Advertisement Detection and Replacement using Acoustic and Visual Repetition

Michele Covell and Shumeet Baluja
Google Research, Google Inc.
1600 Amphitheatre Parkway
Mountain View CA 94043
Email: {covell,shumeet}@google.com

Michael Fink
Interdisciplinary Center for Neural Computation
Hebrew University of Jerusalem
Jerusalem 91904 Israel
Email: fink@huji.ac.il

***Abstract*— In this paper, we propose a method for detecting and precisely segmenting repeated sections of broadcast streams. This method allows advertisements to be removed and replaced with new ads in redistributed television material. The detection stage starts from acoustic matches and validates the hypothesized matches using the visual channel. Finally, the precise segmentation uses fine-grain acoustic match profiles to determine start and end-points. The approach is both efficient and robust to broadcast noise and differences in broadcaster signals. Our final result is nearly perfect, with better than 99% precision, at a recall rate of 95% for repeated advertisements.**

I. Introduction

When television material is redistributed by individual request, the original advertisements can be removed and replaced with new ads that are more tightly targeted to the viewer. This ad replacement increases the value to both distributor *and viewer.* The new advertisements can be fresher, by removing promotions for past events (including self-advertisement of past program material), and can be selectively targeted, based on the viewer's interests and preferences.

However, information about the original broadcast ads and their insertion points is rarely available at redistribution. This forces consideration of how to efficiently and accurately detect and segment advertising material out of the television stream.

Most previous approaches have focused on heuristics based on common differences between advertising and program material [1], [2], [3], such as cut rates, soundtrack volume, and surrounding black frames. However, these approaches seldom work in detecting self-advertisement of upcoming program material.

Instead, we compare the re-purposed video to an automatically created, continuously updated database of advertising material. To create the advertising database, we first detect repetitions across (and within) the monitored video streams. We use fine-grain segmentation (Subsection II-C) to find the exact endpoints for each advertising segment. We then add this advertisement to the database, noting the detected endpoint to the ad. When processing the re-purposed video to replace embedded advertising, we can skip the fine-grain segmentation step. Instead, we can simply use the noted advertisement endpoints, projected through the matching process back onto the re-purposed video. With these endpoints on the re-purposed video stream, we can replace the embedded advertisement with a new advertisement that is still timely and that matches the viewers interests.

In this approach, the two difficult steps are (1) creating a database of accurately segmented advertisements and (2) selecting an approach to repetition detection that is efficient, distinctive, and reliable. We create the advertising database by continuously monitoring a large number of broadcast streams and matching the streams against themselves and each other in order to find repeated segments of the correct length for advertisement material. Since we use the same matching process in creating our advertisement database as we ultimately will use on our re-purposed video stream, we discuss this shared matching techniques as part of our description of the creation of the advertisement database.

While the basic repetition-based approach to detecting advertising is similar to the general approach taken by Gauch *et al.* [4], there are a number of important distinctions. The approach taken by Gauch *et al.* relies on video signatures only for matching. Our approach is based primarily on audio signatures, with video signatures used only to remove audio matches of coincidental mimicry. Furthermore, Guach *et al.* start by segmenting their video stream before detecting repetitions. This may make the segmentation process more error prone. We proceed in the opposite order, first detecting repetitions and using these signals to determine the temporal extent of the repeated segment. We believe that these two differences (the matching features and the order of detection and segmentation) lead to improved performance, compared to that reported by Guach *et al.* [4].

For creating and updating the advertising database and for detecting ads in re-purposed footage, our detection process must be efficient; otherwise, this approach will not be practical on the volume of data that is being processed. For removing and replacing ads in re-purposed footage, we need an extremely low false-positive rate; otherwise, we may remove program (non-ad) material. Finally, our segmentation must be accurate at video-frame rates to avoid visual artifacts around the replaced material.

