Improving Digital Video Commercial Detectors with Genetic Algorithms

J. David Schaffer **Lalitha Agnihotri** **Nevanka Dimitrova** **Thomas McGee** **Sylvie Jeannin**

Philips Research
345 Scarborough Road
Briarcliff Manor, NY 10510, USA
{dave.schaffer, lalitha.agnihotri, nevenka.dimitrova, thomas.mcgee}@philips.com

Abstract

The advent of digital video offers many opportunities to add features that enhance the viewing experience. One much-discussed feature is the possibility that commercials might be automatically detected in the video stream. We report on initial experiments with a class of commercial detection algorithms and show how their performance can be enhanced by applying genetic search to the optimization of some of their internal parameters. We show how a scalar genetic algorithm can locate sets of parameters in a multi-objective space (precision and recall) that outperform the values selected by an expert engineer. While a useful observation in itself, we also argue that this approach may be a necessity as the features that distinguish commercials from other video content will certainly vary with video format, the country of broadcast and possibly over time. We present the results of optimizing a commercial detection algorithm for different data sets and parameter sets. We are convinced that GAs drastically improved our approach and enabled fast prototyping and performance tuning of commercial detection algorithms.

1 INTRODUCTION

Digital consumer storage functionality will appear on many consumer devices such as video recorders advanced set-top boxes and personal mobile storage servers. Content-based video analysis can be applied to introduce more advanced retrieval, scanning and playback features. One of the most important features consumers want is commercial detection, indication, and skipping. In addition, commercial detection plays a major role in automatic analysis and structure detection from multimedia signals for generating summaries and table of contents for a program. A major challenge in bringing these features into consumer devices is to overcome the low processing power inherent in these low-cost consumer devices. It is therefore, important to take full advantage of the MPEG[1] hardware compression and perform the analysis on the features already available during the encoding of the input video stream thereby saving precious computational cycles.

The implementation of a commercial detection algorithm using features derived from MPEG parameters has been described by Dimitrova et al. [3]. This implementation assumes that the target platform includes an encoder and the features extracted from the encoder are processed on a low-end host processor. Consequently, the chosen algorithms are based on simple voting and thresholding techniques. An important challenge is to provide high accuracy commercial detectors for given test material. The process of benchmarking and fine tuning is tedious and requires many experiments with various thresholds. It normally takes weeks to fine tune an algorithm. Also, experiments are needed to see the impact of threshold ranges on the algorithms for different types of TV programs. It is extremely important to provide methodology for fast tuning of the algorithm parameters and providing tools for analysis of the algorithms for given test genres. Here, we report on experiments that used a genetic algorithm (GA) to locate improved sets of thresholds on a chosen commercial detection algorithm [3]. In addition, GAs provide a framework for fast tuning and analysis of parameters for commercial algorithms.

2 THE CHALLENGE OF COMMERCIAL DETECTION

What we encounter is a fairly traditional pattern discrimination problem, yet there are reasons to believe that the achievement of successful performance will require the use of powerful optimization methods, and not just once. The patterns of features and their combinations

[1] MPEG stands for Moving Picture Expert Group. This body establishes standards that are used in the compression, transmission, and decompression of digital video.

that distinguish commercials from other video content are known to change with video format (e.g. NTSC vs PAL[2]), and with culture (cinematic styles differ from culture to culture). Furthermore, these patterns are not well understood, but our initial investigations suggest that they are highly non-linear. In addition, they can be expected to change over time as styles of programming as well as styles of advertising change. The broadcaster and/or advertiser may also change the advertisement characteristics to avoid detection. Therefore, we envision a need to more or less continuously resample video content and re-adjust the detector's parameters. This calls for a robust automatic optimization method such as a GA.

