Real-time Hypothesis Driven Feature Extraction on Parallel Processing Architectures

Ole-Christoffer Granmo
Simula Research Laboratory
P.O. Box 134, N-1325 Lysaker, Norway

Finn Verner Jensen
Department of Computer Science
Aalborg University
Fredrik Bajers V. 7, DK-9220 Aalborg, Denmark

Abstract

Feature extraction in content-based indexing of media streams is often computational intensive. Typically, a parallel processing architecture is necessary for real-time performance when extracting features brute force. On the other hand, Bayesian network based systems for hypothesis driven feature extraction, which selectively extract relevant features one-by-one, have in some cases achieved real-time performance on single processing element architectures. In this paper we propose a novel technique which combines the above two approaches. Features are selectively extracted in parallelizable sets, rather than one-by-one. Thereby, the advantages of parallel feature extraction can be combined with the advantages of hypothesis driven feature extraction. The technique is based on a sequential backward feature set search and a correlation based feature set evaluation function. In order to reduce the problem of higher-order feature-content/feature-feature correlation, causally complexly interacting features are identified through Bayesian network d-separation analysis and combined into joint features. When used on a moderately complex object-tracking case, the technique is able to select parallelizable feature sets real-time in a goal oriented fashion, even when some features are pairwise highly correlated and causally complexly interacting.

Keywords: Real-time Content-based Indexing of Media Streams, Bayesian Networks, Hypothesis Driven Feature Extraction, Parallel Feature Extraction, Bayesian Network D-separation Analysis

1 Introduction

The technical ability to generate volumes of digital media data is becoming increasingly “main stream” in todays electronic world. To utilize the growing number of media sources, both the ease of use and the computational flexibility of methods for content-based access must be addressed; e.g. an end-user may want to access live content automatically under a variety of processing environments ranging from complex distributed systems to single laptops.

In order to make media content more accessible, pattern classification systems which automatically classify media content in terms of high-level concepts have been taken into use. Roughly stated, the goal of such pattern classification systems is to bridge the gap between the low-level features produced through signal processing (feature extraction) and the high-level concepts desired by the end-user. Automatic visual surveillance [1], indexing of TV Broadcast News [2] and remote sensing image interpretation [3] are examples of popular application domains. For instance, in [2], a TV Broadcast News video stream is automatically parted into segments such as Newscaster, Report, Weather Forecast and Commercial. Thereby, the video stream is indexed and an end-user can access the different types of segments directly, without skimming through the complete video stream.

Typically, the feature extraction is computationally much more intensive than the actual classification as feature extraction may involve costly on-line signal processing of multiple features. In contrast, a pattern classifier is typically trained off-line such that efficient classification can be achieved on-line.

There are two common approaches to handling the high computational cost of feature extraction. Firstly, features can be extracted in *parallel* by taking advantage of a parallel processing architecture. In [4] a multi agent based system for coarse-grained distribution of feature extraction is presented. Another framework for coarse-grained distribution of feature extraction (and classification) is the DMJ framework [5]. In the DMJ framework, feature extraction is distributed as a set of interacting components, logically organized in a processing hierarchy. A fine-grained solution for parallel feature extraction is described in [6] where the feature extraction is distributed in a hypercube multicomputer network.

Secondly, features can be extracted selectively, on a per need basis, in order to minimize feature extraction cost while maximizing the classification certainty (*hypothesis driven* feature extraction). In [7] a framework for visual surveillance is proposed. This framework selectively extracts features in order to focus the processing resources on relevant features. Likewise, hypothesis driven feature extraction is used in [3] to reduce the number of information sources required for reliable classification in remote sensing image interpretation. Finally, a technique for real-time hypothesis driven feature extraction in continuous media streams is described in [8].

By combining the above two approaches it should be possible to improve the performance of content-based media indexing systems. Only relevant features (at a given point in a media stream) need to be extracted, and the relevant features can be extracted in parallel, on multiple processing elements. However, an examination of traditional methods for hypothesis driven feature extraction reveals that they do not scale up to handle some of the main difficulties of real-time hypothesis driven parallel feature extraction:

- the set of features to be extracted in parallel must be selected from a very large number of possible feature sets (exponential in the number of features),
- when evaluating a candidate feature set, the redundancy of the features (when it comes to predicting the media content) must be considered,
- due to higher-order feature-content/feature-feature correlation, it may be difficult to evaluate the combined media content prediction capability of multiple features.