In this paper, we propose a method that meets these criteria, for detecting and segmenting advertisements from video streams. We describe this approach in the next section. We present our experimental results for each portion of the pro-

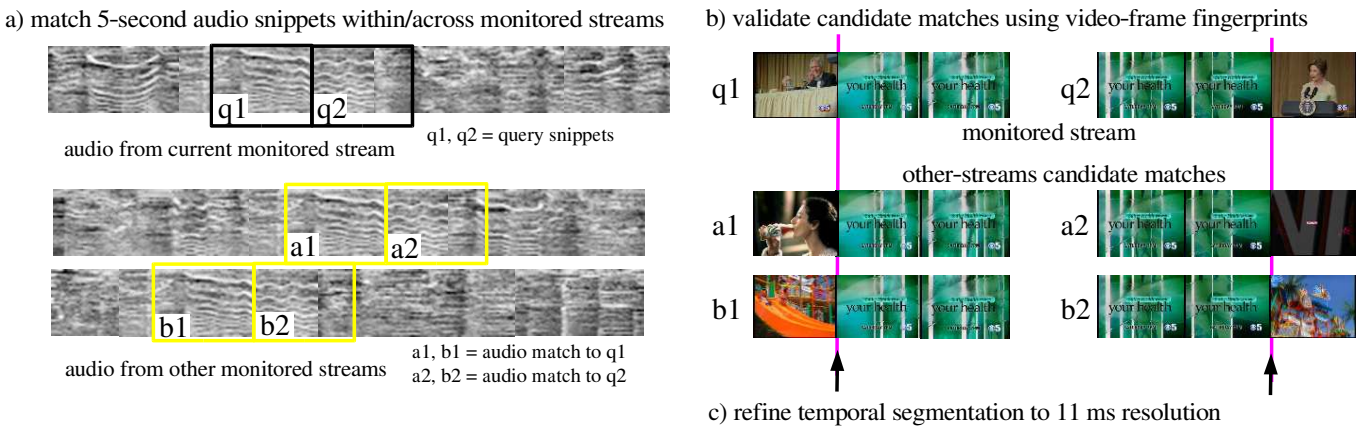

Fig. 1: Overview of the detection, verification, and segmentation process: (a) Five-second audio queries from each monitored broadcast stream are efficiently detected in other broadcast streams (and at other points in the same broadcast stream), using a highly discriminative representation. (b) Once detected, the acoustic match is validated using the visual statistics. (c) A refinement process, using dynamic programming, pinpoints the start and end frames of the repeated segment. This process allows the advertisement database to be continuously updated with new, segmented, advertisement material. The same matching/validation process (steps a and b) is used on the re-purposed video footage, with the addition that the endpoints for replacing the ads in the re-purposed video footage can be inferred using the segmentation found when inserting the advertisement into the database.

posed process in Section III and conclude with a discussion of the scope, limitations, and future extensions of this application area in Section IV.

II. Proposed Method

We use a three-stage approach to efficiently localize repeated content. First, we detect repetitions in the audio track across all monitored streams (Figure 1-a). We then validate these candidate matches using a fast matching process on very compact visual descriptors (Figure 1-b). Finally, we find the starting and ending points of the repeated segments (Figure 1-c). The detection stage finds acoustic matches across all monitored streams. The validation stage only examines the candidates found by the detection stage, making this processing extremely fast and highly constrained. The last stage segments each advertisement from the monitored streams using the fine-grain acoustic match profiles to determine the starting and ending points. These segmented ads are placed in the advertising database for subsequent use in removing ads (by matching) from re-purposed footage.

We use an acoustic matching method proposed by Ke *et al.* [5] as the starting point for our first-stage detection process. We review this method in Section II-A. While the acoustic matching is both efficient and robust, it generates false matches, due to silence and reused music within television programs. We avoid accepting these incorrect matches by using a computationally efficient visual check on the hypothesized matches, as described in Section II-B. The accepted matches are then extended and accurately segmented using dynamic programming, as described in Section II-C.

