Rushes Summarization with Self-Organizing Maps

Markus Koskela
markus.koskela@tkk.fi

Mats Sjöberg
mats.sjoberg@tkk.fi

Jorma Laaksonen
jorma.laaksonen@tkk.fi

Ville Viitaniemi
ville.viitaniemi@tkk.fi

Hannes Muurinen
hannes.muurinen@tkk.fi

Adaptive Informatics Research Centre
Helsinki University of Technology
P.O.Box 5400, FI-02015 TKK, Finland

ABSTRACT

In this paper, we describe our approach for video summarization that was applied to the BBC rushes material as part of the TRECVID 2007 evaluations. The method consists of initial shot boundary detection followed by shot similarity assessment and pruning, with both stages implemented using multiple parallel Self-Organizing Maps and within our content-based multimedia information retrieval and analysis framework named PicSOM. The results indicate that our approach can be successfully applied to rushes summarization. Compared to other submissions, our method resulted in the overall shortest summaries with close to median performance in the fraction of ground-truth inclusions found.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Evaluation/methodology*

General Terms

Experimentation, Algorithms

Keywords

video summarization, self-organizing map

1. INTRODUCTION

Video summarization is a process where a long video is converted to a considerably shorter form. The produced summary can then be used e.g. to facilitate efficient searching and browsing of video files in large collections. The aim of automatic summarization is to preserve as much as possible from the essential content of the original video without any manual input required. What constitutes essential content in videos is naturally subjective and dependent on the intended use of the videos and their overall content.

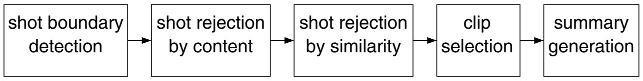

Figure 1: An overview of the used summarization algorithm.

In this paper, we provide a description of our approach to the BBC rushes summarization task [8] of TRECVID 2007 [10]. We use a multiple-stage method shown in Fig. 1 where the first and third stages are implemented using multiple parallel Self-Organizing Maps (SOMs) [4]. The SOM is an artificial neural network capable of dimensionality reduction by a topology-preserving mapping.

First, we apply our shot boundary detection algorithm to the rushes videos. For each video, this provides us with lists of shots, which are used in the following stages as basic units of processing. We detect and remove unwanted shots (color bar test screens, all black or white frames) from the videos, and apply face detection and motion activity estimation. Next, we compute the visual similarities between all pairs of shots and remove overly similar shots. Each remaining shot is then represented in the summary with a one-second clip, with the audio track not included. The selected clips are combined using temporal ordering and fade-outs and fade-ins from black.

The rest of the paper is organized as follows. Section 2 provides an overview of the SOM-based shot boundary detection algorithm. Section 3 describes the methods used for content-based shot pruning and inter-shot similarity evaluation. Section 4 provides a description of our strategy for selecting representative one-second clips from the remaining shots. Section 5 discusses the obtained results compared to other submissions, and Section 6 concludes the paper with lessons learned and some final remarks.

2. SHOT BOUNDARY DETECTION

We used a video shot boundary detection algorithm [7, 9] that spots discontinuities in the visual stream by monitoring video frame trajectories on SOMs. The SOM can be used to visualize high-dimensional data on a two-dimensional lattice. Plotting a *trajectory* is a highly useful visualization technique that can be used when handling data vectors that vary as a function of some scalar variable, usually time. A

TVS'07, September 28, 2007, Augsburg, Bavaria, Germany.

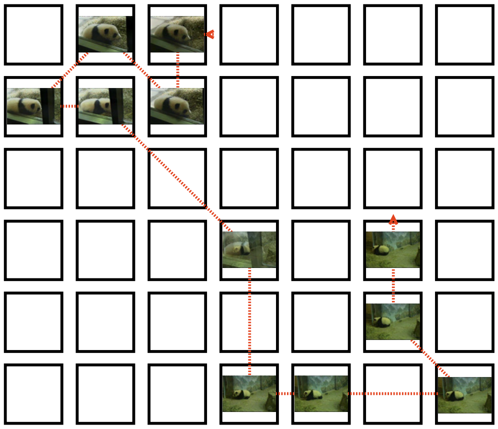

Figure 2: An example of a video trajectory on a SOM. Consecutive frame BMUs are connected to form the trajectory.

temporal SOM trajectory is the "path" that the data mapped on a SOM traverses as time advances. At each time step a feature vector is first calculated from the data, and then the vector is mapped to the best-matching unit (BMU) on the map. When visualizing the trajectory, the consecutive BMUs can be connected with a line to form the full path. An example of such a trajectory of consecutive video frames is shown in Fig. 2.