In this paper we propose a novel technique for hypothesis driven feature extraction in Bayesian networks [9] which targets the above difficulties. Bayesian networks are a general approach to probabilistic inference which are used with success in recent advanced content-based media indexing systems such as [1] and [10]. Our technique extends traditional methods for hypothesis driven feature extraction in Bayesian networks to take advantage of a parallel processing architecture.

2 Bayesian Networks

Bayesian networks (BNs) are used extensively in the current work, and we shall here give a short introduction. For a more thorough treatment, readers are referred to the literature (for example [9] and [11]).

A Bayesian network consists of a qualitative and a quantitative part. The qualitative part is a directed acyclic graph (DAG). The nodes are variables, and the directed links represent causal impact between variables. Figure 1 shows an example network. The node C represents the content of

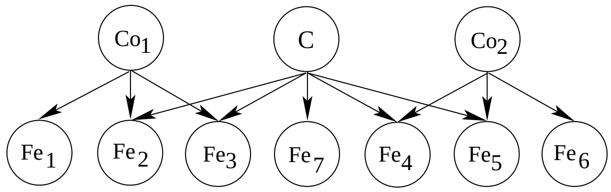

Figure 1. *The structure of a Bayesian network for identifying the content, C, of an image. The state of the environment conditions Co_1 and Co_2 have an impact on the features Fe_2, Fe_3, Fe_4, Fe_5. The nodes Fe_1 and Fe_6 represent observations which may give evidence on the state of the environment conditions.*

an image. In other words, C is a variable with possible states (c_1, c_2, c_3). The nodes $Fe_1, \ldots, Fe_7$ represent seven different features, each with two outcomes (y, n). Finally, the two nodes Co_1 and Co_2 represent two different types of environment conditions, which each can be in two different states $(positive, negative)$. The state of the feature Fe_7 is a result of the content C and there is a directed link from C to Fe_7. The features Fe_1 and Fe_6 are observations which may give evidence on the two environment conditions, and therefore there are directed links (Co_1, Fe_1) and (Co_2, Fe_6). The links to the features Fe_2, Fe_3, Fe_4, Fe_5 indicate that the feature outcome is a result of the content as well as the state of the environment conditions.

The strength of the directed links are represented as conditional probabilities. So, the quantitative part of a Bayesian network consists of a set of conditional probabilities. For each node A with parents $pa(A)$, we shall specify the conditional probabilities $P(A|pa(A))$. If $pa(A) = \emptyset$ we specify the prior probability distribution $P(A)$. For the network in Figure 1 we have to specify $P(C)$, $P(Co_1)$, $P(Co_2)$, $P(Fe_1|Co_1)$, $P(Fe_2|Co_1, C)$, $P(Fe_3|Co_1, C)$, $P(Fe_4|Co_2, C)$, $P(Fe_5|Co_2, C)$, $P(Fe_6|Co_1)$, $P(Fe_7|C)$. In Table 1 a specification of $P(Fe_2|Co_1, C)$ is given.

Table 1. *The conditional probabilities $P(Fe_2 = y|Co_1, C)$. Note that a distribution of Fe_2 is specified for each configuration of the parents.*

	$C = c_1$	$C = c_2$	$C = c_3$
$Co_1 = positive$	0.9	0.1	0.5
$Co_1 = negative$	0.6	0.4	0.5

Bayesian networks are mainly used for calculating posterior probabilities. When some nodes in the network are observed, the state of them can be entered to the network, and the posterior probabilities for the remaining variables are calculated. For example, if Fe_1 is observed to be y, and Fe_3 is observed to be n, then the posterior distribution $P(C|Fe_1 = y, Fe_3 = n)$ for C can be calculated as well as the posterior distributions for Co_1, Co_2, Fe_5 and so forth.

Because the links have a causal interpretation, the structure of a Bayesian network can be used to analyze structural independence. For example, for the network in Figure 1, it can be seen that information on the state of Fe_1 can have an impact on our belief of whether the environment condition Co_1 is positive or negative, but this information does not give any indication on the content of the image. On the other hand, if the feature Fe_3 has been observed, then an observation of Fe_1 may give indications of C. This type of analysis is called *d-separation analysis*, and the general rule is the following:

Definition Two distinct nodes A and B are d-separated if for each path between A and B there is an intermediate node C such that either the connection is serial or diverging and the state of C is known, or the connection is converging and neither C nor any of C's descendants have received evidence (see Figure 2).