A. Audio-Repetition Detection

The most difficult step in creating an advertisement database from monitored broadcasts is determining, accurately and efficiently, what portions of the monitored streams are advertisements. We include in this set of ads "self-advertisements" (*e.g.*, for upcoming programming). These ads for upcoming installments typically cannot be detected using standard heuristics [2], [3] (duration, black frames, cut rate, volume).

This leads us to use repetition detection. When material in any monitored video stream is found elsewhere within the monitored set, the matching material is segmented from the surrounding (non-matching) footage and is considered for insertion into the advertisement database. In this way, we continuously update the advertising database, ensuring that we will ultimately be able to detect even highly time-sensitive advertisements from the re-purposed footage.

In order to handle the large amount of data generated by continuously monitoring multiple broadcasts, our detection process must be computationally efficient. To achieve this efficiency, we use acoustic matching to detect potential matches and use visual matching only to validate those acoustic matches. Acoustic matching is more computationally efficient than visual matching due to the lower complexity decoders, lower data rates and lower complexity discriminative-feature filters. We adapted the music-identification system, proposed by Ke *et al.* [5] to provide these acoustic matches.

We start with one of the monitored broadcast streams and use it as a sequence of probes into the full set of monitored broadcast streams (Figure 1-a). We split this probe stream into short (5-second) non-overlapping snippets, and attempt to find matching snippets in other portions of the monitored broadcasts. Because of noise in the signal (both in the audio and video channels), exact matching does not work, even within a single broadcast. This problem is exacerbated when attempting matches across the many monitored broadcast channels.

To match segments in broadcasts, we start with the music-

identification system proposed by Ke *et al.* [5]. This system computes a spectrogram on 33 logarithmically-spaced frequency bands, using 0.372-second slice windows at 11.6-ms increments. The spectrogram is then filtered to compute 32 simple first- and second-order differences at different scales across time and frequency. This filtering is calculated efficiently using the integral image technique suggested by [6]. The filter outputs are each thresholded so that only one bit is retained from each filter at each 11.6-ms time step. Ke *et al.* [5] used a powerful machine learning technique, called *boosting*, to select these filters and thresholds that provide the 32-bit descriptions. During the training phase, boosting uses the positive (distorted but matching) and negative (not-matching) labeled pairs to select the combination of filters and thresholds that jointly create a highly discriminative yet noise-robust statistic. The interested reader is referred to Ke *et al.* [5] for more details.

To use this for efficient advertisement detection, we decompose these sequences of 32-bit identifying statistics into non-overlapping 5-second-long query snippets. Our snippet length is empirically selected to be long enough to avoid excessive false matching, as may be found from coincidental mimicry within short time windows. The snippet length is also chosen to be short enough to be less than $\frac{1}{2}$ of the shortest-expected advertising segment. This allows us to query using non-overlapping snippets and still be assured that at least one snippet will lie completely with the boundaries of each broadcast-stream advertisement.

Within each 5-second query, we separately use each 32-bit descriptor from the "current" monitored stream to identify "offset candidates" in other streams or in other portions of the same stream. The offset candidates describe the similar portions of the current and matching streams using (1) the starting time of the current query snippet, (2) the time offset from the start of the current query snippet to the start of the matched portion of the other stream, and (3) the time offset from those starting times to the current 32-bit descriptor time. We then combine self-consistent offset candidates (that is, candidates that share the same query snippet (item 1) and that differ only slightly in matching offset (item 2)) using a Markov model of match-mismatch transitions [5]. The final result is a list of audio matches between each query snippet and the remainder of the monitored broadcasts.

Although this approach provides accurate matching of audio segments, similar sounding music often occurs in different programs (*e.g.*, the suspense music during "Harry Potter" and some soap operas), resulting in spurious matches. Additionally, silence periods (between segments or within a suspenseful scene) often provide incorrect matches. The visual channel provides an easy method to eliminate these spurious matches, as described in Section II-B.

