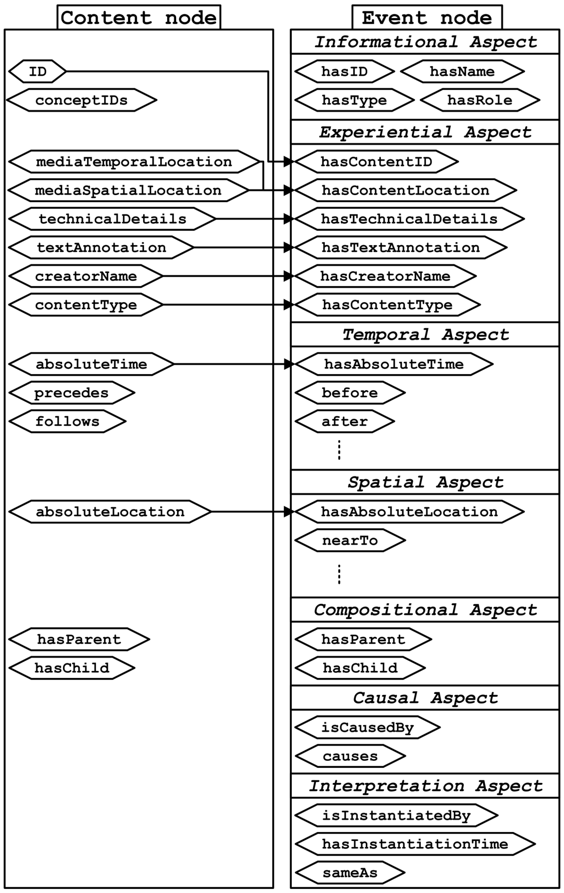


**Figure 2:** *Event and content node properties.*

### 2.2 Event node properties

Event node properties have been selected to cover a number of real-life event aspects. The informational aspect of the event is modelled with the properties hasID, hasName, hasType and hasRole. The hasID property receives a URI to represent the event node in a global scope. Three classes of the ultra light version of DOLCE (DUL) are used to model the type of an event element, i.e., Event, Agent or Place [10]. Following DUL, an event element of type Event is used to model any physical, social, or mental process, event, or state. The Agent class is used to represent any agentive object participating in the event, either physical or social, e.g., a gun or a corporation. The Place class is used to denote a generic location, e.g., Paris. This information is recorded in the hasType property. The hasName property holds the name of the event element after its instantiation, and the hasRole property, also adopted from DUL, is used to classify the event element in different context, e.g., it can classify a person as a policeman during a robbery event, or as the victim of a gunshot event.

The experiential aspect of an event is captured using six properties, namely, hasContentID, hasCreatorName, hasTextAnnotation, hasContentType, hasTechnicalDetails, hasContentLocation. These properties are automatically filled with information directly transferred from the respective properties of the content node, as shown in Figure 2. A connection between a content node and an event node is automatically established using the referencing mechanism of the model described in section 2.3. We should note that the hasTechnicalDetails and the hasContentLocation properties capture all the necessary information to locate and use the content segment itself, avoiding the overhead for accessing the content description part of the model again during a retrieval operation (query).

The absolute time of an event is recorded in hasAbsoluteTime property using the W3C Datetime Format profile of ISO 8601 standard [2], while relative temporal event information is captured using Allen's Time Calculus [4]. Similarly, the hasAbsoluteLocation property captures the absolute spatial location of an event in (latitude, longitude) form defined in Basic Geo (WGS84 lat/long) Vocabulary [1], and the nearTo and farFrom DUL properties along with the Region Connection Calculus (RCC) properties [14] are used to denote relative spatial relations between event elements.

Compositional information between events is captured using the properties hasParent and hasChild to record immediate super- or sub-events respectively, while causal information is captured using the properties causedBy and causes. Finally, to allow different interpretation of same event, we use the properties isInstantiatedBy and hasInstantiationTime to capture the creator and the creation time of an event node, and the sameAs property to link two or more event nodes representing the same entity in different context.