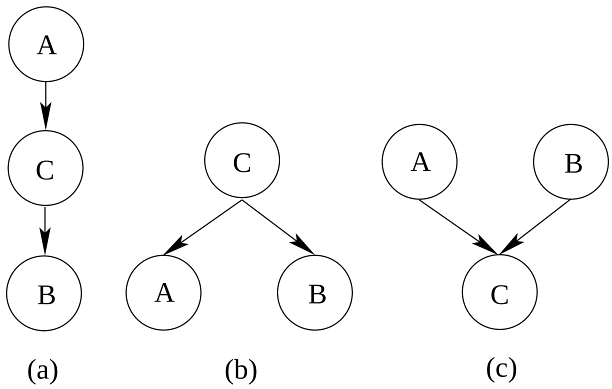

Figure 2. *(a) a serial connection, (b) a diverging connection, (c) a converging connection.*

For example, for Figure 1 we have that Fe_1 and C are d-separated, Fe_2 and Fe_4 are d-separated given C; but Fe_1 and C are not d-separated given Fe_2. The crucial property is that d-separated variables are independent.

3 Hypothesis Driven Feature Extraction

Traditional myopic methods for hypothesis driven feature extraction based on Bayesian networks, such as [12], extract features sequentially, one-by-one, in decreasing order of efficiency (hypothesis information gain divided by feature extraction cost). The feature extraction stops when all features have been extracted, the confidence in a hypothesis is sufficiently high or the hypothesis information gain of each of the remaining features do not justify their extraction cost (their efficiency is below a predefined threshold). By following this procedure the total feature extraction cost can often be reduced while the classification certainty is maintained at an acceptable level. In this section we give a formal description of the traditional myopic hypothesis driven feature extraction procedure, before we discuss its shortcomings when applied to parallel feature extraction.

At each iteration of the traditional myopic hypothesis driven feature extraction procedure, the *entropy* of the content distribution, $P(C)$, is used to measure the current content (hypothesis) uncertainty:

$$H(C) = -\sum_{c \in C} P(c) log[P(c)].$$

The *expected entropy* of the content distribution is used to measure the content uncertainty to be expected after extracting a feature F:

$$H(C|F) = -\sum_{f \in F} P(f) \sum_{c \in C} P(c|f) log[P(c|f)].$$

Finally, the *Mutual Information Gain* (MIG) divided by the feature extraction cost is used as a feature efficiency measure:

$$Ef(F) = \frac{H(C) - H(C|F)}{F_{cost}}.$$

In short, the feature with the highest efficiency (given already extracted features) is extracted at each iteration.

Let c_{max} be the most probable state of C given extracted features ε:

$$c_{max} = \operatorname*{argmax}_{c \in C} P(c|\varepsilon).$$

c_{max} is given as output and the hypothesis driven feature extraction procedure stops when all features have been extracted, the probability $P(c_{max}|\varepsilon)$ is sufficiently high, or the highest feature efficiency falls below a predefined threshold.

We now consider the limitations of the straightforward and naive approach for extending the above procedure to select a set $S = \{F_1, \ldots, F_n\}$ of n features at each iteration, rather than a single feature. Namely, the most efficient feature set can be found by calculating the efficiency $Ef(S)$ of all possible feature sets of size n. This approach has two significant limitations. Firstly, the number of feature sets to be considered is very large. Secondly, the number of calculations required to calculate the expected entropy of a feature set (which is necessary to calculate the MIG) grows exponentially with the number of features, $F_1, \ldots, F_n$, in the set:

$$H(C|\bar{F}) = -\sum_{\bar{f} \in F_1 \times \cdots \times F_n} P(\bar{f}) \sum_{c \in C} P(c|\bar{f}) log[P(c|\bar{f})].$$

E.g. if the features are binary the above calculation is of order 2^n.