3 RELATED WORK

In the literature there are many methods that have been proposed for detecting commercials extending back more than 20 years [1, 2, 5, 6, 7, 8, 9, 10, 11]. One common method is detection of high activity rate and black frame detection coupled with silence detection before a commercial break. These methods show partially promising results [8]. The use of monochrome images, scene breaks, and action (the number of edge pixels changing between consecutive frames and motion vector length) as indicative features have also been reported [8]. Blum et al. used black frame and "activity" detectors [1]. Activity is the rate of change in luminance level between two different sets of frames. Commercials are generally rich in activity. When a low activity is detected, the commercial is deemed to have ended. Unfortunately, it is difficult to determine what is "activity' and what is the duration of the activity. In addition, black frames are also found in dissolves. Any sequence of black frames followed by a high action sequence can be misjudged and skipped as a commercial. Another technique by Iggulden is using the distance between black frame sequences to determine the presence of a commercial [6]. Lewine et al. determined commercials based on matching images. Similarly, Forbes et al. use a video signal identifier to memorize repetitive television signals in order to automatically control recording of TV programs [5]. However, the commercial has to be identified to the system before it can recognize it. Nafeh proposed a method for classifying patterns of television programs and commercials based on learning and discerning of broadcast audio and video signals using a neural network [10]. However, none of the reported methods used an automatic optimization method for tuning of algorithm parameters and analysis of the parameters behavior. In this sense, we are proposing a fast method for algorithm benchmarking and fine tuning using GAs.

[2] NTSC is the National Television Standards Committee and PAL is Phase Alternating Line. NTSC designates the video standard used in North America and some other countries and PAL is standard for most of Europe.

4 MPEG-RELATED FEATURES

There are different sets of features that are extractable during the MPEG-encoding process. The encoder internal parameters, called low-level features, extracted during the encoding process are:

- frame type indicator, which discriminates between intra-coded (I), predicted (P) and bi-directional (B) frames;
- luminance DC value at macroblock level on I-frames only; A macroblock is a coding layer used in MPEG.
- VTS (video time stamp) of the observed frame, which assures correct synchronization of extracted video features and matching video frames;
- *macroblock correlation factor*, which represents the correlation between the current macroblock and reference macroblock in the reference frame.

A schema of a representative MPEG-2 encoder is shown in Figure 1. The luminance DC values are available in the information chain at point *a* after the *Discrete Cosine Transformation* and the *Quantizer* before the *Variable Length Coding* . These DC values can be used for content analysis algorithms. The *Motion Estimation* encoding block at point *b* generates a spatio-temporal representation of the macroblock motion for an efficient encoding process. This macroblock motion recovery process uses the correlation of an actual macroblock and possible reference macroblocks in the reference frame. The correlation factor of the best match of actual macroblock and reference counter macroblock is tapped for further content analysis.

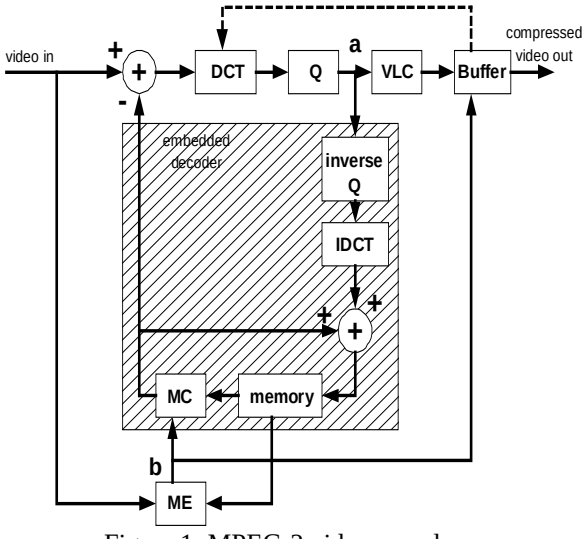

Figure 1: MPEG-2 video encoder.
DCT = Discrete Cosine Transform,
Q = Quantization, VLC = Variable Length Coding,
MC = Motion Compensation, ME = Motion Estimation,
IDCT = Inverse DCT.