SOM trajectories can be used to monitor changes in the input data over time. They have been applied to various tasks including speech recognition [3] and chemical process monitoring [11]. Similar vectors are mapped spatially close to each other on the SOM, and thus the trajectory should hover over the same map region if the input vectors do not change significantly. On the other hand, when there is a sudden large change between the values of consecutive input vectors, there should be a relatively large leap in the trajectory on the map from some region to another. If the input vector values drift slowly over time in the input vector space from some region to another, there should also occur an analogous drift in the SOM trajectory.

We use the evolution of the frame trajectory to monitor the rate of change in consecutive frames by observing the distances between the consecutive trajectory points. This approach is taken in the shot boundary detection method described in [7], and also used in these experiments. The SOM mapping compensates for the probability density differences in the feature space, and consequently distances between SOM coordinates are more informative than distances between plain feature vectors. The method compares two sliding best-matching unit windows instead of just measuring distances between two trajectory points, which increases the robustness of the detector. The robustness is further enhanced by using a committee machine of multiple SOM-based detectors.

In these experiments, we used large (256×256 map units) SOMs trained with three different visual features extracted from all the frames of the development set. The used features were: *color layout* (DCT coefficients of average color in 20×20 grid), *edge co-occurrence* and *edge histogram* (co-occurrence matrix and histogram, respectively, of four Sobel edge directions). The frames of the test videos were then directly mapped to their feature-wise BMUs on these parallel SOMs. The shot boundary detection parameters were learned with the development set using a manually-generated ground truth for one of the development videos. Due to the diverse material found in the development videos, these parameters resulted in a large number of shots for some of the videos and only a few shots for some others. The average length of shots varied from 4.7 seconds to 430 seconds within the videos in the development set. A large number of resulting shots was deemed relatively harmless since the overly similar shots will be pruned at a later stage (Section 3.2). On the other hand, detecting too few shots constituted a problem as our current method supports the inclusion of only one clip from each shot. Therefore, to increase the number of shots detected we conditionally reduced the value of the vote result threshold T_v (see [7]) from its initial value of $T_v = 1.0$. We used $T_v = 0.8$ if the average length of shots was initially more than 30 seconds and $T_v = 0.6$ if it was more than 100 seconds.

3. SHOT EVALUATION

After the shot boundaries have been detected, we proceed to the rejection of undesired or uninformative shots. Then, for the remaining shots, we estimate the novelty values of the shots for the purpose of selecting as different shots as possible to the summary.

3.1 Detection of Special Contents

A video summary should be concise, but still contain the most noteworthy or special contents of the original video. Similarly, any useless or trivial parts of the material should not appear in the summaries. Both the noteworthy and useless types of content are naturally domain-specific. In this work, we detect certain types of content by using specialized detectors. In order to follow the standard naming convention of semantic multimedia analysis, we denote these special types of content as *concepts*, although in this work the detected concepts are of rather low level.

We use simple feature-space detectors for the following three concepts: *color bar test screens* (cb), *black frames* (bf) and *white frames* (wf). Due to their simplicity, these concepts were detected using Euclidean distance thresholds in the *color layout* feature space.

For *face detection* (fd), we used the detector included in Intel's OpenCV Library[1]. The detector is based on Haar-like features and a cascade of boosted tree classifiers. Furthermore, the face candidates returned by the OpenCV detector were pruned by using a simple skin color detector in the YCbCr color space [1].

For *motion activity* (ma) information, we extracted the thresholded intensity from the MPEG-7 Motion Activity descriptor [2] for one second intervals from the entire videos.