## 2.3 Referencing mechanism

The target of the referencing mechanism is to automatically index a content segment with an event element. To achieve this, it takes advantage of trained concept detectors [13] which may exploit audiovisual or other information of a content segment to derive a model vector representation of it [17]. The resulting model vector serves as input, according to the developed referencing mechanism, to a discriminant subspace learning algorithm, which is used for obtaining a low dimensional feature space where different event classes are expected to separate better. The motivation behind the choice of SDA [23] for learning the low dimensional feature space is: a) Similar to other subspace methods, classification of testing samples is performed in a reduced feature space, thus allowing for low storage requirements and fast processing times. This advantage is especially important when computationally intensive classifiers, such as Nearest Neighbor (NN), are used in combination with large-scale training datasets. b) SDA approximates any data distribution with a mixture of Gaussians, in contrast to other discriminant analysis (DA) algorithms, e.g. linear discriminant analysis (LDA) [8], applicable to only specific data distribution types. Moreover, compared to kernel DA methods, such as [12], that can also be used to fit various data distributions, SDA has lower computational complexity. c) The optimal number of mixture components (subclasses) in SDA is automatically identified using an optimality criterion, in contrast to heuristics used by competing DA methods. d) In particular for our problem, the possibility to associate event elements with SDA subclasses may worth further investigation, e.g., by exploiting real-world event element relationships (spatial, temporal, etc.) to improve the effectiveness and efficiency of the referencing mechanism.

The proposed indexing mechanism has to be trained first for a number of different event classes or specific events of interest, and can then be used for automatically indexing non-annotated content with one or more of these events.

### 2.3.1 *Training*

Let $\mathcal{U} = \{(\mathbf{s}_1, y_1), \dots, (\mathbf{s}_N, y_N)\}$ be an annotated training database of $N$ content segments belonging to one of $C$ event classes, where $\mathbf{s}_i$ is the feature vector representation of the $i$-th content segment and $y_i \in \{1, \dots, C\}$ its class label. We assume that a set $\mathcal{G} = \{(d_\kappa(), h_\kappa),\ \kappa = 1, \dots, K\}$ of trained concept detectors is given, where $d_\kappa()$ is the $\kappa$-th concept detector functional and $h_\kappa$ is the respective concept label. The trained concept detectors are used to represent the $i$-th content segment with a model vector $\mathbf{x}_i = [x_{i,1}, \dots, x_{i,K}]^T$, $\mathbf{x}_i \in \Re^K$, where $x_{i,\kappa} = d_\kappa(\mathbf{s}_i)$ is a number in the range $[0, 1]$ expressing the degree of confidence (DoC) that the $\kappa$-th concept is present in the $i$-th content segment.

The model vectors are used as the feature vectors in the input space of SDA. SDA initiates with the eigenanalysis of the sample covariance matrix $\mathbf{\Sigma}_X$ for obtaining the respective eigenvectors $\mathbf{u}_\kappa$:

$$\mathbf{\Sigma}_X \mathbf{U} = \mathbf{U}\mathbf{\Lambda}_X \rightarrow \mathbf{U} = [\mathbf{u}_\kappa] \ . \qquad (1)$$

An iterative algorithm is then used to estimate the optimal subclass division of event classes. At each iteration a different subclass division is evaluated. The $r$-th iteration of SDA consists of three steps:

1) The application of a Nearest Neighbor-based clustering algorithm, to partition each event class to $H^{(r)}$ subclasses.

