XXIV Simpozijum o novim tehnologijama u poštanskom i telekomunikacionom saobraćaju – ***PosTel 2006****, Beograd, 12. i 13. decembar 2006.*

BRIDGING THE GAP: AUTOMATIC MULTIMEDIA CONTENT DESCRIPTION GENERATION

Andrej Košir
Faculty of Electrical Engineering, University of Ljubljana,
Tržaška 25, Ljubljana, Slovenia

Abstract: *Data mining procedures are indispensable tools in multimedia content retrieval, user modeling and user interface adaptation. They provide a variety of new functionalities of modern communication systems and are a hot research topic. There is a need for a design of procedures capable of automatic content descriptors generation based on multimedia content. A high-level multimedia content description is presented. As from the other end, a framework for low level image and video feature extraction is given. A procedure for bridging the gap between the low level and the high-level image and video content description is proposed and discussed.*

Keywords: *multimedia content descriptor, image and video feature extraction*

1. Introduction

The goal of this paper is to describe a procedure for generation of automatic multimedia content description based on low level image and video objects feature extraction. For instance, the identification of an actor in a given movie can be performed by locating and tracking human faces from the movie and recognizing their identities and comparing this information to database of movie actors. The performance of video feature extraction based procedures does not meet the requirements of much simpler tasks like simple video scene identification. Therefore, there is a gap between video feature extraction algorithms and requirements of multimedia content description data mining.

First, a high level content description based on a TV-anytime standard [1] is introduced. A low-level feature selection framework together with the methodology of extraction techniques validation procedure is presented. An optimal feature selection procedure is then given. Some conclusion remarks are added.

2. High level content description

Multimedia content description is based on semantic description of the material and is called high-level content description [2]. This is also to indicate the gap between semantic content description and visual data object identification called a low level content description. In order to encode the extracted image and video features as a high level multimedia content description, a multi-attribute content description is introduced.

Sets and maps introduced along with the terminology are based on TVAnytime metadata standard representation [1], but can be used for any multi-attribute description standard. The hierarchical taxonomy of the main attribute is recommendable but not mandatory.

A class of multimedia content items is denoted by H and multimedia content item is denoted by $h \in H$. Multimedia content item h is a given multimedia item, for instance a CD Bob Hoskins 's film.

Genre	Movies: drama
Title & Synopsis	Mona Lisa George, after getting out of prison, begins looking for a job, but his time in prison has reduced his stature in the criminal underworld…
Actors	Bob Hoskins, Cathy Tyson, Michael Caine, Robbie Coltraine, Clarke Peters, …
Director(s)	Neil Jordan
Country	UK
Production year	1986

Figure 1**:** An example of a metadata item

Each multimedia content item $h \in H$ possesses attributes from classes of attributes $A_1,...,A_n$. In order to maintain our formal presentation, for $1 \leq i \leq n$ the extraction of the attribute $a^i \in A_i$ is denoted by a map $md_i : H \to A_i$. For instance, the third attribute of the movie $h \in H$ is $\text{"Bob Hoskins"} = md_3(h) \in A_3$. Maps $md_1,...,md_n$ are coordinates of the map $md : H \to MD$, where $md(h) = (md_1(h),...,md_n(h))$ and $MD = A_1 \times \cdots \times A_n$ is a set of metadata items. To summarize, the metadata item $md(h) \in MD$ is a n-tuple of attributes of the multimedia content item $h \in H$. Each content item carries its own metadata denoted by and belongs to a certain genre denoted by $md_1(h) \in A_1 = V(T_G)$. An example of a metadata item is shown in Figure 1.

Hierarchical genre taxonomy is presented by a directed tree (for reference see [4]) $T_G = (V(T_G), E(T_G))$ of genres. The set of tree vertices is a set of genres $V(T_G) = \{g_1,...,g_m\}$. An example of a genre is movie. A hierarchical genre ordering is given by the set of directed tree edges $E(T_G) = \{(g_i,g_j) : g_i,g_j \in V(T_G)\}$, i.e. the directed edge (g_i,g_j) denotes that the genre g_j is a sub-genre of the genre g_i. Typically, the genre drama is a subgenre of the genre movie. Symbolically, $\text{"drama"} = g_j$, $\text{"movie"} = g_i$ and $\text{"drama is a subgenre of the genre movie"} = (g_i,g_j) \in E(T_G)$. An example of the genre hierarchy is presented in Figure 2.