After running the concept detectors, we analyzed the content of each shot based on the concepts detected within the frames of the shot. A shot i was rejected from further processing if

$$-\sum_{k\in\{\mathrm{cb,bf,wf}\}} w_k \log(c_k(i)+1) + \sum_{k\in\{\mathrm{fd,ma}\}} w_k \log(c_k(i)+1) < 0 \quad (1)$$

where $w_k > 0$ are heuristically set weights for the concepts and $c_k(i)$ the number of frames in shot i marked positive for the concept k.

[1] http://www.intel.com/technology/computing/opencv/

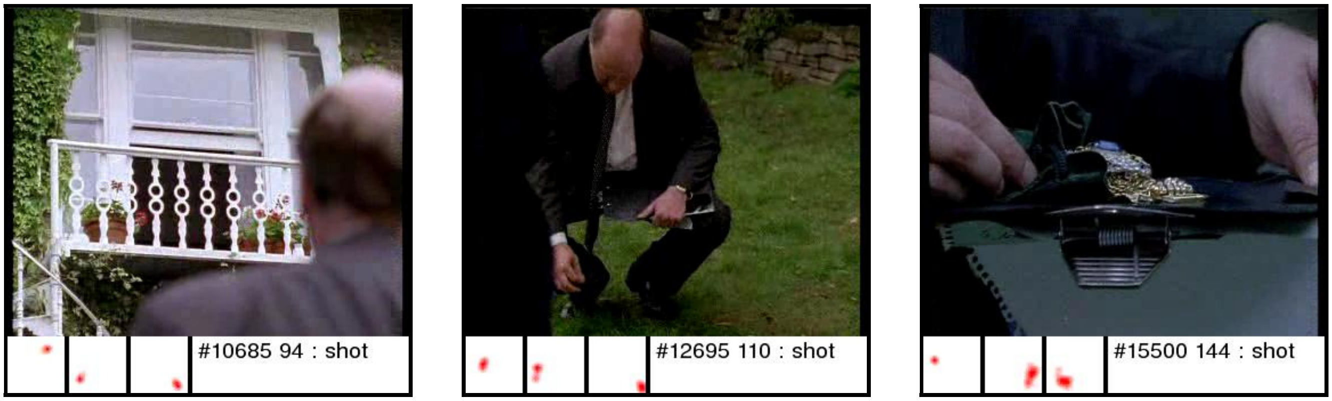

Figure 3: Representative frames and SOM signatures of three video shots.

3.2 Shot Similarity Pruning

We determine the novelty of a shot based on the shots' visual contents using a SOM-based method described in this section. For simplicity, we assume here that we are using a single SOM, although in practice we used three parallel SOMs trained with different features to increase robustness. The overall result is then obtained as the average of the individual SOMs. We trace the trajectory of the frames within the shot in question and record the corresponding BMUs. The set of BMUs constitutes a SOM-based signature for the shot, which can then be compared to other shots' signatures to determine whether a shot is visually unique or similar to some other shots.

There are two distinct schemes for constructing these shot-wise signatures. First, we could analyze the trajectory of the consecutive BMUs during the shot and obtain a temporal signature representing the dynamic structure of the shot. Comparing two shots then involves comparing their respective BMU trajectories on the surface of the SOM in question. This approach might be needed, for example, to distinguish a scene where a man is walking into a room from another scene where he is walking out of the room, i.e. it takes into account the temporal or causal aspect of the video. The usefulness of this dynamic structure is, however, questionable and domain-dependent. It may also lead to large differences between shots that are overall similar, but have different temporal signatures for some reason, e.g. due to failed shot boundary detection.

For these experiments, we decided to ignore the temporal element and treat all the frames of the shot equally as separate unordered data items. The set of frame-wise BMUs representing a shot can be treated as any data subset or class, whose distribution on a SOM surface can be obtained by counting the number of BMU hits for each map unit. Normalized to unit sum, these hit frequencies give a discrete histogram representing the visual contents of the shot in question. Due to the topology preservation property of the SOM, one may now smooth the representation by forcing the neighboring SOM units to interact with each other. This is realized by low-pass filtering or convolving the hit distributions on the SOM surface. Then, by enumerating the units of the two-dimensional SOM grid, we can represent the distribution of the mth shot as a vector $P_m \in \mathbb{R}^k$, where k is the total number of SOM units, and use a suitable distance metric $d(P_m, P_n)$ (e.g. Euclidean distance) to measure the dissimilarity of two shots m and n. This process closely resembles the way how SOMs are being used for image and video retrieval and concept detection in our PicSOM system [5, 6].