Obviously, you could just calculate the efficiency of each feature individually and then pick the n best, but this approach has an important disadvantage: correlated features which are efficient individually may be inefficient as a set (the feature extraction cost is increased while the MIG remains the same), and some individually inefficient features may be very efficient in combination. As an example of the latter, initially the feature Fe_1 from Figure 1 only gives information about the state of the environment condition Co_1 and no information about the content C. Likewise, the feature Fe_2 may give no information about the state of Co_1, and only give information about the content C if the environment condition is known. As a result, the features are inefficient individually and efficient in combination.

Apparently, another technique for evaluating and searching for an efficient feature set is needed.

4 Hypothesis Driven Parallel Feature Extraction

In this section we propose a technique for hypothesis driven parallel feature extraction. Features are extracted sequentially in parallelizable sets, rather than one-by-one. Thereby, the advantages of parallel feature extraction can be combined with the advantages of hypothesis driven feature extraction. The technique is based on a sequential backward feature set search and a correlation based feature set evaluation function. In order to reduce the problem of higher-order feature-content/feature-feature correlation, causally complexly interacting features are identified through Bayesian network d-separation analysis and combined into joint features.

4.1 Correlation Based Feature Set Evaluation

In [13] a correlation based feature set evaluation function, the CFS-function, is introduced within the area of off-line feature selection in machine learning. The CFS-function measures the prediction capability of a feature set $S = \{F_1, \ldots, F_n\}$ with respect to a hypothesis H, based on the first-order correlation between each feature and the hypothesis and between pairs of features. The correlation is measured in terms of the Symmetrical Uncertainty Coefficient (SUC):

$$SUC(X, Y) = 2 \times \frac{H(Y) + H(X) - H(X, Y)}{H(Y) + H(X)}$$

which normalizes the MIG into the range $[0, 1]$. The CFS-function relates the pairwise feature-feature and feature-hypothesis correlations as follows:

$$CFS(S) \equiv \frac{\sum_i SUC(F_i, H)}{\sqrt{\sum_{ij} SUC(F_i, F_j)}}.$$

The numerator in the CFS-function is simply the sum of the individual feature-hypothesis correlations. This feature-hypothesis correlation sum is weighted by the square root of the sum of feature-feature correlations in order to reflect feature redundancy; features may give similar information about the hypothesis when they are correlated. For instance, given n features, if the pairwise feature-feature correlation is 1 for each of the n^2 feature pairs, the feature-hypothesis correlation sum is divided by n. On the other hand, if there are no feature-feature intercorrelations, the feature-hypothesis correlation sum is divided by $\sqrt{n}$. In other words, if the sums of the individual feature-hypothesis correlations are identical, the latter feature set has a higher prediction capability compared to the former feature set.

As the CFS-function only considers first-order (pairwise) correlation, the computational complexity is low compared to the MIG-function. On the other hand, the CFS-function is inferior to the MIG-function when it comes to representing the prediction capability of causally complexly interacting features. Despite this limitation, feature selection based on the CFS-function scores comparable to other state-of-the-art feature selection approaches on several feature selection benchmarks [13] (as long as higher-order correlation is limited). Due to its low computational complexity and due to its state-of-the-art performance on feature selection benchmarks, the CFS-function seems to be a good candidate for evaluating the prediction capability of a feature set in real-time hypothesis driven feature extraction. This is even more true if the problems caused by higher-order correlations can be reduced.

4.2 Higher-order Correlations

Due to the causal interpretation of links in a Bayesian network, features which are not d-separated when the state of the hypothesis is known and which in addition are connected to the hypothesis through at least one converging connection (see Section 2) may give rise to higher-order feature-feature correlations and higher-order feature-hypothesis correlations. For instance, the features Fe_1, Fe_2 and Fe_3 in Figure 1 are not d-separated when the state of C is known. Furthermore, Fe_1 and Fe_2 are connected to C through the converging connections in Fe_2 and Fe_3. Consequently, the causal interactions between Fe_1, Fe_2, Fe_3 and C may give rise to higher-order correlations.

Let the function $j(Fe_q, Fe_r)$ be *true* if and only if one of the following conditions hold

- Fe_q and Fe_r are not d-separated when the state of the hypothesis is known,

- $q = r$,
- $j(Fe_r, Fe_q) = true$,
- $\exists Fe_z, j(Fe_q, Fe_z) = true \wedge j(Fe_z, Fe_r) = true$.

