Use of Context in Automatic Annotation of Sports Videos

Ilias Kolonias, William Christmas, and Josef Kittler

Center for Vision, Speech and Signal Processing,
University of Surrey,
Guildford GU2 7XH, UK

{i.kolonias,w.christmas,j.kittler}@surrey.ac.uk
<http://www.ee.surrey.ac.uk/CVSSP/>

Abstract. The interpretation by a human of a scene in video material is heavily influenced by the context of the scene. As a result, researchers have recently made more use of context in the automation of scene understanding. In the case of a sports video, useful additional context is provided by formal sets of rules of the sport, which can be directly applied to the understanding task. Most work to date has used the context at a single level. However we claim that, by using a multilevel contextual model, erroneous decisions made at a lower can be avoided by the influence of the higher levels. In this work, we explore the use of a multilevel contextual model in understanding tennis videos. We use Hidden Markov models as a framework to incorporate the results of the scene analysis into the contextual model. Preliminary results have shown that the proposed system can successfully recover from errors at the lower levels.

1 Introduction

Constructing cognitive vision systems which can extract knowledge from visual data so as to derive understanding of the scene content and its dynamics, as well as to facilitate reasoning about the scene, has recently moved to the top of the research agenda of the computer vision research community. One of the many potential applications of such systems is to provide automatic annotation of videos so that the user can retrieve the desired visual content using iconic or linguistic queries in a user-friendly way. This would help to facilitate access to the huge quantities of video information currently stored in archives (and growing daily) and make its retrieval more efficient.

However, it is generally accepted that automatic analysis and interpretation of video is a challenging problem in computer vision for a number of reasons. First of all, there is no direct access to 3D information, as video material is invariably captured using a monocular camera. Second, as the camera is not under our control, we are not able to invoke the active vision paradigm which might help us to interpret a scene by directing the camera to observe each scene object from different views. Third, very little is typically known about the camera system that captured the data.

On the positive side, a video captures contextual information which can be made to play a crucial role in the cognitive process. The merit of contextual information in sensory data perception has been demonstrated in application domains such as optical character recognition (OCR) and speech processing. For instance, in the former case, sequences of interpreted characters must constitute valid words. Thus, neighbours (preceding and ensuing characters) convey crucial contextual information for the correct recognition of each character. Similarly, at the next level, language grammar provides context aiding the correct interpretation of words by restricting the admissible sequences of word types and form. At the language understanding level, the circumstances relevant to the subject of discourse furnish contextual clues enabling a text processing system to gain understanding of the content.

In contrast, the use of context in visual data processing is less prevalent. Historically, the focus has been on dealing with the 3D nature of objects under varying illumination which makes object recognition in computer vision a much harder problem than OCR or speech perception. Moreover, in classical computer vision scenarios, such as a robot assembly cell or an office, both spatial and temporal context are relatively limited. For instance, key, books, cups can be placed almost anywhere and the scenes are largely static. Consequently, there are hardly any scene evolution rules that would restrict scene events in a productive way to assist interpretation. The lack of motivation for contextual processing has limited the development of tools that could be used to model and exploit contextual information in a routine manner.

However, in a number of application domains addressed more recently the contextual information is much richer and can play an important role in scene interpretation. A typical example is sports videos where the scene evolution is governed by strict rules. In this paper we explore the role of context in scene interpretation, with a focus on tennis videos. We demonstrate that in this event-driven sport, the detection of visual events and highlights, as well as the symbolic interpretation of the state of play, can be aided by contextual decision making. We construct a multilevel contextual model of tennis game evolution based on the hidden Markov model approach. We then use this model to annotate a tennis video. We show that the model has the capacity to derive the correct interpretation of the scene dynamics, even in the presence of low-level visual event detection errors.

The paper is organised as follows. In Section 2 we review related work. The contextual model adopted is developed in Section 3. The resulting interpretation scheme is described in Section 4. The method is evaluated experimentally in Section 5. Conclusions are drawn in Section 6.