For this stage, we trained three separate 64 × 64-sized SOMs ($k = 4096$) for each video to be summarized. The three video-wise SOMs were always trained using the same visual features as in shot boundary detection, i.e. color layout, edge co-occurrence and edge histogram. Fig. 3 shows example frames from three shots and the convolved SOM-based trajectory signatures of those shots as red-colored responses on the SOM surfaces. The three signatures correspond to the color layout, edge co-occurrence and edge histogram features, respectively.

The pruning of the most similar shots was done as follows. First, we calculate the pairwise Euclidean distances $d(P_m, P_n)$ of all shot pairs. We then start deleting shots beginning from the most similar pair until the total number of remaining shots is below the allowed limit and an experimentally set minimum similarity threshold is exceeded. This additional threshold was included based on a subjective analysis of the repetition remaining in the summaries created for the development videos.

4. CLIP SELECTION

The next stage after determining the shots to be included in the summary is the selection of the most informative clips or subshots from within the selected shots. From each remaining shot, we selected a representative clip of fixed length of one second to the final summary. In this stage, we again utilize the concept detectors described in Section 3.1. Initially, we favor frames near the center of the shot using a linear weighting, and award increased scores for frames containing faces and frames with increased motion activity. Correspondingly, scores for any frames detected as color-bars, black frames, or white frames are reduced. The relative weights for these concepts were again set heuristically.

It was observed that the emphasis can vary greatly between the shot-level evaluation and intra-shot analysis, depending on the input video. With a video containing only a small number of long shots, the subshot selection becomes the dominant step, whereas the shot-level selection is crucial for the summarization of a video with numerous short shots. An effective summarization algorithm should be able to handle both these cases equally well.

Table 1: An overview of selected summarization results. The values are averages over all 42 test videos.

	Ours	Max	Median	Min
Duration (DU)	26.1	64.2	53.6	26.1
Total time (TT)	61.7	119.3	94.2	61.7
Inclusion (IN)	0.45	0.68	0.49	0.25
Understandability (EA)	3.23	3.60	3.33	1.97
Duplicate video (RE)	3.87	3.98	3.66	3.02

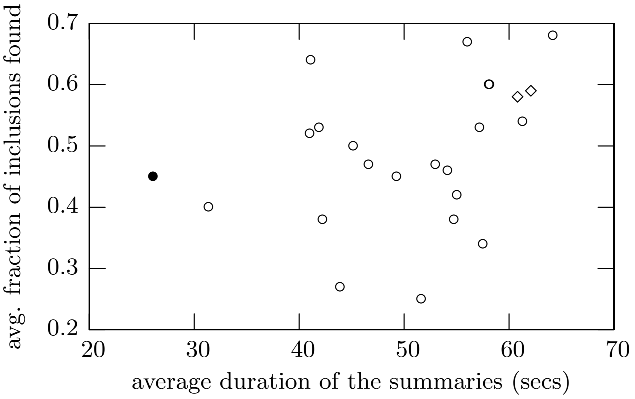

Figure 4: The average fraction of inclusions found vs. average duration of all submitted summaries; our run shown as "•", baseline runs as "◇".

5. RESULTS

An overview of our summarization results is given in Table 1. The shown measures are from the standard measures provided by NIST and described in [8]. In brief, the two topmost results (DU and TT) are measures of time (in seconds), IN lists the fraction of ground-truth inclusions found in the summaries, and the two remaining results are from an assessor questionnaire with the range of 1–5 (5 is the desired value in both cases). The most striking characteristics of our summaries when compared to the other submissions are both the duration of and total time spent assessing the summaries—in both these senses our submitted summaries were the shortest. This is also reflected in the assessments of the amount of duplicate video present as our mean result is clearly above the median. The average duration (in seconds) also equals the average number of distinct clips in our summaries since we used fixed one-second clips.