Then j is an equivalence function which partitions features into equivalence classes. We define $J[S]$ to be a partitioning of the features in S, as given by j, where partitions not containing any feature connected to the hypothesis through a converging connection, are repartitioned into singletons. Roughly stated, features belonging to the same partition in $J[S]$ are causally complexly interacting with the hypothesis, whereas other features are not.

We reduce the problems caused by higher-order correlations by using $CFS(J[S])$ rather than $CFS(S)$ when evaluating a feature set S. For instance, rather than using the correlation measurements $SUC(Fe_1, Fe_1)$, $SUC(Fe_2, Fe_2)$, $SUC(Fe_3, Fe_3)$, $SUC(Fe_1, Fe_2)$, $SUC(Fe_1, Fe_3)$, $SUC(Fe_2, Fe_3)$, $SUC(Fe_1, C)$, $SUC(Fe_2, C)$ and $SUC(Fe_3, C)$ we treat Fe_1, Fe_2 and Fe_3 as a joint feature. Consequently, we only use the correlation measurements $SUC(\{Fe_1, Fe_2, Fe_3\}, \{Fe_1, Fe_2, Fe_3\})$ and $SUC(\{Fe_1, Fe_2, Fe_3\}, C)$ when calculating the CFS-function. Thereby, the identified higher-order correlations are taken into account. Note that this approach increases the computational complexity of the CFS-function exponentially in the number of features in a partition, and consequently is only computationally practical when the number of features in each partition is low.

4.3 Backward Feature Set Search

We now propose a sequential backward search procedure for identifying an efficient set of n features to be extracted in parallel, at a given step of the hypothesis driven feature extraction procedure defined in Section 3. It is assumed that the processing cost of features are similar, that the communication time of features are insignificant and that features always can be extracted in parallel. However, our approach can easily be extended to handle features with non-similar processing costs, significant feature communication time, and restrictions on which feature extractors can be executed in parallel, by using the ARCAMIDE search procedure we propose in [14].

The search is performed backwards (features are removed sequentially) in order to avoid a computationally expensive n-step look-ahead forward search, made necessary by causally complexly interacting features. For efficiency reasons the sequential backward search is done in two stages.

In the first stage a simplified CFS-function, CFS', is used to remove irrelevant features. The CFS-function is simplified by ignoring feature-feature intercorrelation:

$$CFS'(S) \equiv \sum_i SUC(F_i, H).$$

Consequently, the complexity of CFS' is linear in the number of features rather than quadratic. We use the CFS'-function until αn features remain. α is a constant which optimally should be equal to the number of features in the largest group of mutually significantly intercorrelated features, but in practice α should simply be set to take advantage of the available processing resources.

In the second stage, the complete CFS-function is used to remove redundant and irrelevant features until n features remain.

The overall procedure can be summarized as follows:

1. While the number of features in S is greater than αn, remove feature F'_{max} from S
$$F'_{max} \equiv \operatorname*{argmax}_{F_i} \mathrm{CFS}'(J[S \setminus F_i]).$$
2. While the number of features in S is greater than n, remove feature F_{max} from S
$$F_{max} \equiv \operatorname*{argmax}_{F_i} \mathrm{CFS}(J[S \setminus F_i]).$$
3. Extract the n remaining features in S.

5 Empirical Results

In this section we introduce an image stream indexing problem which we use to evaluate the hypothesis driven parallel feature extraction technique proposed in the previous section. The image stream indexing problem consists of a stream of images where each image is parted into 64 regions by an 8x8 grid. Here, three different types of features can be extracted from each region, that is, there are in total 192 features available for extraction in an image. However, due to processing resource limitations, only 10 of the 192 available features can be extracted from any given image in the stream (the features are assumed to be extracted in parallel on 10 tightly coupled processing elements). The goal is to detect and track the position of an object when it appears in the image stream.

We have designed two different simulations, one where the three features within an image region are pairwise highly correlated and another where the three features within an image region are causally complexly interacting, giving rise to higher-order correlations. The structure of the respective Bayesian networks are shown in Figure 3 and Figure 4. These Bayesian networks are comparable in complexity to Bayesian networks used in recent advanced content-based media indexing systems such as [1] and [10].

In order to track objects in the image stream rather than just locating an object in a single image, the object grid position probability distribution must be carried on from image to image, possibly transformed according to an object movement model. This is indicated by the dotted variable and link in Figure 3 and Figure 4. Here we let the probability of the object moving to a vertically or horizontally adjacent grid position (between two images) be 0.1.