2 Related Work

In order to provide a high-level analysis of video material, there has to be some means of bridging the gap between the contextual information generated by the visual processing modules and the set of assumptions that humans make

when viewing the same material. In general, these assumptions are often fuzzy and ill-defined, which makes the bridging task difficult. However, as we have mentioned, in the case of sports material, we are helped by the existence of a set of well-defined rules that govern the progress of a sporting event; these rules can potentially provide strong additional context. For these reasons, Hidden Markov Models (HMMs) and other variants of Dynamic Bayesian Networks (DBNs) are often employed to bridge the semantic gap in sports video analysis. There have been several successful attempts of such use of HMMs in relation to the analysis and annotation of sporting events. Here we discuss some of this work, in particular relating to tennis [1, 2], snooker [3], Formula 1 racing [4] and baseball [5]. The hand gesture recognition system of [6] is also relevant.

- In [4] the authors are developing a complete cognitive audio/visual system for the analysis of videos of Formula 1 race events. They extract semantic information from both the audio and the visual content of the sequence, and annotate the material by detecting events perceived as highly important. For example they include visual events such as overtaking, and cars running off the road. In addition, as the sequences used came from off-air video, they also extracted and used textual information about the race, including the drivers' classification and times. Since off-air material was used, the audio part of the sequences was dominated by the race commentary; from that, features such as voice intensity and pause rates were used. Having extracted all of this information, the authors attempted to infer events of semantic importance through the use of Dynamic Bayesian Networks, inferring content semantics either by using audio and video information separately or combining this information in the temporal domain. Both approaches yielded promising results when tested on simple queries (for example finding shots in the Formula 1 race where a car runs out of the race track).
- In [1] Petkovic et al. use HMMs in order to classify different types of strokes in tennis games, using the body positions of the players. The hit types detected include fore-hands, back-hands, serves, etc. To do this, the authors initially segmented the players out of the background; then they used Fourier descriptors to describe the players' body positioning; finally they trained a set of HMMs to recognise each type of hit. The results of this work show that this method can be quite successful in performing the recognition task it was designed for.
- Kijak et al. [2] also use an HMM to classify a tennis game into these scenes: first missed serve, rally, replay, and commercial break. In this work they used the HMM to fuse both visual cues, including dominant colours, spatial coherency and camera motion activity, and audio cues, including speech, applause, ball hits, noise and music. A second level of HMM is also used in this method, to reflect the syntax of a tennis game.
- Video footage of snooker is used in the work of [3]. The white ball is detected and tracked, and its motion analysed to identify the strategy of the player. The table is partitioned into five strategic areas, which are used by a set of four HMMs to classify the play into five categories: shot-to-nothing,

in which a shot pots a single coloured ball, but with no further strategic benefit; building a break by keeping the white ball in the centre of the table; conservative play (similar to the shot-to-nothing, but with no coloured ball potted); escaping from a snooker; and a miss, in which no collision is detected. The maximum likelihood measures from each HMM are compared to generate the classification.

- The work of Chang et al. [5] describes a method to extract highlights from baseball games. They use a fusion of static and dynamic information to classify the highlights into home run, good hit, good catch and within-diamond play. The static information is represented as statistical modules in the form of histograms, that can be classified into seven different types of play. The dynamic information is in the form of a set of HMMs, one for each highlight type. Each HMM models the transitions between the types of play that are representative of the corresponding highlight.
- In [6], the authors use HMMs to analyse hand gestures. Their framework decomposes each complex event into its constituent elementary actions. Thus in one of the examples the authors have used, a gesture is broken down into simple hand trajectories, which can be tracked more successfully via HMMs. Then, they apply Stochastic Context-Free Grammars to infer the full gesture. Such a paradigm can be compared to a tennis match; if we consider all elementary events leading up to the award of a point in a tennis match to be the equivalent of the elementary gestures in this work, and the tennis rules related to score keeping as an equivalent of the grammar-based tracking of the full gesture the authors have implemented, we can easily see the underlying similarities between the authors' work and reasoning on tennis video sequences.

Although these examples are a small subset of the applications that these inference tools have been tested upon, in each case the HMM (or DBN) provides a useful mechanism for classifying a sequence of detected events with labels that have some semantic content. However we note that in general the HMMs are working at a single semantic level. Thus if mistakes occur, higher level information cannot be brought to bear to influence the decision making. (The exception is [2]), although it is not clear to what extent the higher-level HMM is being used to drive decision making in the lower level one.) In any sports activity, there exists a set of rules that, in the case of professional events, can more or less be relied upon. In particular, in the case of a professional tennis tournament there exists a rich and complex set of formal rules that can be structured in a hierarchical fashion. In the work of [4] these rules are not exploited at all, and in our own previous work, only rules at the lowest level, relating to the award of points, are used.