B. Visual Verification of Audio Matches

Television contains broadcast segments that are not locally distinguishable using only audio. These include theme music segments, stock music segments (used to set the emotional tone at low cost), and silence periods (both within suspenseful segments of a program and between segments).

We use a simple procedure to verify that segments which contain matching audio tracks are also visually similar. Although there are many ways of determining visual similarity, the requirements for our task are significantly reduced from the task of general visual matching. We are only looking for exact (to within systematic transmitter and receiver distortions) matches. Furthermore, the audio matching already finds only matches that are acoustically similar (again, to within systematic transmitter and receiver distortions). Since an audio match has already been made, the hypothesized match is likely to be one of two cases: (1) different broadcast of the same video clip or (2) 'stock'-background music that is used in a variety of scenarios. In the latter case, the case that we need to eliminate, we observed little evidence that the visual signal associated with the same background sounds will be similar. For example, Figure 2 shows a sequence that matched in the audio track, but contained very different visual signals.

Given this simplified task, the visual matching can be easily implemented, not requiring the complexity (and associated computation) of more sophisticated image matching techniques [7]. Each frame in the two candidate sequences is reduced to a 5×5 24-bit RGB image. The only preprocessing of the images is subtraction, from each color band, of the overall mean of the band; this helps eliminate intensity and other systematic transmitter/receiver distortions. We use the L_2-norm distance metric on these reduced visual representations.

We examined the verification performance using four alternative methods for keyframe-sequence matching: with and without replacement and with and without strict temporal ordering. Matching with replacement allows for a larger degree of audio-visual desynchronization within the potential matches. Matching without temporal constraints is more robust to partial matches, where some number of keyframes do not have a good visual match. These results are given in the next section.

We found that sampling the visual match 3 times a second taken from the middle 80% of the detected match was sufficient for this visual verification of the acoustic match. Using only the center 80% of the match helps reduce the sensitivity to partial matches, where the candidate match straddled the segment boundary. Temporal subsampling to only 3 frames per second allows us to reduce the temporal resolution (and therefore size) of that visual database. In the visual statistics database, we only include the signature data from every tenth frame. When testing a match hypothesis that was generated from the acoustics, we then pull out the frames from the to-be-segmented stream that, using the match offset, will line up with those frame times in the database streams.

C. Segment recovery

Those matches that pass both acoustic and visual consistency checks are hypothesized as being parts of advertisements. However, there still are two limitations in our snippet

a. match between different programs with similar music

b. match different positions within a single program

Fig. 2: Two sequences that matched acoustically but not visually. These incorrect matches are removed by the visual verification.

matches: (1) the individual matches may over-segment an advertisement sequence and (2) the match boundaries will only coarsely locate the advertisement boundary.

We correct both of these shortcomings by endpoint detection on the temporal profiles created by combining the fine-grain acoustic match confidences across all matching pairs. For each 5-second snippet from the current probe video, we collect a list of all the times/channels to which it matched, both acoustically and visually. We force this multi-way match to share the same start and end temporal extent, as measured from the center of the snippet and its matches. A single profile of fine-grain match scores for the full list is created by, at each 11-ms frame, using the minimum match similarity generated by the match pairs within the current list. This typically increases the accuracy of segmentation when the transitions to or from the ad are silent or are theme music. The increased accuracy is seen whenever the monitored footage has some other occurrence of the same ad with a different surrounding context.

We use forced Viterbi [8] starting from the center of the snippet match and running forward in time to find the end point of the ad segment. We use it starting from the center of the snippet match and running backward in time to find the start point of the segment. In each case, we use a two-state first-order Markov model and find the start/end point by finding the optimal transition point from "matching" to "not matching", given the minimum-similarity profile. The Viterbi decoder is allowed to run for 120 seconds forward (or backward) in time from the match center. At each time step, the decoder tracks two probabilities and one "decoding variable". The first probability is that the profile from the center point to that time step matches. The second is the probability of the *most-likely path from matching to not matching,* assuming that the current time step does not match. The decoding variable gives the maximum-likelihood transition point under this second scenario.