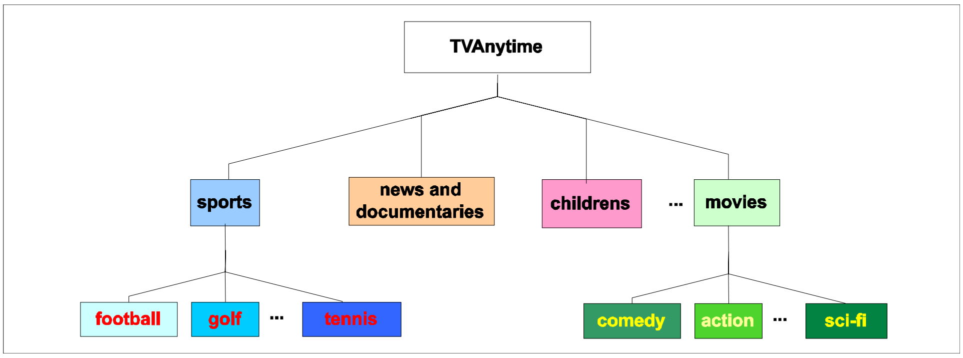

Figure 2: Example of the genre hierarchy

According to the standard [1], it is chosen $n = 7$ and the following attribute classes are used in this presentation: genre classification A_1, keywords (derived from the title and synopsis) A_2, actors A_3, director(s) A_4, country of origin A_5, year of origin A_6 and group CRID A_7. Consequently, the typical attribute values are: name of an actor a_1^1, a keyword of the title a_1^2, country of origin a_5^1, etc. The first class of attributes A_1 is a set of vertices of the genre hierarchy tree T_G, i.e. $A_1 = V(T_G)$.

3. Low level image and video feature extraction

Since visual data content analysis procedures in general do not meet the requirement of automatic metadata content description generation, the optimal image object feature procedure is of crucial importance here.

3.1. Basic concepts

Let X be an observation space. In our context, it is a space of features of image or video objects and is assumed to be a subset $X \subseteq \mathbf{R}^n$, see [3]. A computation of a feature vector $x \in X$ for a given input image or video $I \in \Im$ is denoted by a feature extraction map $f : I \mapsto x = f(I) \in X$. Coordinates of a feature vector are called components, $f(I) = (f(I)^1,...,f(I)^n)$ and coordinates of a feature map are called features, $f = (f^1,...,f^n)$.

Input information to the digital image and video object recognition system is a training set U. Typical examples of such objects are human faces to be recognized in the context of a video surveillance system. Assume our recognition system involves M known classes of objects. Their identities are denoted by a set of class labels $\Omega = \{\omega_1,...,\omega_M,\omega_{M+1}\}$ where the last label ω_{M+1} is assigned to the class of unrecognized objects. A set of recorded images (or video sequences) $S \subseteq \Im$ is available together with identities of objects captured, usually provided by a human expert. Images of objects labeled by ω_M are gathered into a subset $S_i \subset S$. This gives a partition of the set $S = \bigcup_i S_i$ and the training set equals to

$$U = \{(I, \omega_i) : I \in S_i, i \in L(M)\}.$$

To simplify the notation we denote the first k natural numbers by $L(k) = \{1,...,k\}$. According to the above construction, an observation space X is expected to be partitioned into M classes $C_1,...,C_M$, where the class C_i represents a class of objects labeled by ω_i. A feature space partition can be given as a decision rule

$$\delta : X \to \Omega,$$

where an object described by a feature vector $x \in X$ is adhered to the class C_i when $\delta(x) = \omega_i$. Clearly, the recognition system uses information contained in a training set U to determine (learn) a decision rule δ. The process of determining the decision rule is an example of a machine learning procedure. Since a set of classes is defined by a training set U, one expects if $x = f(I)$ for $I \in S_i$ then $\delta(x) = \delta(f(I)) = \omega_i$.