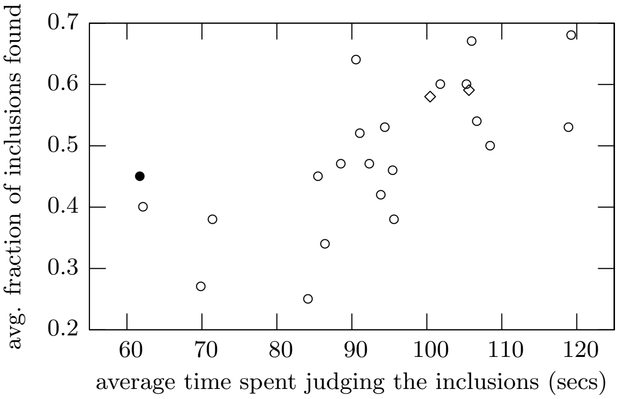

Figure 5: The average fraction of inclusions found vs. average time spent judging the inclusions; our run shown as "•", baseline runs as "◇".

On the other hand, the fraction of ground-truth inclusions found in our summaries (45% on average) was slightly below the median of all scores. This is undoubtedly directly affected by the comparatively short durations of our summaries. To analyze this relation further, Figs. 4 and 5 show plots of the average fractions of inclusions found over the average values of durations and total times spend judging the inclusions, respectively, for all submissions. Our submission is highlighted as a filled bullet and the two baseline runs provided by CMU are shown with diamond shapes. The figures show a clear relation between these measures as the summaries with high fractions of inclusions found tend to have high time expenditure values, as was to be expected. In fact, the summary with the longest duration and most time spent has the highest fraction of inclusions found. Furthermore, the baseline runs perform surprisingly well, suggesting that a reasonable performance level can be reached with relatively straightforward methods given the 4% limit for the analyzed material. Finally, Figs. 6 and 7 show the fractions of ground-truth inclusions found in our summaries and relative durations of our summaries for each test video, compared to the corresponding maximum, median, and minimum values.

5.1 Computational Requirements

The computational requirements of our summary generation method were high, 377 minutes per summary on average, using a cluster of Linux servers with AMD Opteron SE 2220 2.8 GHz processors and 16 GB of memory. The reported execution times are for a single processor, though.

The clear majority of the computational effort was required by feature extraction. The used shot boundary detection algorithm requires the extraction of all used features from each frame of the video. The used features were our standard ones previously used in various multimedia analysis tasks and not optimized for this purpose.

Apart from feature extraction, the other computationally demanding task was the training of the video-wise SOMs for the shot similarity pruning stage, taking about 12 minutes per summary on average. The remaining processing steps were computationally considerably lighter, taking on average approximately 10 minutes per summary in total.

6. CONCLUSIONS

The video summarization task differs from the other tasks in the TRECVID evaluations due to the inherent subjectivity in defining a good summary and the difficulty of generating ground-truth data. Such ground truth would, however, be extremely useful as otherwise it is difficult to systematically optimize and analyze the effects of the various parameters involved. A pivotal parameter in summarization is the duration of the summary. However, as each team submitted only a single summary per test video, it is hard to estimate the actual relation between increasing the summary durations and the amount of important content captured.

In particular, we observed that our summaries of short videos were often too short and lacked important content. The shot similarity threshold should thus depend on the length of the original video. On the other hand, for some

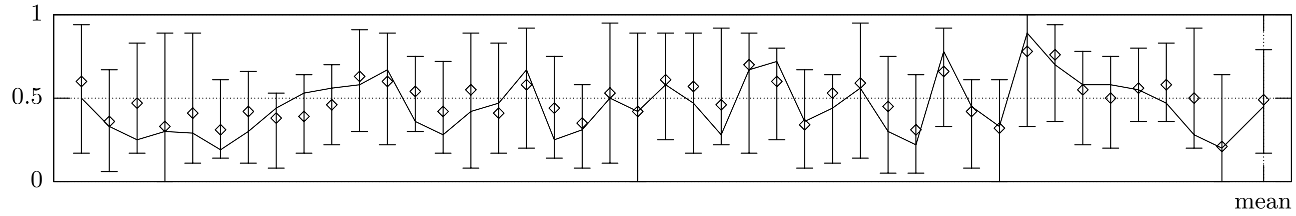

Figure 6: The fractions of inclusions found in our summaries for each test video (continuous plot). The error bars show the corresponding maximum, median (◇), and minimum values over all the submitted summaries.