The low-level parameters are used to derive more meaningful features, called mid-level features, such as black/unicolor frame, scene change, and letterbox. A

simple threshold technique is sufficient to implement a reliable black frame detection algorithm by using the sum of the luminance DC values of the entire frame (*Luminance DC Summation*) as input value to the algorithm. Luminance DC values can also be used to discriminate between a 4:3 and 16:9 aspect ratio video frame (Letterbox) by similarly evaluating the appropriate the luminance values of upper and lower macroblock slices of the video frame. Unicolor frames are detected by applying a threshold on the average of the absolute differences of luminance DC values between adjacent blocks, on the whole frame. The absolute differences of luminance DC values between adjacent blocks can also be used for black frame and letterbox detection. The use of the variation of the *Luminance DC Summation* over consecutive frames and the *macroblock correlation value* facilitate the implementation of a scene change detection algorithm. The mid-level features such as black frame, letterbox, and scene changes are used for commercial detection.

5 THE COMMERCIAL DETECTOR ALGORITHM

There are different families of algorithms that can be used for commercial detection based on low and mid-level features. In this implementation we have experimented with the available mid-level features:

1. black frame
2. unicolor frame
3. keyframe distance (i.e. consecutive scene changes distance, also denoted as KF distance in the following)
4. letterbox (i.e. 4:3 versus 16:9 aspect ratio discrimination)

All the above values are computed for each I frame. In the first step, the algorithm checks for "triggers" that could flag the possible start of a commercial break. The algorithm, then verifies if the detected segment is a commercial break.

5.1 TRIGGERS

In the current experiments we have used the time interval between detected black or uni-color frames as triggers. Normally, black frames (or unicolor frames) are used by the content creators to delineate commercials within a commercial break, as well as at the beginning and ending of a whole commercial break. We assume that a commercial break starts with a series of black (unicolor) frames and that during the commercial break we will encounter black (unicolor) frames within a predetermined threshold (e.g. 50 seconds). Also, we have placed constraints on the duration of the commercials. We have determined by looking at a number of commercials that commercial breaks can not be shorter than one minute and can not be longer than six minutes. An additional constraint that is derived from the material we have seen is that commercial breaks have to be at least one and a half minute apart. This last constraint is important for the linking of the segments that potentially represent commercials. If the linking is allowed for a long period of time, we might end up with very long commercial breaks, which in fact might contain a commercial break and an action scene from a movie. After some number of black sequences the probability of commercial being present increases and potential commercial end is searched for.

5.2 VERIFIERS

Once a potential commercial is detected, other features are tested to increase or decrease the probability of a commercial break. Presence of a letterbox change or high cut rate expressed in terms of low keyframe distance can be used as verifiers. In the case of letterbox change, the probability that the given area is a commercial break is increased. In the case of low keyframe distance (or high cut rate), the probability of a commercial being present is increased. If the cut rate is below a certain threshold then the probability is decreased. Average keyframe distance is defined as the average shots duration between the last *n* scene cuts. The threshold used for the keyframe distance can be varied from 6 to 10 for good results. Again, segments which are close by can be linked to infer the whole commercial break. There are commercials such as Calvin Klein which are very slow and this can increase the average cut distance temporarily. We allow for the keyframe distance to be high for 30 seconds before decreasing the probability of being in a commercial break. As with the black frame indicators, we have placed constraints on the duration of the commercials. Other mid-level features can also be extracted from MPEG-2 encoding parameters, and be used as verifiers. Progressive versus interlaced video material changes or coding cost are some examples. As can be seen, there are a number of thresholds that need to be experimented with for algorithm fine tuning. In the next section we explain the experiments that we carried out in order to determine the optimal thresholds for eleven different parameters for best accuracy.

6 THE EXPERIMENTS

We obtained two samples of broadcast video. One contains about 8 hours of Dutch television and comprised 13 different TV programs of various genres including movies, news, sports programs, talk shows, and sitcoms. It contains 28 different commercial breaks, for a total duration of about 1.5 hours of commercials. This video was in PAL format with a GOP[3] of six. We refer to it as the EMPIRE set after the name of the MPEG video encoder chip that extracted its low-level features. The other set contained about 5 hours of US content from 11 different programs including sports, movies, games, talk shows, MTV music videos, and news. It contained 35 different commercial breaks, for a total duration of more

[3] GOP is Group of Pictures and six means that every sixth frame was an I frame.

than 1.5 hours of commercials. This video was in NTSC format with a GOP of 12 and was called the EMPRESS set, again after the encoder chip.