By running the Viterbi decoder forward (or backward) for 120 seconds, starting from the match certainty at the center, we can examine the relative probabilities of the match still being valid or invalid, after 120 seconds. If the full match profile (from the detected starting point to the detected ending point) extends for 2 minutes or more, it is most likely a repeated program. Since we are unlikely to be matching advertisements over such a long period, we can safely remove that over-long match from consideration. Otherwise, we use the location indicated by the decoding variable as our transition point and are assured of using the optimal end (start) point for our segments. Finally, if the duration given by combining the optimal start and end points is too short (less than 8 seconds), we also discarded the match list as being simple coincidences.

III. Experimental Results and Discussion

In this section, we provide a quantitative evaluation of our advertisement identification system. For the results reported in this section, we ran a series of experiments using 4 days of video footage. The footage was captured from three days of one broadcast station and one day from a different station. We jack-knifed this data: whenever we used a query to probe the database, we removed the minute that contained that query audio from consideration. In this way, we were able to test 4 days of queries against 4 days (minus one minute) of data.

We hand labeled the 4 days of video, marking the repeated material. This included most advertisements (1348 minutes worth), but omitted the 12.5% of the advertisements that were aired only once during this four-day sample. In addition to this repeated advertisement material, our video included 487 minutes of repeated programs, such as repeated news programs or repeated segments within a program (*e.g.*, repeated showings of the same footage on a home-video rating program).

For the results reported in Subsections III-A (acoustic matching) and III-B (visual verification), the performance statistics are for the detecting any type of repeated material, *both* advertising and main programming: missed matches between repeated main-program material are counted as false negatives and correct matches on these regions are counted as true positives. For the results reported in Subsection III-C (segment recovery), the performance statistics are for detecting repeated advertising material only: for this final step, any program-material matches that remain after the segment-recovery process are counted as false positives.

A. Acoustic-Matching Results

Our results on our acoustic matching step, using non-overlapping 5-second queries is shown in the top row of Table I. Since no effort was made to "pre-align" the query boundaries with content boundaries, about $\frac{1}{6}$ of the queries straddled match-segment boundaries. For these straddle-queries,

TABLE I: Results from each stage of our advertisement detection. Only the performance listed as our final results have a visible effect on the re-purposed video stream. However, the quality of the acoustic-matching and visual-verification results have a direct effect on the computational efficiency of the final system. For example, if the acoustic-matching stage generates many false matches (that are removed by one of the later stages), the computational load for the visual verification stage goes up.

Stage and	*detection target*	False-positive rate	False-negative rate	Precision	Recall
Acoustic-matching stage	*all repeated material*	6.4%	6.3%	87%	94%
After visual verification	*all repeated material*	3.7–3.9%	6.6–6.8%	92%	93%
Final results, after fine-grain segmentation	*repeated advertising only*	$< 0.1\%$	5.4%	$> 99\%$	95%

False-positive rate = FP/(TN+FP). False-negative rate = FN/(TP+FN). Precision = TP/(TP+FP). Recall = TN/(TP+FN).

we counted each match or missing match as being correct or not based on what type of content the majority of the query covered. That is, if the query contained 3 seconds of repeated material and 2 seconds of non-repeated material, then the ground truth for that query was "repeated" and vice versa. As shown in Table I, our precision (the fraction correct from the material *detected as* repeating) is 87% and the recall (the fraction correct from the material *actually* repeating) is 94%, even with these difficult boundary-straddling snippets. Many of the false positives and false negatives (27% and 42%, respectively) were on these boundary cases. These false-positive and false-negative rates are 60% and 150% higher than seen on the non-boundary snippets, respectively.

On the non-boundary cases, most of the false positives were due to silences within the television audio stream. Some false positives were also seen on segments that had stock music without voice overs that were used in different television programs. On the non-boundary cases, the false negatives seemed to be due to differences in volume normalization. These were seen near (but not straddling) segment boundaries when the program material just before or after the match on the two streams were set to radically different sound levels.