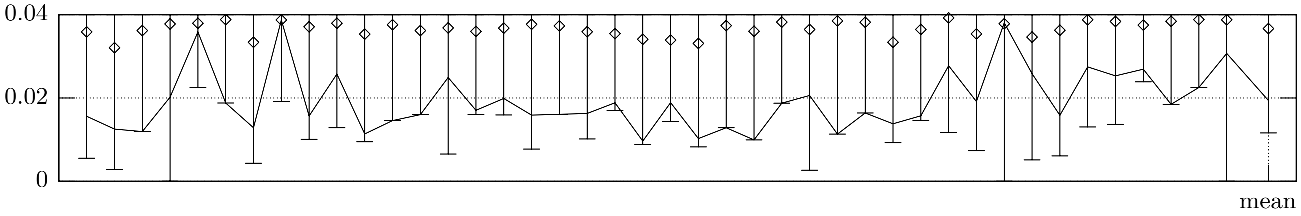

Figure 7: The relative durations of our summaries (continuous plot). The error bars show the maximum allowed duration (always 4%), and median (◇) and minimum values over all the submitted summaries.

videos our shot boundary detection algorithm found only a small number of shots. This could be accommodated e.g. by defining a maximum shot length or using more adaptive shot boundary detection parameters. Other conceivable improvements to our summarization algorithm include supporting longer clips and multiple clips from long shots when needed.

The concepts used in these experiments were preliminary and simple. The use of existing high-level concept detectors for summarization should be studied further. With rushes videos, the clapper boards in particular constitute a problem and should preferably be detected and removed.

In further development of the summarization method, we will also consider the use of temporal shot signatures instead of the static ones used here, the inclusion of audio into the summaries, and more sophisticated ways of creating the final summaries than the concatenation of the selected clips using fade-outs and fade-ins from black.

In the current system, we have not yet considered issues relating to the required computational effort. The current shot boundary detection algorithm requires features extracted from each frame, inevitably resulting in high computational requirements. The used low-level features were also not selected based on computational complexity.

7. ACKNOWLEDGMENTS

Supported by the Academy of Finland in the projects *Neural methods in information retrieval based on automatic content analysis and relevance feedback* and *Finnish Centre of Excellence in Adaptive Informatics Research.*

8. REFERENCES

[1] D. Chai and K. N. Ngan. Face segmentation using skin-color map in videophone applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(4):551–564, June 1999.

[2] ISO/IEC JTC 1. Multimedia content description interface - Part 3: Visual, 2002. 15938-3:2002(E).

[3] T. Kohonen. The 'neural' phonetic typewriter. *Computer*, 21(3):11–22, March 1988.

[4] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, third edition, 2001.

[5] J. Laaksonen, M. Koskela, and E. Oja. PicSOM—Self-organizing image retrieval with MPEG-7 content descriptions. *IEEE Transactions on Neural Networks*, 13(4):841–853, July 2002.

[6] J. Laaksonen, M. Koskela, and E. Oja. Class distributions on SOM surfaces for feature extraction and object retrieval. *Neural Networks*, 17(8-9):1121–1133, October-November 2004.

[7] H. Muurinen and J. Laaksonen. Video segmentation and shot boundary detection using self-organizing maps. In *Proceedings of 15th Scandinavian Conference on Image Analysis (SCIA 2007)*, pages 770–779, Aalborg, Denmark, June 2007.

[8] P. Over, A. F. Smeaton, and P. Kelly. The TRECVID 2007 BBC rushes summarization evaluation pilot. In *Proceedings of the TRECVID Workshop on Video Summarization (TVS'07)*, pages 1–15, New York, NY, September 2007. ACM Press.

[9] M. Sjöberg, H. Muurinen, J. Laaksonen, and M. Koskela. PicSOM experiments in TRECVID 2006. In *Proceedings of the TRECVID 2006 Workshop*, Gaithersburg, MD, USA, November 2006.

[10] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[11] V. Tryba and K. Goser. Self-organizing feature maps for process control in chemistry. In *Proceedings of International Conference on Artificial Neural Networks (ICANN-91)*, volume 1, pages 847–852, Espoo, Finland, June 1991.