The Bayesian network used in the first simulation consists of a hypothesis Pos (object grid position) as well as 192 object tests Fe_{ij} parted into 64 groups of 3 object tests (one group per grid position ij). The object tests are conditionally independent given the hypothesis, and each object test has a probability of 0.1 of false detections and a probability of 0.1 of missing detections (i.e. the error rate is 0.1) in the respective image region. As a result, the object tests within an image region are pairwise highly correlated. The DAG of the resulting Bayesian network is shown in Figure 3.

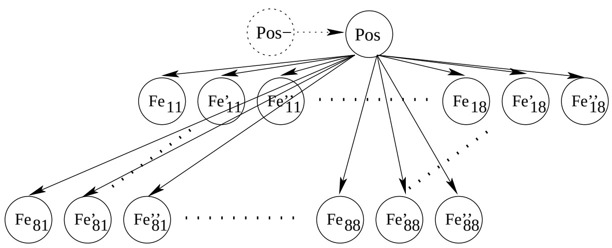

Figure 3. *A Bayesian network modeling the correspondence between the position of an object in an image and the results of object tests. The object tests at each grid position ij are pairwise highly correlated.*

Before the object is located in the first simulation, the hypothesis driven parallel feature extraction technique systematically searches for an object, extracting 10 features in parallel from each image. As pairwise feature-feature correlations are taken into account, features from the same grid position are not extracted in parallel. In contrast, the naive approach of selecting the 10 best features only with respect to their individual hypothesis prediction capability (see Section 3), results in up to three features being extracted at each grid position. Consequently, less grid positions are covered and the object detection capability is reduced. Similarly, when the object has been located the hypothesis driven parallel feature extraction technique systematically extracts features at probable object grid positions (reflecting the object movement model). The naive approach covers less grid positions (it may extract up to three features at each grid position) and consequently the probability of losing the object increases.

The Bayesian network used in the second simulation consists of 192 features and 64 environment conditions parted into 64 groups of 2 object tests Fe'_{ij}, Fe''_{ij}, 1 environment condition test Fe_{ij}, and 1 environment condition Co_{ij}. If the environment condition at a grid position is $positive$ the corresponding object tests behave as the object tests in the first Bayesian network. On the other hand, if the environment condition at a grid position is $negative$, the results of the corresponding object tests are inverted. An environment condition test determines the environment condition in the respective image region with error rate 0.01. This is an extreme example of higher-order correlations where neither the object tests nor the condition tests are individually correlated with the hypothesis. On the other hand, the condition test and the object tests in a region are highly correlated with the hypothesis when seen in combination. The DAG of the Bayesian network is shown in Figure 4.

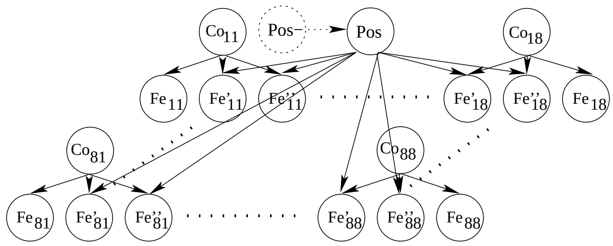

Figure 4. *A Bayesian network modeling the correspondence between the position of an object in an image and the results of object tests. The two object tests and the condition test at each grid position ij are causally complexly interacting with respect to the hypothesis, giving rise to higher-order correlations.*

In the second simulation, when following the approach from Section 4 of combining causally complexly interacting features into joint features, the hypothesis driven parallel feature extraction technique executes one object test and one environment condition test in 5 image regions. Consequently, the tracking behaves as in the first simulation. On the other hand, if the features are not combined into joint features, features are extracted blindly and the tracker is unable to detect or track the object.

We have integrated the proposed technique into a tracker based on the particle filter [8] to test its execution performance. The tracker uses the hypothesis driven parallel feature extraction technique to extract 15 features from each image in the image stream (here the α constant is set to 2), before it updates the probability distribution of the object grid position. 128 samples are used to approximate the joint Bayesian network variable distribution. The object grid position probability distribution and the necessary Symmetrical Uncertainty Coefficients are calculated based on this approximation. We used the Bayesian network from Figure 3 to model the interaction between object grid position and feature results. When run on a 933MHz Pentium III the tracker was able to select efficient feature sets to be extracted in each image at the rate of 55 images per second (and achieved an error rate of 0.051). Consequently, the processing cost of the hypothesis driven parallel feature extraction technique seems to be insignificant when compared to the potential reduction in feature extraction resource usage when features are extracted selectively.