B. Visual-Verification Results

As can be seen in Table I, the performance of our visual verification step was nearly identical under all four of the sequence-matching approaches (with or without temporal ordering and with or without replacement). In all cases, the false-positive rate dropped to between 3.7% and 3.9% and the false-negative rate rose slightly to between 6.6% and 6.8%, giving a precision of 92% and a recall of 93%. This is a relative improvement in the precision of 40%, associated with a relative degradation in recall rate of 10%.

As mentioned above, the different matching metrics did not provide significant differences in performance. All four metrics correctly excluded incorrect matches that were across unrelated program material, such as shown in Figure 2-a. The two metrics with temporal constraints performed better on segments that were from different times within the same program, such as might occur during the beginning and ending credits of a news program (Figure 2-b) but were more prone to incorrectly discarding matches that included small amounts of unrelated material, such as occurs at ad/ad or ad/program boundaries. When thresholds were selected to give equal recall rates across the difference sequence-matching approaches, the associated false-positive rates were all within $\frac{1}{2}\%$ of one another.

Due to the nearly equal performance, we selected our sequence-matching technique according to computational load. Matching with temporal constraints and without replacement takes the least computation, since there is only one possible mapping from one sequence to the other. All of the other criteria require comparison of alternative pairings across the two sequences.

C. Segment Recovery

We used the approach described in Section II-C to recover advertising segments. Since we discard match profiles that are longer than 120 seconds, we collected our performance statistics on the ad repetitions only: the repetitions associated program reruns were all long enough that we discarded them using this test.

As can be seen from Table I, all performance measures improved with fine-grain segmentation. The false-positive rate fell by 97%, relative to that seen after that visual-verification stage. At the same time, the false-negative rate fell, relative to that seen after that visual-verification stage, by 20%. The corresponding improvements in precision and recall were 98% and 32%, relative to those seen after the visual-verification stage. The improvement in the precision was due to the use of the minimum similarity profiles to determine repetition. The improvement in the recall rate was due to the match profile from neighboring matches correctly extending across previously-missed matches on straddled segment (ad/ad or ad/program) boundaries. Note that this improvement recovers the loss in recall introduced by the visual-verification stage and even improves the recall to better than seen on the original acoustic-matching results.

Our results improve significantly on those reported previously. For commercial-detection, Hua *et al.* [1] report their precision and recall as 92% on a *10* $\frac{1}{2}$*-hour* database. Gauch *et al.* [4] reports combined precision and recall, F_C. The formula suggested by Gauch *et al.* [4] is $\frac{2PR}{P+R}$ where P and R are precision and recall. For this metric, for commercial detection, Gauch reports $F_C = 95\%$ on a *72-hour* database.[1] For a similar combination of precision and recall, we achieve a quality metric of 97% on a *96-hour* database. By this metric, our results provides a relative improvement of 40-62% even

[1]Since Hua *et al.* [1] report equal precision-recall results of 92%, their $F_C = 92\%$.