An event when an image I caries an object of the class ω_i is denoted by $[I \in S_i]$. By the construction it is clear that for $x = f(I)$ the following events are equal, $[I \in S_i] = [f(I) \in C_i] = [x \in C_i]$. An event when a feature vector x is ascribed to a class C_i is denoted by $[\delta(x) = \omega_i]$. Since the recognition system is not ideal, it may happen that $[x \in C_i] \neq [\delta(x) = \omega_i]$. When this happens, a classification error has been encountered. An event $[\delta(x) = \omega_i]$ when an event $x \in C_j$ is given (an image I represents an object that should be ascribed to C_j) is denoted by $[\delta(x) = \omega_i \mid x \in C_j]$. Indexes i and j can be or cannot be the same.

Assume $k \in L(M)$. An event $[\delta(x) = \omega_k \mid x \in C_k]$ is called *detection*, an event $[\delta(x) = \omega_k \mid x \notin C_k]$ is called *false alarm*, an event $[\delta(x) \neq \omega_k \mid x \in C_k]$ is called *miss* and an event $[\delta(x) \neq \omega_k \mid x \notin C_k]$ is called *correct dismissal*. Observe that all four events can be defined for each class C_k. An occurrence of the event miss $[\delta(x) \neq \omega_k \mid x \in C_k]$ is called an *error of the type I* and the occurrence of the event false alarm $[\delta(x) = \omega_k \mid x \notin C_k]$ is called an *error of the type II*. Their probabilities are denoted by $P_M = P[\delta(x) \neq \omega_k \mid x \in C_k]$ and $P_{FA} = P[\delta(x) = \omega_k \mid x \notin C_k]$. We also denote $P_D = P[\delta(x) = \omega_k \mid x \in C_k]$ and $P_{CD} = P[\delta(x) \neq \omega_k \mid x \notin C_k]$. Clearly it holds $P_M = 1 - P_D$ and $P_{FA} = 1 - P_{CD}$.

To illustrate the difference between two types of errors, assume we are dealing with a building security system and the distinguished class C_k is a class of persons that are allowed to enter the building. An error of the type I occurs when a person who should be allowed to enter is rejected and an error of the type II happens when a person who should be rejected is allowed to enter. In terms of security there is a significant difference between the mentioned events. In order to model the detection system one needs to control the probability of errors of both types.

3.2. A validation of the classification results

A classification procedure performs best when a feature vector $x = f(I)$ of any image $I \in \Im$ containing an object of a class C_i is classified in the class C_i. Assume the recognition system-learning phase based on the training set U has been performed. To test the performance of the system, a testing set of images U^T containing objects of interest is generated. Images are labeled by an expert (usually human), that is $U^T = \{(I,\omega_i) : I \in S_i^T, i \in L(M)\}$. Therefore, a set of images S_i^T contains a subset of test images containing an object of the class labeled by ω_i. Thus, a perfect classification procedure would give $[\delta(f(I)) = \omega_i]$ for any image $I \in S_i^T$. An important information about the performance of a given classification procedure is a table called a *confusion matrix*. For illustration purposes only, assume we have only $M = 3$ classes and the outcome of the classification procedure is given by Table 1.

	C_1	C_2	C_3
C_1	23	3	1
C_2	1	19	2
C_3	2	4	22

Table 1: An example of confusion matrix

The i-th row represents the outcome of the classification procedure on the feature vectors that should be classified to a class C_i. The matrix entry at the position ij is denoted by n_{ij}. In our example we have $|S_1^T| = n_{11} + n_{12} + n_{13} = 23 + 3 + 1 = 27$, $|S_2^T| = 1 + 19 + 2 = 22$, $|S_3^T| = 2 + 4 + 22 = 28$ and $|S^T| = 32 + 22 + 28 = 82$. There were $|S_2^T| = 22$ feature vectors that should be classified to the class C_2. But the confusion matrix shows that only 19 of them were classified to the class C_2, one was misclassified to the class C_1 and two were misclassified to the class C_3. In an ideal situation the confusion matrix is diagonal matrix.