6 Conclusion and Further Work

We have proposed a novel technique for hypothesis driven parallel feature extraction based on Bayesian networks. The empirical results indicate that the approach effectively deals with features that are pairwise highly correlated and that are causally interacting in complex ways. As a consequence, the technique opens up for combining the processing benefits of parallel feature extraction and hypothesis driven feature extraction.

In our further work we will combine the proposed technique with the ARCAMIDE algorithm from [14] to support real-time resource-aware adaptation of content-based media indexing systems to the processing architecture and the processing resources at hand. In this context we will examine characteristics of a content-based media indexing system such as: processing cost, processing time (latency), classification error rate, as well as the interaction between these characteristics. Thereby, we will work on establishing the notion of Quality of Service in real-time content-based media indexing systems.

References

[1] S. Hongeng, F. Bremond, and R. Nevatia, "Bayesian Framework for Video Surveillance Applications," in *15th International Conference on Pattern Recognition*. 2000, vol. 1, pp. 164–170, IEEE.

[2] S. Eickeler and S. Muller, "Content-based Video Indexing of TV Broadcast News using Hidden Markov Models," in *Conference on Acoustics, Speech and Signal Processing*. 1999, vol. 6, pp. 2997–3000, IEEE.

[3] H. Ganster A. Pinz, M. Prantl and H. Borotschnig, "Active fusion—a new method applied to remote sensing image interpretation," *Special Issue on Soft Computing in Remote Sensing Data Analysis*, vol. 17, no. 13, pp. 1340–1359, 1996.

[4] Y. Nakamura and M. Nagao, "Parallel Feature Extraction System with Multi Agents -PAFE-," in *11th IAPR International Conference on Pattern Recognition*. 1992, vol. 2, pp. 371–375, IEEE.

[5] V. S. W. Eide, F. Eliassen, and O. Lysne, "Supporting Distributed Processing of Time-based Media Streams," in *Distributed Objects and Applications, DOA2001*, 2001, pp. 281–288.

[6] V. Manian and R. Vasquez, "A Computational Framework for Analyzing Textured Image Classification," in *IEEE Conference on Intelligent Systems for the 21st Century*. 1995, vol. 1, pp. 717–723, IEEE.

[7] H. Buxton and S. Gong, "Visual Surveillance in a Dynamic and Uncertain World," *Artificial Intelligence*, vol. 78, no. 1-2, pp. 431–459, 1995.

[8] O.-C. Granmo, F. Eliassen, and O. Lysne, "Dynamic Object-oriented Bayesian Networks for Flexible Resource-aware Content-based Indexing of Media Streams," *Scandinavian Conference on Image Analysis, SCIA'2001*, 2001.

[9] F. V. Jensen, *Bayesian Networks and Decision Graphs*, Series for Statistics for Engineering and Information Science. Springer Verlag, 2001.

[10] A. Singhal, L. Jiebo, and C. Brown, "A Multilevel Bayesian Network Approach to Image Sensor Fusion ," in *Proceedings of the Third International Conference on Information Fusion*. 2000, vol. 2, pp. 9–16, IEEE.

[11] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic Networks and Expert Systems*, Series for Statistics for Engineering and Information Science. Springer Verlag, 1999.

[12] F. V. Jensen, H. I. Christensen, and J. Nielsen, "Bayesian Methods for Interpretation and Control in Multi-agent Vision Systems," in *Applications of Artificial Intelligence X: Machine Vision and Robotics, SPIE*, 1992.

[13] M. A. Hall, *Correlation-based Feature Selection for Machine Learning*, Ph.D. thesis, Department of Computer Science, Waikato University, New Zealand, 1999.

[14] O.-C. Granmo, "Automatic Resource-aware Construction of Media Indexing Applications for Distributed Processing Environments," in *Proceedings of the 2nd International Workshop on Pattern Recognition in Information Systems (PRIS2002)*. April 2002, pp. 124–139, ICEIS Press.