These video sequences were scanned by a human and the beginning and end of each commercial break was marked. Thus we obtained a ground truth value (yes/no) for every video frame. The performance of any instantiated detector algorithm was assessed by counting the number of true positives (TP, commercial frames labeled as such), false positives (FP) and false negatives (FN). From these we computed the usual measures of recall and precision used in pattern recognition:

$$Recall = \frac{TP}{TP + FN} \qquad Precision = \frac{TP}{TP + FP}$$

Since we ran experiments with a scalar GA, we experimented with a number of different combinations of these data as fitness metrics. It should be noted that because the commercial detector algorithm has built-in contiguity constraints, isolated error frames do not occur – to be labeled as a commercial, a block of frames must fall between certain minimum and maximum limits.

To guard against any optimization procedure's tendency to overfit the training data, we split the data set into training and test sets. For the EMPIRE data set, we split each TV show roughly in half, approximately 50% of the shows had the data taken from first-half of the show while for the other 50% had the data from the second half used in the training set. A slightly different procedure was followed for the EMPRESS data set: within each genre, shows were paired for similarity (human judgement) and one whole show of each pair was used for training. The test set was always the complement of the training set. When we sought to validate any given detector, we ran the algorithm on the combined test and training sets. This we call the validation set – which was the set used by the engineer when manually tuning the algorithm.

The GA used was Eshelman's CHC [4]. CHC is a generational style GA with three distinguishing features. First, selection in CHC is monotonic: only the best *M* individuals, where *M* is the population size, survive from the pool of both the offspring and parents. Second, CHC prevents parents from mating if their genetic material is too similar (i.e., incest prevention). Controlling the production of offspring in this way maintains genetic diversity and slows population convergence. Finally, CHC uses a soft-restart mechanism. When convergence has been detected, or the search stops making progress, the best individual found so far in the search is preserved. The rest of the population is reinitialized, using the best string as a template and flipping some percentage (i.e., the divergence rate) of the template's bits. This is known as a soft-restart and introduces new diversity into the population to continue search. CHC does not use mutation between restarts. We used the recommended parameter settings (e.g popsize 50, normal triggers for the dropping of the incest threshold and the initialization of soft-restarts [4]) with one exception: the divergence rate was 0.5. This means that a soft restart created a population with one copy of the best-so-far individual and the rest of the population was completely re-randomized. Initial experiments suggested that a lower divergence rate leads to the population quickly stagnating with inferior results.

6.1 EMPIRE EXPERIMENTS

The EMPIRE data were in hand first and had served as the data upon which the commercial detection algorithm was originally developed [3]. Hence, for this data set we had the algorithm developed and tuned by the engineer as a benchmark. As the first experiments performed, we wanted to get an idea what fitness measure would be best to use. The set of algorithm parameters (the genes in the chromosome) for these experiments are listed in Table 1. Note that the last three parameters were used only in experiment 5. Parameter 1= SeparationThreshold, 2= DistForSuccThreshold1, 3= DistForSuccThreshold2, 4= DistForSuccThreshold3, 5= UnicolorInSuccThreshold, 6= MinCommThreshold, 7= MaxCommThreshold, 8= RestartThreshold, 9= BlackIFrameThreshold, 10= UnicolorIFrameThreshold, 11=LowInfoIFrameThreshold.

Table 1: Parameters Used in EMPIRE Experiments

PARAMETER	BITS	MIN-MAX	STEPS	EXPERIMENTS
1	6	100 - 3250	50	1-5
2	3	50 - 225	25	1-5
3	3	50 - 225	25	1-5
4	3	50 - 225	25	1-5
5	4	1 - 16	1	1-5
6	3	500 - 5750	750	1-5
7	2	7000 - 10000	1000	1-5
8	3	250 - 775	75	1-5
9	4	1 - 16	1	5
10	3	100 - 450	50	5
11	4	30 - 105	5	5