The success rate is computed for each class C_i and denoted by $Sr(C_i) = \frac{n_{ii}}{\sum_j n_{ij}}$. For our example we calculate $Sr(C_2) = \frac{n_{22}}{n_{21} + n_{22} + n_{23}} = \frac{19}{22} = 0.86$. Moreover, for each class C_i the probability of an error of the type I P_M and the probability of an error of the type II P_{FA} can be estimated using, see [7],

$$P_M(C_i) = \frac{\sum_{j \neq i} n_{ij}}{\sum_j n_{ij}} = 1 - Sr(C_i), \qquad P_{FA}(C_i) = \frac{\sum_{j \neq i} n_{ji}}{\sum_j n_{ji}}.$$

Loosely speaking, to estimate the probability P_M one needs the row sums and to estimate P_{FA} one needs column sums of the confusion matrix. In our example we estimate $P_M = \frac{1+2}{1+19+2} = 0.14$ and $P_{FA} = \frac{3+4}{3+19+4} = 0.29$.

4. A feature selection procedure

Since we have agreed on the measurement of the quality of classification, we are ready to introduce the feature selection procedure. Assume we have a feature vector $x = f(I)$ computed using a feature extraction map $f = (f^1,...,f^m)$. The purpose of the feature selection procedure is to find a subset of indices $s = \{i_1,...,i_m\} \subset L(n)$ such that a reduced feature map $f = (f^{i_1},...,f^{i_m})$ provides best classification results in a certain prescribed sense. To simplify the notation, a reduced feature map is denoted by $s \circ f = (f^{i_1},...,f^{i_m})$ and a subset $s = \{i_1,...,i_m\}$ is called a selection. For instance, for a selection $s = \{2,4,5\}$ we have $s \circ f = (f^2, f^4, f^5)$. Clearly, for $s = L(n)$ it holds $s \circ f = f$.

The difficulty of the feature selection procedure lays in the fact that there are $2^n - 1$ of possible selections and the exhaustive search for the optimal one is not feasible. To overcome this problem, we introduce an *optimal feature selection task* and propose an algorithm to find such optimal selection.

To motivate our reasoning, assume we have two features f^1 and f^2 combined into a simple feature map (f^1, f^2) and only $M = 3$ classes C_1, C_2, C_3 to separate. A visual presentation of feature vectors $(f^1(I), f^2(I))$, $I \in S^T$ is grouped into clusters. This is an idealized situation.

A simple class presentation was implemented to compute the pertained confusion matrices for computed (f^1), (f^2) and (f^{i_1}), (f^{i_2}), respectively. All three classes are separable. When dealing with a single feature f^1 or f^2 each class is presented by an interval of admissible values for that feature. In the case of the feature map (f^1), (f^2), each class is presented by a Cartesian product of the mentioned intervals. A feature vector x is classified into a selected class if its values lie inside the assigned intervals. When there is an ambiguous situation, the selected class is chosen randomly. For instance, the feature f^1 cannot separate classes C_1 and C_2. When we have $I \in S_1$, it is clear that $f^1(I)$ is not in the classes C_3, but one cannot tell whether $f^1(I) \in C_1$ or $f^1(I) \in C_2$. This is an ambiguous situation and the selected class is chosen from C_1 and C_2 randomly.

4.1. A combination of extracted features

The theoretical analysis and experimental show that the above idealized situation summarize the real situation well in the aspect that a combination of carefully selected features yields the best results. First we define a measure of an efficiency of a given selection and a formula for determining the efficiency of an union of two selections follows naturally.