2) The computation of the between-subclass matrix,

$$\begin{aligned}\mathbf{\Sigma}_B^{(r)} \ &= \ \sum_{i=1}^{C-1} \sum_{j=1}^{H^{(r)}} \sum_{k=i+1}^{C} \sum_{l=1}^{H^{(r)}} \Big( p_{i,j}^{(r)} p_{k,l}^{(r)} \times \\ & (\mathbf{m}_{i,j}^{(r)} - \mathbf{m}_{k,l}^{(r)})(\mathbf{m}_{i,j}^{(r)} - \mathbf{m}_{k,l}^{(r)})^T \Big) \ , \qquad (2)\end{aligned}$$

and its eigenanalysis for deriving the respective eigenvectors $\mathbf{w}_i^{(r)}$, where $p_{i,j}^{(r)}$ and $\mathbf{m}_{i,j}^{(r)}$ are the prior and mean of the $j$-th subclass of the $i$-th class.

3) The computation of a subclass partitioning evaluation criterion,

$$O^{(r)} = \frac{1}{a^{(r)}} \sum_{i=1}^{a^{(r)}} \sum_{j=1}^{i} (\mathbf{u}_j^T \mathbf{w}_i^{(r)})^2 \ , \qquad (3)$$

where $a^{(r)} < rank(\mathbf{\Sigma}_B^{(r)})$ [23].

After several iterations the optimal subclass partitioning $H^{(r_o)}$ is the one corresponding to the iteration $r_o$ given by

$$r_o = \arg\min_{r}(O^{(r)}) \ . \tag{4}$$

To this end, the following generalized eigenvalue problem is solved

$$\mathbf{\Sigma}_X^{-1}\mathbf{\Sigma}_B^{(r_o)}\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \ , \tag{5}$$

where $\mathbf{\Sigma}_B^{(r_o)}$ is the between-subclass matrix that corresponds to the $r_o$-th iteration, and the final projection matrix $\mathbf{V}^*$ is formed by taking the eivenvectors of $\mathbf{V}$ that correspond to the $q$ largest eigenvalues.

The training samples are then projected to the discriminant subspace

$$\mathbf{z}_i = \mathbf{V}^{*T}\mathbf{x}_i, \ i = 1, \ldots, N \ , \tag{6}$$

and stored in the database in order to be used during the testing stage for automatically associating non-annotated content segments with one or more of the learned events.

### 2.3.2 Indexing of non-annotated content

Given a set of multimedia data that is possibly related with one or more of the events learned during the training stage (section 2.3.1), automatic analysis techniques, such as temporal segmentation to shots and scenes [16, 19] in case of video content, are initially applied to derive a set of content segments. Each content segment is then represented with a model vector using the trained concepts detectors, and these model vectors are further projected to the discriminant subspace using the SDA projection matrix. The $n$-th test content segment with feature vector representation $\acute{\mathbf{z}}_n$ in the discriminant subspace and unknown label $\acute{y}_n$ can then be associated with one or more of the learned event classes by examining the quantity $\mathbf{z}_i^T\acute{\mathbf{z}}_n$ for every $i$. Assuming that every content segment can be associated with only one event class, in this work the following NN classifier is used to this end:

$$\acute{y}_n = \arg\min_{i\in[1,\ldots,N]}(\mathbf{z}_i^T\acute{\mathbf{z}}_n) \ . \tag{7}$$

As a result, the automatic association of each content segment with an event or sub-event is achieved, exploiting the results of realistic, non-perfect concept detectors.

## 3. EXPERIMENTAL RESULTS

In this section we use the MediaMill Challenge dataset [18] as a testbed for evaluating the referencing mechanism of the proposed model. Five sport events are considered in this preliminary evaluation, namely basketball, soccer, football, baseball and golf.

### 3.1 Dataset description

The MediaMill Challenge dataset has become a popular benchmark set due to its extensive ground truth annotation provided at the shot level. It contains about 85 hours of news broadcasts from 13 international TV programs in Arabic, Chinese, and English, and is annotated with 101 concepts covering a wide range of topics ranging from abstract one (e.g., indoor, outdoor) to more specific one such as sport events (e.g., golf, baseball) or names of well known people (e.g., B. Clinton, T. Blair). Ground truth data are provided at the shot level, where a shot may be annotated with multiple concepts, e.g., a shot may be labelled simultaneously with the concepts soccer, grass, people, and other. In overall, this extended annotation effort offers a collection of more than 40 thousand multi-labelled shots.