The experiments 1-5 are briefly summarized in Table 2 where R stands for recall and P for precision, FP and FN are false positives and false negatives respectively. The only difference among experiments 1-4 was the fitness metric used by the GA for selection: R+P, R*P, FP+FN, 4*FP+FN. There was intuitive reasoning for having multiple fitness metrics backed up by experiments. R*P should be sensitive to either measure being small (both recall and precision are numbers between zero and one). FP + FN seemed a reasonable metric to minimize, but

perhaps in this domain, false positives should be weighted more heavily than false negatives (better to let a commercial through than risk cutting out some TV program content). Consequently, we tried 4*FP + FN. Then the simple R+P was included for completeness. Experiment 5 used the R+P fitness metric for reasons given below and added three additional parameters in an attempt to improve the commercial detector. We should also note that the only reason experiments 2 and 3 did not achieve 30 replications was that power failures caused the computers to crash and we observed that all experiments located chromosomes with the same best performance – 12 replications were obviously enough.

Although all experiments discovered the same best performance level, there was still diversity in the final populations indicating that performance was less sensitive to some parameters than others.

Table 2. Summary of Experiments with the EMPIRE Data Set. R-Recall, P=Precision, FP= Fales Positives, and FN = False Negatives

EXP. NUM	# OF PARAMS	# OF BITS	REPLICATIONS	FITNESS METRIC
1	8	27	30	R*P
2	8	27	12	4FP+FN
3	8	27	12	FP+FN
4	8	27	30	R+P
5	11	38	30	R+P

One way to examine the outcomes of these experiments is to look at the non-dominated individuals encountered at any time in each experiment. Figure 2 shows these data in the Precision/Recall space. Since the experimental data were all generated on the training data set, some of the best performers were selected (by hand) and run on the complete set of validation data. These points are shown as + signs in the figure and each is connected by a dashed line to its corresponding training set performance point. While not statistically rigorous, these observations do suggest the region in performance space where those commercial detectors are likely to perform. The performance of the engineer's best manually tuned algorithm is also shown, this time as a * symbol. (located at recall 0.83 and precision 0.95.)

It is interesting to observe that the non-dominated set was identical for experiments 1, 2, and 3 even though they used different fitness metrics. We speculate that this is because the parameter sets that yield these R and P values are readily produced even though the selection pressures on the populations are slightly different. It may also indicate that, although the fitness metrics appear to be different, the ranking of the individuals involved is not different; the discretization of the search space may

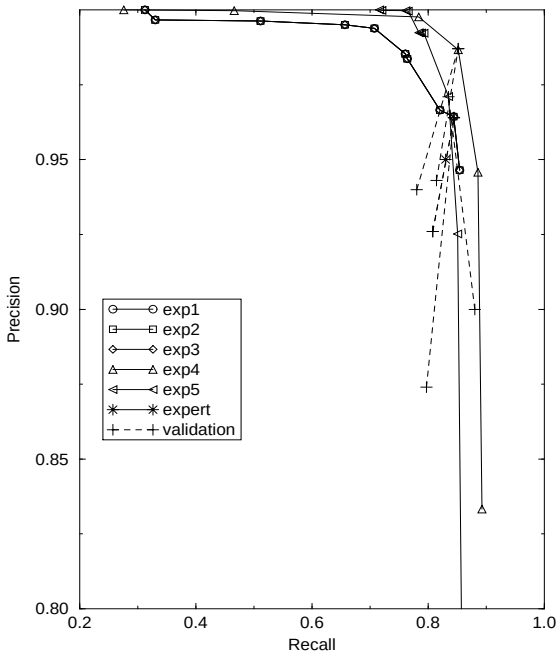

Figure 2. Results from EMPIRE experimental set

simply not permit that much variety. It also appears that the results from experiments 1, 2, and 3 were inferior to those from experiment 4. The comparison of experiment 4 with 1, 2, and 3 caused us to select the R+P metric for all subsequent experiments. The addition of the extra genes in experiment 5 seems not to have provided significant improvement.

The dashed lines between the test and validation performances are designed to suggest the expected intervals for the respective commercial detectors. We do see that the evolved detectors are performing in same region with the best performance achieved by an expert engineer after many months of tinkering.