Recall the previously assumed notation; see Section 1 and Subsection 3.2. A measure of an efficiency of a feature selection s from a feature vector $(s \circ f)(I)$ is defined by

$$Sr(s,C_i) = \frac{n_{ii}}{\sum_j n_{ij}},$$

where $i \in L(M)$ and entries n_{ij} are taken from the confusion matrix of the feature selection $s \circ f$. This measure is defined for each class C_i. We use the notations $Sr(s \circ f,C)$ and $Sr(s,C)$ interchangeably.

According to the above reasoning, the efficiency of the feature map (f^1,f^2) can be estimated from the efficiencies of feature maps f^1 and f^2. It is given by $Sr((f^1,f^2),C_i) \geq max\{Sr(f^1,C_i),Sr(f^2,C_i)\}$ for $i = 1,2,3$. This relation can be generalized to the arbitrary chosen selections s_1 and s_2 as

$$Sr((s_1 \cup s_2),C_i) \geq max\{Sr(s_1,C_i),Sr(s_2,C_i)\},$$

where $i \in L(M)$.

4.2. Optimal feature selection algorithm

A heuristic algorithm for finding an approximate solution to the optimal feature selection task is introduced in this subsection. The idea of its construction is the following. To avoid an infeasible exhaustive search of optimal feature selection s ($2^n - 1$ possible selections), the efficiency of single features $Sr(f^j,C_k)$ is computed and the above-derived estimation is applied to estimate the success rate of an arbitrary chosen selection s. Since any selection can be decomposed to $s = \bigcup_{j \in s} \{j\}$, one can derive estimation

$$Sr(s,C_i) \geq max_{j \in s}\{Sr(f^j,C_i)\}.$$

A proposed heuristic is a simple greedy algorithm, given by the Algorithm 1.

Algorithm 1 Optimal feature selection algorithm

```
Reindex classes such that $|C_1| \geq |C_2| \geq \cdots \geq |C_M|$
Reindex feature vector coordinates such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$
```

```
$s = \{\ \}$
for ($k = 1,\dots,M$)
  while ($Sr(s,C_k) < r \wedge s \neq L(n)$)
    Find $i$ such that $Sr(f^i,C_k)$ is maximal
    $s = s \cup \{i\}$
  end
  if ($s = L(n) \wedge k \neq M$)
    Return "No solution"
  end
end
Return $s$
```

The algorithm requires a lower bound r on success rate and returns the selection s or the string "No solution" if there is no solution to the optimal feature selection task. When a selection s is returned, it is clear by that we have $Sr(s,C_k) > r$ for all $k \in L(M)$. There is absolutely no guaranty that the proposed greedy Algorithm 1 returns an optimal solution according to the specification of the optimal feature selection task given in the previous subsection.

5. Metadata content description generation

When at least some facts about a given multimedia content item h are recognized, its metadata descriptors $md(h)$ are filled according to gathered information. For demonstration purposes assume the multimedia content item $md(h)$ is a movie. Based on procedure techniques described in Section 2, certain characteristics like name of actors starring are established and its metadata $md(h)$ is filled, see [2].

Approach presented in this paper is in the development phase. The procedure of metadata content description is given by the Algorithm 2 in a rather loosely form. Prior to it a feature extraction maps is selected using Algorithm 2.

Algorithm 2 : Metadata content description generation procedure

1: Identify features of content item h .
2: Analyze identified features.
3: Retrieve data from metadata items database.
4: Analyze retrieved data based on a-priori known facts.
5: Write metadata content description $md(h)$.

The implementation of the procedure Algorithm 2 is quite content specific and we explain it in the context of the mentioned movie h metadata content description. We follow steps of the Algorithm 2.