We split this annotated shot database to two equally sized, independent datasets, $\mathcal{D}_1$ for training the concept detectors, and $\mathcal{D}_2$ for evaluating the model referencing mechanism. For the evaluation of the referencing mechanism we are interested in the shots with the following labels: $h_6$ = baseball, $h_7$ = basketball, $h_{41}$ = football, $h_{42}$ = golf, $h_{82}$ = soccer; it is these labels that we treat as event classes in our experiments with the proposed content-event model. Shots annotated with the above 5 labels are extracted from $\mathcal{D}_2$ to form an event evaluation set $\mathcal{U}$ of 492 shots in total ($N = 492$). This process yields a set of disjointed event classes, i.e., each shot in $\mathcal{U}$ belongs to only one event class but is also annotated with multiple concepts out of the remaining 96 ones that we do not treat as event classes. The number of shots in each event class are shown in Table 1.

| **basketball** | **soccer** | **football** | **baseball** | **golf** |
|---|---|---|---|---|
| 119 | 198 | 71 | 53 | 51 |

**Table 1:** *Event dataset.*

For applying concept detectors to this dataset, we use one keyframe per shot; one exemplary keyframe for each event class is shown in Figure 3.

### 3.2 Training of the concept detectors

One of the methods used in the TRECVID experiments of [13] is applied on the $\mathcal{D}_1$ dataset for training one concept detector $d_\kappa()$ for each concept $h_\kappa$. This method is briefly described in the sequel.

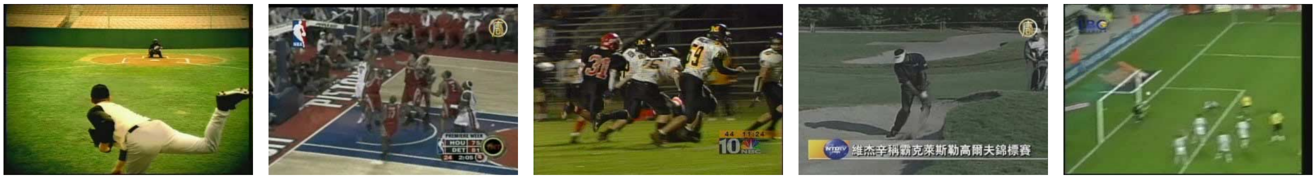
A Bag-of-Words (BoW) procedure is used to represent each shot keyframe with a 100-dimensional feature vector. This is done by firstly extracting keypoints from each keyframe and describing each keypoint with a 128-dimensional SIFT vector, secondly clustering the SIFT vectors to create a vocabulary of 100 Visual Words, and thirdly using the created vocabulary to represent each shot on the basis of the keypoints extracted from it.

The feature vectors described above are used for training 101 SVM-based concepts detectors ($K = 101$) using the one-against-all method. That is, the $\kappa$-th SVM is trained considering all shot keyframes labelled with the concept $h_\kappa$ as positive samples and the rest of the keyframes as negative samples. The output of the each SVM is a number in the range $[0, 1]$ expressing the DoC that the concept $h_\kappa$ is present in the keyframe, as explained in section 2.3.1. From our TRECVID 2008 experiments it has been shown that the employed concept detection method [13] ranks close to the median, hence, it generates moderately accurate concept detectors compared to the current state-of-the-art.

### 3.3 Evaluation of the referencing mechanism

The performance of the referencing mechanism is evaluated on the $\mathcal{U}$ dataset (described in section 3.1) by applying a 50-fold cross-validation procedure. At each validation cycle 20% of the samples from each event class are removed to form the test set, while the remaining 80% of the samples form the training set.