3 Contextual Model

As we have seen, a large body of work has been done in the area of creating decision-making schemes; one of the most important pieces of work has been the

introduction of Hidden Markov Models (HMMs) [7, 8]. A formal definition of a Hidden Markov Model would be that it is a doubly stochastic process where we can *only* observe the outcomes of *one* of the processes; the underlying stochastic process *cannot* be directly observed, but can only be *inferred* through the existing observations, as if it was a function of the latter. A typical example of an HMM could be to consider ourselves in a room and be told about the results of a die being rolled in another room; since we don't know *how* the die is rolled, we have to consider this procedure as a stochastic process – as is the outcome of the die roll itself. Since we only know the outcome of the latter process, the system (the roll and the result processes) is properly described by a Hidden Markov Model. The notation most commonly used in the literature for the parameters in HMMs is the following:

- N , the number of distinct states in the model. In the context of a tennis match, it could be a set of possible playing states within a point (like expecting a hit, point to one player etc.)
- M , the number of distinct observation symbols for each state, i.e. the alphabet size. In this context, this could include the set of possible events within a tennis match.
- τ , the length of the observation sequence – in the scope of this analysis, it would be the length of a play rally.
- i^t will be the hidden state of the HMM at time t . Hence, $1 \leq i_t \leq N$.
- $P(i^t = k)$ denotes the probability of the state having the label k at time t .
- O_t will be the observation symbol at time t – equivalent to ‘*what was observed at time t* ’

When using HMMs to perform inference on a given problem, their most useful feature is their ability to calculate the *most likely* state sequence from an observation sequence. This is done by applying the *Viterbi algorithm*[9, 10], an inductive algorithm based on dynamic programming.

The Markov property for the sequence of hidden states $i_1, \dots, i_\tau$ and the related observation sequence $O_1, \dots, O_\tau$ for any $\tau \geq 2$ can be stated as

$$P(i^\tau | i^1, \dots, i^{\tau-1}) = P(i^\tau | i^{\tau-1})$$

Assuming stationarity for i^τ , we can write, for $\tau \geq 2$ and $1 \leq k_1, k_2 \leq N$,

$$P(i^\tau = k_1 | i^{\tau-1} = k_2) \equiv P(k_1 | k_2)$$

Now, the previous state of the process is characterised by the (known) probability measure $P(i^{\tau-1} | O_1, \dots, O_{\tau-1})$. Thus the *a priori* probability at time τ , which is $P(i^\tau | O_1, \dots, O_{\tau-1})$, will be given by

$$P(i^\tau = k_1 | O_1, \dots, O_{\tau-1}) = \sum_{k_2=1}^N P(k_1 | k_2) \cdot P(i^{\tau-1} = k_2 | O_1, \dots, O_{\tau-1})$$

If we also assume that that $O_1, \dots, O_\tau$ are conditionally independent, ie. that

$$P(O_\tau | i^\tau, O_1, \dots, O_{\tau-1}) = P(O_\tau | i^\tau)$$

the *a posteriori* probability at time τ will be given by

$$P(i^\tau | O_1, \dots, O_\tau) = \frac{P(i^\tau | O_1, \dots, O_{\tau-1}) \cdot P(O_\tau | i^\tau)}{\sum_{k=1}^N P(O_\tau | i^\tau = k) \cdot P(i^\tau = k | O_1, \dots, O_{\tau-1})}$$

This probability will be the optimal decision criterion at time τ and, at the same time, the *a priori* probability for time $\tau + 1$.

The Viterbi algorithm is appropriate for discovering the sequence of states that will yield a *minimum* path cost for the sequence. However, this is the case where we *only* want to extract the current state by *only* looking at the current observation symbol. This approach is bound to yield a large number of errors if faced with erroneous input data; therefore, we will need to compensate for that fact in some way if we are to use HMMs in a real-world system. There are two main methods of achieving this:

- Using multiple hypotheses for the evolution of a given sequence – in this case, we allow more than one event sequence to develop and be updated simultaneously, and we *eliminate* those where a transition from its previous state to its current one is *not allowed*.
- Directly taking under consideration the fact that the system will be using more than one observation to perform inference for a given event while developing the appropriate mathematical formulation of this problem. That means that we need to assume that the Markov processes that describe the problem can use the subsequent event in order to decide whether the current one is valid or not. This mode of decision is known in the literature as the *look-ahead ahead decision mode for a Markov chain* and is analysed in detail in [11]. In our context, *both the observation* (ie. the game event) *and the state* (ie. state of play) sequences would fall into this category.