6.2 EMPRESS EXPERIMENTS

Experiments 6 and 7 were performed with the EMPRESS data set. This data set was acquired after the above work had already been done and may serve as a test of our claim that a robust GA is a valuable tool for adjusting a commercial detector to new data. The parameters used in these experiments are summarized in Table 3. The last column of Table 3 summarizes the location of the best performers in parameter space. We can see that some parameters are very sensitive (all bests have the same allele value) and some are completely insensitive (all permitted values are present among the bests). Parameter 1= SeparationThreshold, 2= DistForSuccThreshold1, 3=

UnicolorInSuccThreshold, 4= MinCommThreshold, 5= MaxCommThreshold, 6= AdjSceneCutsThreshold, 7= AverageCutDistThreshold, 8= CutNumberInAverage, 9= BlackIFrameThreshold, 10= LetterboxLengthThreshold, 11= LetterboxThreshold.

Table 3. Parameters Used in EMPRESS Experiments

PARAMETER	BITS	MIN-MAX	STEP SIZE	EXP NUM	BEST SOLUTION
1	5	100 - 7850	250	6-7	100-7850
2	6	100 - 6400	100	6-7	4500-4700
3	3	0 - 7	1	6-7	0
4	5	100 - 7850	250	6-7	100-850
5	3	7000 - 14000	1000	6-7	13000-14000
6	5	100 - 1650	50	6-7	3
7	4	100 - 850	50	6-7	100-850
8	4	2 - 17	1	6-7	2-17
9	3	80000 - 115000	5000	6-7	95000
10	5	10 - 320	10	7	150-160
11	5	7500 - 32000	500	7	20500-21500

Experiments 6 and 7 conducted for the EMPRESS dataset are briefly summarized in Table 4 and the comparable results are shown in Figure 3.

Table 4. Summary of Experiments with the EMPRESS Data Set

EXP. NUM	# OF PARAMS	# OF BITS	REPLICATIONS	FITNESS METRIC
6	9	38	30	R+P
7	11	48	30	R+P

Figure 3 tells two intersecting tales. Experiment 6 was the first run on the EMPRESS data set. Independently, the expert searched for (using the entire validation set) and reported a detector whose performance on the validation data is shown as the leftmost asterisk in the figure. We see that the GA located points that look superior to the expert's, but these points may represent overfittting to the training data. When we run two (manually-selected) of the GA's detectors on the validation set, we see that performance deviates and no longer dominates the expert's point. At this point, we experimented with additional features that assess the likelihood that the video frame is in letterbox format. Believing this may be valuable additional information for commercial detection, the detector algorithm was augmented with logic that considered this new feature. The expert, being in possession of the experiment 6 results, used them as the starting point for an effort to discover good parameter settings for the new letterbox thresholds. The result of this effort (again using the entire validation set) is the righmost asterisk in Figure 3. In experiment 7 the GA searched the augmented threshold set. Two validation tests from this run are also shown. We see that the GA located the best performance seen to date. We believe this shows the potential of the GAs for this emerging application.

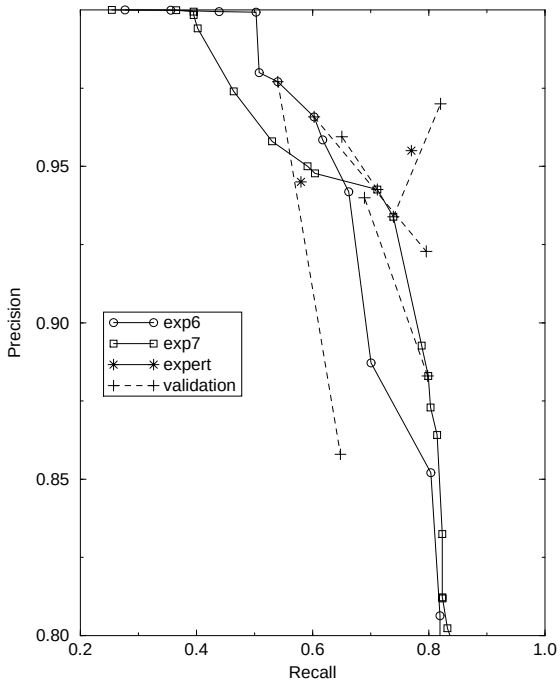

Figure 3. Results of the EMPRESS experimental set

7 DISCUSSION

We have presented results from first experiments using a GA to fine tune the parameters for a commercial detection algorithm for digital video.