The BoW procedure described in section 3.2 is used to rep-



Baseball Basketball Football Golf Soccer

**Figure 3:** *Shot keyframes of the five events.*

resent each training shot keyframe with a 100-dimensional feature vector, and the 101 trained concept detectors are used to further map the BoW feature vectors to 101-dimensional model vectors, as explained in section 2.3.1. The resulting model vectors are used by the referencing mechanism for learning the event classes. During testing the same procedure is followed to represent the test shot keyframe with a 101-dimensional model vector and the trained referencing mechanism is used to index the non-annotated shot with one of the considered events.

For comparison purposes we also evaluated the performance of three other approaches described in the following:

1) *Concept-based indexing using the Max Rule*: The $n$-th test shot keyframe is represented with a 100-dimensional feature vector $s_n$ using the BoW method described in section 2.3.1. The concept detectors that correspond to the five event classes are then evaluated, $\acute{x}_{n,\kappa} = d_\kappa(\acute{\mathbf{s}}_n)$, $\kappa \in [6, 7, 41, 42, 82]$, and the Max Rule [11] is used to assign the test keyframe to the event class with the maximum DoC

$$\acute{y}_n = \underset{\kappa\in[6,7,41,42,82]}{\arg\max} \ (\acute{x}_{n,\kappa}) \ . \tag{8}$$

2) *Event-based indexing in the Input Space*: The test model vector $\acute{\mathbf{x}}_n$ is perceived as a feature vector representation of a non-annotated keyframe belonging to one of the five event classes (i.e., we do not only consider the $\acute{x}_{n,\kappa}$ components that correspond to the five event concepts). The NN classifier is directly used in the input space to index the test keyframe

$$\acute{y}_n = \underset{i\in[1,\ldots,N]}{\arg\min} \, (\mathbf{x}_i^T \acute{\mathbf{x}}_n) \ . \tag{9}$$

3) *Event-based indexing using LDA*: The training model vectors are used to compute a linear projection matrix $\mathbf{W}$ using LDA [8]. LDA seeks for the linear projection $\mathbf{W}$ that maximizes the criterion $J_{LDA}(\mathbf{W}) =\mid \mathbf{W}^T \boldsymbol{S}_b \mathbf{W} \mid / \mid \mathbf{W}^T \boldsymbol{S}_w \mathbf{W} \mid$, where $\mathbf{S}_w$, $\mathbf{S}_b$, are the within and between scatter matrices respectively. In case that the number of the training samples $N$ is adequately larger than the dimensionality of the input space $K$, $\mathbf{W}$ is formed by the generalized eigenvectors that correspond to the largest eigenvalues of $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$. The indexing of a test model vector $\acute{\mathbf{x}}_n$ is then done by first projecting it in the discriminant subspace using $\mathbf{W}$ and then applying the NN classifier. That is, the equations (6), (7) are used respectively, where $\mathbf{V}$ is replaced by $\mathbf{W}$.

**Concept-based indexing (Max Rule)**

| | baseball | basketball | football | golf | soccer |
|---|---|---|---|---|---|
| baseball | **9.3%** | 11.8% | 32% | 31.4% | 15.4% |
| basketball | 16.2% | **52.3%** | 12.8% | 4.7% | 13.9% |
| football | 6.3% | 21.4% | **54.6%** | 3.4% | 14.3% |
| golf | 27.4% | 7.6% | 10% | **51.2%** | 3.8% |
| soccer | 5.5% | 2.1% | 3.6% | 4.8% | **83.9%** |

**Event-based indexing (Input Space)**

| | baseball | basketball | football | golf | soccer |
|---|---|---|---|---|---|
| baseball | **33.6%** | 16.3% | 11.4% | 17.6% | 20.9% |
| basketball | 6.9% | **64%** | 8.2% | 4% | 16.8% |
| football | 7.3% | 12.1% | **58.6%** | 3.7% | 18.3% |
| golf | 7.6% | 6.2% | 2.4% | **69.4%** | 14.4% |
| soccer | 5.8% | 5.8% | 2.9% | 2.6% | **82.7%** |