The baseline technique can be generalised so as to also include *future* observation patterns, like $O_{\tau+1}$, in a similar way it can be generalised for higher-order Markov sources. Let us assume that, in time τ , a decision has to be made on classifying an input pattern O_τ . The decision can then *be postponed* until the *next pattern*, $O_{\tau+1}$, is also observed. Therefore, the probability required now is $P(i^\tau | O_1, \dots, O_\tau, O_{\tau+1})$. For that to be calculated, $P(i^\tau | O_1, \dots, O_\tau)$ is also required; this is calculated as we have previously seen. Therefore

$$P(i^\tau | O_1, \dots, O_\tau, O_{\tau+1}) = \frac{P(i^\tau | O_1, \dots, O_\tau) \cdot P(O_{\tau+1} | i^\tau)}{\sum_{k=1}^N P(O_{\tau+1} | i^\tau = k) \cdot P(i^\tau = k | O_1, \dots, O_\tau)}$$

and, since $O_1, \dots, O_{\tau+1}$ are independent,

$$P(i^\tau | O_1, \dots, O_\tau, O_{\tau+1}) = \frac{P(i^\tau | O_1, \dots, O_\tau) \cdot P(O_{\tau+1} | i^\tau)}{\sum_{k_1=1}^N \sum_{k_2=1}^N P(O_{\tau+1} | i^{\tau+1} = k_1) \cdot P(k_1 | k_2) \cdot P(i^\tau = k_2 | O_1, \dots, O_\tau)}$$

4 Proposed Scheme

In our context (the analysis of tennis video sequences), the rules of the game of tennis provide us with a very good guideline as to what events we will have to be capable of tracking efficiently, so as to follow the evolution of a tennis match properly. Such events would include:

- The tennis ball being hit by the players
- The ball bouncing on the court
- The players' positions and shapes (that is, body poses)
- Sounds related to the tennis match (either from the commentary or the court)

These events can be used as a basis on which to perform reasoning for events of higher importance, like awarding the current point from the events witnessed during play. Based on them, the full graphical model for the evolution and award of a point in a tennis match is illustrated in the graph of Figure 1.

Fig. 1. Graphical model for awarding a point in a tennis match

As we can readily see from this diagram, it is a graphical model where a number of loops exist; the state transitions drawn with **bold** lines indicate where these loops close. Moreover, this graphical model *only* tackles the problem of awarding a single point in the match; there is some more detail to be added to it if we wish to include the awarding of games, sets or the full match. How these stages are going to be implemented will be discussed in more detail later in this section. Finally, this figure also shows us that, in order to address the problem

of ‘understanding’ the game of tennis more effectively, we will have to convert this complex evolution graph into a set of simpler structures.

Thus, we propose to replace the original scene evolution model with a set of smaller models, each one trying to properly illustrate a certain scenario of the match evolution. The most important thing we need to ensure during this procedure is that, when we combine all of the models in this set, we *must* have a model equivalent to the original one (so that it reflects the rules of tennis). The set of sub-graphs proposed to replace the original one is illustrated in Figure 2.

Fig. 2. Switching model and its respective set of sub-models for awarding a point in a tennis match, as separated out from the original graphical model

As we can see from the set of models above, we have opted for a more ‘perceptual’ way of selecting the set of event chains that will formulate the new model set for our purposes. This design strategy is going to be particularly helpful in the (quite frequent, as it is expected) cases where the system receives a query to extract sequences which contain some specific event; since the scene description will consist of a series of events occurring in the match, such queries

can be dealt with relatively easily. Moreover, choosing to break the initial graph down to a number of sub-graphs and train each one of them separately will be beneficial in many more different ways. Some of the resulting benefits are:

- Using probabilistic reasoning tools (like HMMs) will allow for correction of wrongly detected elementary events within the game – so we can have a point correctly awarded even if we haven’t tracked *all* events accurately from the start.
- Since the models are simpler, we will *not* have to acquire a huge training data set; a relatively small amount of data per model will suffice for our analysis. This will also make the training procedure a *much less* computationally expensive process as well – due to both the reduced volume of the training data and the simplicity of the models as well.
- In some cases, we can *considerably speed up* the training process due to prior knowledge for this type of content. For example, the amount of statistics available for events occurring in a tennis match helps us get a very good initial estimate for the HMM parameters *without* the need of a training process; we only need to pick up some existing measurements for this purpose.
- It will be *easier* to determine which areas (ie. sub-models) of the reasoning engine need to be improved to boost the overall performance of the system and which low-level feature extraction modules are more suspect to produce misleading results – so that we can either improve them or replace them with a different approach.