We propose that this approach is particularly well suited to this domain because 1) the complexity of the mapping from algorithm parameters to accuracy is unknown, but unlikely to be very simple, 2) it is unlikely that the best parameter sets to use will be the same from culture to culture or for different video standards, 3) the low-level

features available from different encoder chips is likely to change and 4) the mapping is unlikely to remain unchanged over time (perhaps because of deliberate attempts to avoid automatic detection). These properties all suggest that an automated method to readjust the detector algorithm will be needed in the industry. Dynamically downloading new parameters to products in the field is already being practiced.

From our experiments we learned that we can benchmark and fine tune the performance of the commercial detection algorithm rapidly for a new set of data. This means that we used the GAs as a tool for mapping from algorithm parameters to accuracy. In addition, this allowed us to experiment with new parameters (e.g. letterbox in experiment 7) and obtain an optimized version of the algorithm without spending additional weeks of effort.

The class of algorithms explored was limited, constrained by a desire to use algorithms that could be driven by low-level features immediately available from MPEG encoder chips and that could run on the modest computing resources available in today's consumer electronic products. In addition, only a limited amount of video material was available for these initial experiments.

There are several directions in which to continue this work. Clearly more validation work is needed before the utility of the detectors can be assessed against levels needed for consumer acceptance. In addition, a broader class of detector algorithms could be explored, including allowing the GA to explore this dimension in addition to simply tuning parameters. A truly multi-objective GA also should be tried.

References

1. D. W. Blum, "Method and Apparatus for Identifying and Eliminating Specific Material from Video Signals, " US patent 5,151,788, September 1992.
2. Edgar L. Bonner and Nelson A. Faerber, "Editing system for video apparatus," US patent 4,314,285, February 1982.
3. N. Dimitrova, S. Jeannin, J. Nesvadba, T. McGee, L. Agnihotri, G. Mekenkamp, "Real time commercial detection using MPEG features," Information Processing and Management Uncertainty in Knowledge-based System, IPMU 2002, Annecy, France, July 1-5, 2002.
4. L.J. Eshelman, "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination," Foundations of Genetic Algorithms, Gregory Rawlins (ed.), Morgan Kaufmann, San Mateo, 1991.
5. S. J. Forbes, F. F. Forbes, "Video Signal Identifier for Controlling a VCR and Television based on the Occurrence of Commercials," US patent 5,708,477, Jan. 13, 1998.
6. J. Iggulden, K. Fields, A. McFarland, J. Wu, "Method and Apparatus for Eliminating Television Commercial Messages, " US Patent 5,696,866, Dec. 7, 1997.
7. Y. Li and C.C.J. Kuo, "Detecting commercial breaks in real TV programs based on audiovisual information," Proc. Of SPIE The International Society for Optical Engineering (USA), vol.4210, p.225-236, 2000.
8. R. Lienhart, C. Kuhmunch and W. Effelsberg, "On the Detection and Recognition of Television Commercials," in Proc. Of IEEE International Conference on Multimedia Computing and Systems, pp. 509-516, 1997.
9. T. McGee, N. Dimitrova, "Parsing TV programs for identification and removal of non-story segments," SPIE Conference on Storage and Retrieval in Image and Video Databases VII, vol. 3656, pp. 243-251, San Jose, 1999.
10. J. Nafeh, "Method and Apparatus for Classifying patterns of Television Programs and Commercials Based on Discerning of Broadcast Audio and Video Signals," US patent 5,343,251, Aug. 30, 1994.
11. A. P. Novak, "Method and System for Editing Unwanted Program Material from Broadcast Signals," US patent 4,750,213, Jun. 7,1988.