**Event-based indexing (LDA)**

| | baseball | basketball | football | golf | soccer |
|---|---|---|---|---|---|
| baseball | **30.2%** | 18.7% | 13.1% | 20.9% | 17.1% |
| basketball | 7% | **62.7%** | 11.2% | 4.2% | 14.9% |
| football | 11.3% | 18.7% | **54.3%** | 3.6% | 12.1% |
| golf | 14.6% | 8.4% | 6.4% | **56.2%** | 14.4% |
| soccer | 5.8% | 8.1% | 4.1% | 4.2% | **77.6%** |

**Event-based indexing (SDA)**

| | baseball | basketball | football | golf | soccer |
|---|---|---|---|---|---|
| baseball | **38.3%** | 12.2% | 14.7% | 14.9% | 19.8% |
| basketball | 5.3% | **62.6%** | 10% | 3% | 18.9% |
| football | 5.4% | 14.4% | **67%** | 0.8% | 12.3% |
| golf | 7.4% | 8% | 2.8% | **68.6%** | 13.2% |
| soccer | 4.2% | 6.5% | 3.5% | 2.7% | **83.1%** |

**Table 2:** *Confusion matrices.*

In order to analyze the results of event detection according to the proposed approach and the three others discussed above, confusion matrices are shown for each approach in Table 2. From the confusion matrices we observe the following: a) Most of the soccer event shots are correctly classified; however, a very large portion of the shots of the other event classes are misclassified as soccer events. This is probably expected as soccer, football, baseball and golf are all outdoor sports involving a green playfield, while both soccer and basketball are also intensive sports, many times involving groups of people in action. b) The basketball, football and soccer are quite distinct from golf (less than 5% of their shots are misclassified as golf for all the approaches), probably because they are team, intensive sports, in contrast to golf which involves individuals moving rather slowly. c) The golf event is highly confused with the baseball event when the Max Rule and the LDA-based approaches are used (27.4% and 14.6% of the golf shots are confused as baseball shots respectively). This is probably due to the fact that golf and baseball shots most of the times depict individuals holding a club (a bat or a golf club) moving rather slowly on a green playfield. d) The baseball event is often confused with the others, especially in the case of the Max Rule approach, where only 9.3% of the baseball test samples are correctly classified.

The overall performance of each approach is measured using the average correct classification rate (ACCR) for the whole 50-fold cross-validation procedure. This is done by first computing the correct classification rate (CCR) at each validation cycle, and then averaging them over all the validation cycles. The CCR for one validation cycle is computed by dividing the number of correctly classified test samples by the number of all test samples in this cycle. The ACCRs of the four methods are given in Table 3. We observe that the best ACCR is provided by the approach using SDA, indicating that event classes have indeed a subclass structure that is appropriately captured by SDA. Moreover, it is verified that event-based indexing provides superior results in comparison to directly examining the concept detector scores (Max Rule).

| | **Max Rule** | **Input Space** | **LDA** | **SDA** |
|---|---|---|---|---|
| ACCR | 60.5% | 67.9% | 63.2% | **69.4%** |

**Table 3:** *Comparison of the four methods.*

# 4. CONCLUSIONS

A joint content-event model for indexing multimedia content was proposed that addresses several limitations of the current state of the art models. The core of the model is a referencing mechanism that allows automatic indexing of multimedia content with events. This mechanism has been evaluated on the MediaMill dataset for indexing video content with five sport events, providing promising results.

## Acknowledgment

This work was supported by the European Commission under contract FP7-248984 GLOCAL.

# 5. REFERENCES

[1] Basic geo (wgs84 lat/long) vocabulary. http://www.w3.org/2003/01/geo/, accessed 2010-04-07.

[2] Date and time formats. http://www.w3.org/TR/NOTE-datetime, accessed 2010-04-07.

[3] International Press Telecommunications Council. http://www.iptc.org, accessed 2010-04-07.