1. Identify features of content item h **.** Different features of video frames of the movie h are extracted. Segmentation procedures are followed by face detection features to identify human faces and their properties. Texture classification techniques are applied to classify sub regions of frames.
2. Analyze identified features. Movie frames regions are classified as urban area, forest, field, roads, interior of a building etc. Specific objects like cars, trees etc are recognized. Human faces are registered and their characteristic features are stored. Basic statistics of mentioned features are computed and stored.
3. Retrieve data from metadata items database. Detected human faces are compared to the ones stored in the movie database and a list of most frequently encountered faces is made. Basic statistics of certain specific objects are related to the database entries. In this fashion, other available databases are used (containing cars etc.).
4. Analyze retrieved data. A list of candidates for actresses is made based on the list of recognized faces. According to data provided by the movie database, a list of possible movie titles is composed based on the actress appearances. A set of genres is generated according to the statistics of appearances of specific objects. A hierarchy of genres is constructed.
5. Write metadata content description $md(h)$ **.** According to the results of the previous step, an attempt of a unique movie title selection is made. If successful, a metadata content description is filled according to the available information in the movie database. If not successful, only determined metadata entries are stored.

It is somehow clear how to generalize the above-described procedure to the multimedia content items other than movie. Beside that, the outline given clearly indicates that the implementation of the procedure is content item specific in all phases. Video feature extraction techniques applicable here are dependent on what we expect the analyzed multimedia content item might be. The a-priori known facts stored in the items database are even more content specific. The same is true for the step 4. of the Algorithm 2. Above all, the multimedia content description standard, for example see [1], is clearly content specific.

6. Conclusion

A procedure for automatic multimedia content item metadata generation is proposed in the paper. It is in the early stage of development; merely a work plan. All elements of such procedure listed here seem required in every procedure for automatic metadata content description.

The efficiency of the proposed procedure is governed by efficiencies of the sub-procedures; the weakest ones are digital image and video content recognition algorithms. Formal system, see [5], can be applied to enhance image and video content understanding procedures. In the future, a development of more efficient visual data content understanding algorithms is necessary in order to broaden the applicability of automatic content description procedures.

References

[1] TVAnytime: "*Specification series: S-3 metadata, SP0003 v1.3, part A"*, ftp://tva:tva@ftp.bbc.co.uk/pub/Specifications/SP003v13.zip, 2002.

[2] Pogačnik, M., Tasič, J., Košir, A. "*Optimization of multi-attribute user modeling approach*", AEÜ, Int. j. electron. commun., vol. 58, no. 6, 2004, pp. 402-412.

[3] Theodoridis, S., Kouroumbas K., "*Pattern Recognition*", Academic Press, London, 1999.

[4] Korte, B., Vygen, J. "*Combinatorial Optimization*", Springer-Verlag, Berlin, 2000.

[5] Košir, A., Tasič, J. "*Formal system based on fuzzy logic applied to digital image scene analysis*", Proceedings of 10th IEEE Mediterranean Electrotechnical Conference MELECON'02, Cairo, Egypt, 2002, pp. 409-413.

[6] Hastie, T., Tibshirani, R., Friedman, J. "*The Elements of Statistical Learning*", Springer, New York, USA, 2001.

[7] Stark, H., Woods, J. W. "*Probability and Random Processes with application to Signal Processing*, Prentice Hall, New Jersey, 2002.

[8] Pogačnik, M., Tasič, J., Košir, A. "*Personal content recommender based on a hierarchical user model for the selection of TV programmes*", User model. User-adapt. interact. vol. 15, no. 5, 2005, pp. 425-457.

Sadržaj: *Procedure dejta majninga su neophodni alati pretraživanja multimedijalnih sadržaja, modelovanja korisnika i adaptacije na korisnički interfejs. One obezbeđuju različitost novih funkcionalnosti kod modernih komunikacionih sistema, i predstavljaju aktuelnu istraživačku oblast. Postoji potreba za projektovanjem procedura koje su u stanju da automatski generišu deskriptore multimedijalnog sadržaja. Pokazan je i postupak za ekstrakciju video oblika i slika niskog nivoa. Razmotrena je i diskutovana procedura za premošćavanje procepa između slika niskog i visokog nivoa, kao i opisa video sadržaja.*

Ključne reči: *deskriptor multimedijalnog sadržaja, ekstrakcija slika i video oblika*

PREMOŠĆIVANJE PROCEPA: AUTOMATSKO GENERISANJE OPISA MULTIMEDIJALNOG SADRŽAJA

Andrej Košir