Since the detection of such elementary events will be done using machine vision algorithms and techniques, we are bound to encounter event detection errors in the process. Hence, another crucial issue is to examine the robustness of the proposed scheme to false events in its input. To address this, it has been decided that the system will be implemented through the use of *Hidden Markov Models with a look-ahead decision mechanism*, therefore allowing us to take into account events that occur *after* the one currently examined. This will help the system establish whether the current event can actually occur or it needs to be corrected and re-examined, as the rules of tennis will allow us to detect and correct contradictory hypotheses. The length of the look-ahead window has been limited to 1 event forward, thus allowing us to correct isolated errors – if we encounter 2 or more consecutive errors, the output will generally *not* be correct. However, there are two reasons for this choice:

- If the length of the look-ahead window is too large, small rallies (like aces) with errors in them may be wrongly interpreted. That can happen as, at the end of the chain, the Viterbi algorithm will *not* have enough evidence to correct errors. For cases like this, we will need to implement an error-correcting scheme on top of the proposed method, that will rely on different context – as will be discussed later on.
- Out of the events required for correct evolution tracking in this context, the type of events that is *clearly* most susceptible to errors is the ball bounces – the difficulty of tracking a ball in a tennis sequence stemming from the fact

that it is a very small, fast-moving object. However, since *only one bounce* can occur between two successive player hits if the point is still to be played, detecting a player hit (*not* a serve) automatically removes the chance of a point being awarded, even if the bounce point is wrongly detected (or not detected *at all*).

This graph will be implemented through using a ‘*switch*’ variable to select which of the constituent sub-models will be used to model the scene at a given moment. Therefore, the proposed system’s structure is similar to a Switching HMM. As soon as we have determined which way the points are going to be awarded, we can move on to the award of games and sets in the match. This can either be done through the use of probabilistic reasoning tools (such as HMMs), or with simpler, rule-based tools – such as grammars. The latter is possible due to the fact that at this level of abstraction in modelling tennis game sequences, it is the rules of the game of tennis that stipulate the evolution of the scene rather than low-level events in it. However, since the uncertainty stemming from the successful (or unsuccessful) detection of the elementary events mentioned above *cannot always* be considered to have been effectively addressed in the lower-level stages of the reasoning process (ie. up to the level of point awarding), we will *have to* opt for using probabilistic tools to perform higher levels of reasoning for these video sequences. However, the fact that these models will *directly* correspond to the rules of tennis and that most of the spurious low-level data have already been corrected will allow for far greater confidence in the output of the higher-level system.

5 Results and Discussion

The scheme described above has been tested on one hour’s play from the Men’s Final of the 2003 Australian Tennis Open. The sequence contained a total of 100 points played – equivalent to approximately one and a half sets of the match. Out of these 100 exchanges, a total of 36 were played on a second serve. The data that was used as input in this experiment were *ground-truth*, *hand-annotated event chains* from the broadcast match video in the first case, and the same data with *one* error per point sequence randomly inserted into it in the second. Therefore, we have examined *both* the ability of the proposed scheme to model the evolution of the match accurately *and* the robustness of it at the same time. To do that, we had to introduce four sets of models – one for every combination of which player serves and which side of the court he/she serves from (left or right). Moreover, in the second case, we limited the system to *only* check the next event for error correction. In those 100 exchanges, we have intentionally left in a few *unfinished* points, so as to examine whether the selection of the hidden states for these models can lead to an accurate representation of the scene at *any* given time – *not only* at the end of the scene. They were 4 in total – 2 leading to a second serve and 2 were cut short while still on play. An overall view of the results is given in Tables 1 and 2 below.