[4] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, Nov. 1983.

[5] P. K. Atrey. A hierarchical model for representation of events in multimedia observation systems. In *Proc. 1st ACM Int. Workshop on Events in Multimedia (EiMM '09)*, pages 57–64, Oct. 2009.

[6] L. Ballan, M. Bertini, A. D. Bimbo, and G. Serra. Video event classification using string kernels. *Multimedia Tools Appl.*, 48(1):69–87, May 2010.

[7] W.-H. Cheng, Y. Chuang, B.-Y. Chen et. al. Semantic-event based analysis and segmentation of wedding ceremony videos. In *Proc. Int. Workshop on Multimedia Information Retrieval (MIR '07)*, pages 95–104, Sept. 2007.

[8] R. Duda, P. Hart, and D. Stork. *Pattern Classification, (2nd ed.).* John Wiley & Sons, Inc., New York, USA, 2001.

[9] A. Francois, R. Nevatia, J. Hobbs, and R. Bolles. VERL: An ontology framework for representing and annotating video events. *IEEE Multimedia*, 12(4):76–86, Oct. 2005.

[10] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider. Sweetening ontologies with DOLCE. In *Proc. 13th Int. Conf. on Knowledge Engineering and Knowledge Management (EKAW '02)*, pages 166–181, London, UK, Oct. 2002.

[11] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):226–239, Mar. 1998.

[12] J. Lu, K. Plataniotis, and A. Venetsanopoulos. Face recognition using kernel direct discriminant analysis algorithms. *IEEE Trans. Neural Netw.*, 14(1):117–126, Jan. 2003.

[13] J. Molina, V. Mezaris, P. Villegas et. al. MESH participation to TRECVID2008 HLFE. In *Proc. TRECVID 2008 Workshop*, USA, Nov. 2008.

[14] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *Proc. 3rd Int. Conf. on Knowledge Representation and Reasoning*, pages 165–176, Jan. 1992.

[15] A. Scherp, T. Franz, C. Saathoff, and S. Staab. F–a model of events based on the foundational ontology dolce+dns ultralight. In *Proc. 5th Int. Conf. on Knowledge Capture (K-CAP '09)*, pages 137–144, Redondo Beach, California, USA, Sep. 2009.

[16] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, and I. Trancoso. Multi-modal scene segmentation using scene transition graphs. In *Proc. ACM Multimedia*, pages 665–668, Beijing, China, Oct. 2009.

[17] J. Smith, M. Naphade, and A. Natsev. Multimedia semantic indexing using model vectors. In *Proc. Int. Conf. on Multimedia and Expo (ICME '03)*, pages 445–448, Baltimore, MD, USA, July 2003.

[18] C. G. Snoek, M. Worring, J. C. van Gemert, J. Geusebroek, and A. W. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proc. ACM Multimedia*, pages 421–430, Oct. 2006.

[19] E. Tsamoura, V. Mezaris, and I. Kompatsiaris. Gradual transition detection using color coherence and other criteria in a video shot meta-segmentation framework. In *Proc. IEEE ICIP Workshop on Multimedia Information Retrieval (ICIP-MIR 2008)*, pages 45–48, San Diego, CA, USA, Oct. 2008.

[20] X. Wang and X.-P. Zhang. Ice hockey shot event modeling with mixture hidden markov model. In *Proc. 1st ACM Int. Workshop on Events in Multimedia (EiMM '09)*, pages 25–32, Oct. 2009.

[21] U. Westermann and R. Jain. Toward a common event model for multimedia applications. *IEEE Multimedia*, 14(1):19–29, Jan. 2007.

[22] J.M. Zacks, T.S. Braver, M.A. Sheridan et. al. Human brain activity time-locked to perceptual event boundaries. *Nature Neuroscience*, 4(6):651–655, June 2001.

[23] M. Zhu and A. Martinez. Subclass discriminant analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8):1274–1286, Aug. 2006.