Table 1. Total results for Ground-Truth Data

	Ground Truth	Correctly Awarded	Wrongly Awarded	Not awarded (still on play)
Near Player Points	59	59	0	0
Far Player Points	37	37	0	0
Unfinished Points	4	4	0	n/a
TOTAL	100	100	0	0

Table 2. Total results for Data with Errors Inserted

	Ground Truth	Correctly Awarded	Wrongly Awarded	Not awarded (still on play)
Near Player Points	59	57	0	2
Far Player Points	37	35	1	1
Unfinished Points	4	3	1	n/a
TOTAL	100	95	2	3

As we can see in Table 1, *all* of the points were successfully tracked by the proposed system when ground-truth data were used, whereas in the case of errors in the data set (shown in Table 2), we did have some errors in recognition. This happened because of the fact that, in those points, the *very last* event on the chain was wrong – and since there was no ‘future’ to the event sequence after that event, the system had no way of inferring that this was wrong. However, this is a predictable problem, because this event is the deciding one for the point and the uncertainty in it *cannot* be removed; therefore, the uncertainty will propagate to the decision *for this level* as well. This observation shows that the application of a similar error-correcting scheme on top of the proposed system (in which point award is to be processed) will further enhance the robustness of the system.

6 Conclusions

As we can readily see from the results shown above, the proposed system has tackled the problem of tracking the evolution of a tennis match very effectively. Whereas in the case where no error input has been provided, such accuracy serves only to illustrate that the use of such an approach is not inherently wrong; it is the performance of the proposed scheme when errors were added in the observation sequences that makes it seem a very promising solution for a fully automated evolution tracking system for tennis videos.

However, there are still some issues to be addressed in this area. First of all, the aim of developing such an automatic evolution tracking scheme is for use as an automatic video annotation system, in conjunction with a set of fully automatic low-level feature extraction tools that will be able to detect the basic events required for input to the proposed system. As in any fully automatic computer vision system, the low-level feature extraction tools are bound to produce recognition errors which will propagate to the proposed decision-making

scheme. Therefore, and although some testing has already been carried out for the proposed scheme, it is essential that the proposed scheme is also tested with the actual low-level feature extraction tools integrated into it, so that we can have a clearer view of what errors occur in the lower levels and provide accurate statistics about the performance and robustness of the higher levels of the inference engine as a whole.

Moreover, the proposed system is only the first step in a hierarchical model that will fully describe the evolution of a tennis match – it will only cover the award of a single point in the match. The creation of a full system will require the design of a similar error-correcting scheme to award games, sets and finally the match – and which will all rely on the efficiency of this method.

Acknowledgements

This work has been supported by the EU IST-2001-34401 Project VAMPIRE.

References

1. Petkovic, M., Jonker, W., Zivkovic, Z.: Recognizing strokes in tennis videos using Hidden Markov Models. In: Proceedings of Intl. Conf. on Visualization, Imaging and Image Processing, Marbella, Spain. (2001)
2. Kijak, E., Gravier, G., Gros, P., Oisel, L., Bimbot, F.: HMM based structuring of tennis videos using visual and audio cues. In: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'03). (2003)
3. Rea, N., Dahyot, R., Kokaram, A.: Modeling high-level structure in sports with motion-driven HMMs. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004), Montreal, Quebec, Canada (2004)
4. Petkovic, M., Mihajlovic, V., Jonker, W., Djordjevic-Kajan, S.: Multi-modal extraction of highlights from TV Formula 1 programs. In: Proceedings of the IEEE International Conference on Multimedia and Expo. Volume 1. (2002) 817–820
5. Chang, P., Han, M., Gong, Y.: Extract highlights from baseball game video with hidden Markov models. In: Proceedings of the International Conference on Image Processing (ICIP'02), Rochester, New York (2002)
6. Ivanov, Y., Bobick, A.: Recognition of Visual Activities and Interactions by Stochastic Parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** (2000) 852–872
7. Rabiner, L.R., Juang, B.H.: An introduction to Hidden Markov Models. *IEEE Signal Processing Magazine* **61** (1986) 4–16
8. Dugad, R., Desai, U.B.: A tutorial on Hidden Markov Models. Technical Report SPANN-96-1, Signal Processing and Artificial Neural Networks Laboratory Department of Electrical Engineering Indian Institute of Technology – Bombay Powai, Bombay 400 076, India (1996)
9. Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory* **13** (1967) 260–269
10. Forney, G.D.: The Viterbi algorithm. *Proceedings of the IEEE* **61** (1973) 263–278
11. Devijver, P.A., Kittler, J.: *Pattern Recognition – A Statistical Approach*. Prentice Hall International, London, UK (1982)