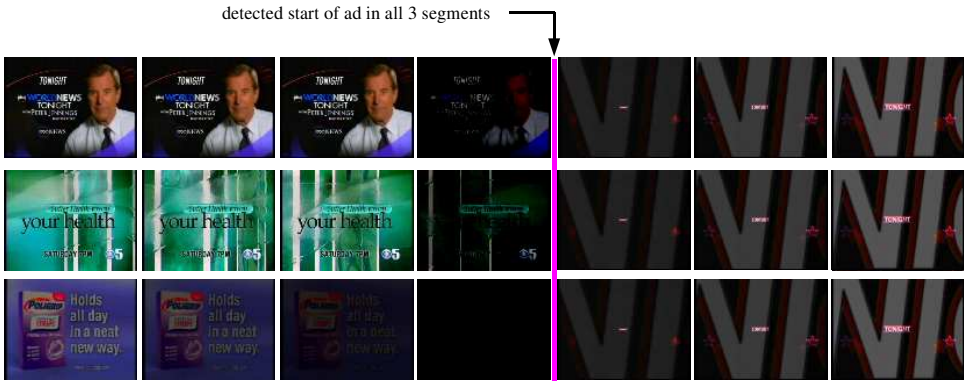

Fig. 3: Segmentation result for the start of an advertisement across 3 broadcast streams. Each row shows the frames from a different broadcast stream. The figure shows full video-rate time resolution (all video frames are shown). The detected endpoint was indicated using Viterbi decoding of the optimal transition point, given a temporal profile of the minimum match similarity on each 11-second audio frame period. Note the frame accuracy of ad-boundary detection. Also note that the transition does not always include a black frame, making that common heuristic less reliable in detection of advertising boundaries.

on a database that is larger than the previously reported test sets.

Our detected segment boundaries are also very accurate. Figure 3 shows an example of our segmentation results, on a set of aligned repetitions of an ad. The use of minimum similarity measures allows the correct transition point to be detected, even when the previous segments are faded down before the start of the new segment. When we replayed the video with the advertising segments removed, we saw no flashes or visual glitches. There was the perception of an acoustic pops, probably due to the cut-induced sudden change in the background levels. These acoustic artifacts could be avoided by cross fading instead of splicing the audio across the ad removals.

IV. Conclusions and Future Work

We have presented an approach to detecting and segmenting advertisements in re-purposed video material, allowing fresher or specifically targeted ads to be put in the place of the original material. The approach that we have taken was selected for computational efficiency and accuracy. The acoustic matching process can use hash tables keyed on the frame descriptors to provide the initial offset hypotheses. Only after these hypotheses are collected is the overhead of the visual decompression and matching incurred. Since the acoustic matching provides strong support for a specific match offset, the visual matching does *not* need to be tuned for discriminating between neighboring frames (which is difficult due to temporal continuity in the video). Instead the visual matching need only test for clear mismatches, such as occur when stock music is reused.

Once the original advertisements are located (and removed), new (potentially targeted) ads can be put into their place, making the advertisements more interesting to the viewer and more valuable to the advertiser. By using the original ad locations for the new ads, we avoid inserting ads at arbitrary locations in the program content. This ability to remove stale ads and replace them with targeted, new ads may be a crucial step in ensuring the economic viability of alternative TV-content distribution models.

There are numerous possibilities for extending this work. Foremost is using this in conjunction with a full advertisement-replacement system, and determining not only the technical limitations when employed on a large scale, but also end-user satisfaction. Secondly, deployment on a large scale allows us to build a database of advertisements from which we can build more intelligent classifiers, for example to determine broad interest/topic-categories, that may help us determine which new advertisements to insert. Repeated-occurrence statistics will also give the ability to autonomously monitor and analyze advertiser trends, including spend and breadth, across broadcast channels and geographies.

Acknowledgements

The authors would like to gratefully acknowledge Y. Ke, D. Hoiem, and R. Sukthankar for providing an audio fingerprinting system to begin our explorations. Their audio-fingerprinting system and their results may be found at: http://www.cs.cmu.edu/ yke/musicretrieval

References

[1] X. Hua, L. Lu, and H. Zhang, "Robust learning-based TV commercial detection," in *Proc. ICME*, 2005, pp. 149–152.

[2] P. Duygulu, M. Chen, and A. Hauptmann, "Comparison and combination of two novel commercail detection methods," in *Proc. ICME*, 2004, pp. 1267–1270.

[3] D. Sadlier, S. Marlow, N. O'Connor, and N. Murphy, "Automatic tv advertisement detection from MPEG bitstream," *J. Pattern Recognition Society*, vol. 35, no. 12, pp. 2–15, 2002.

[4] J. Gauch and A. Shivadas, "Identification of new commercials using repeated video sequence detection," in *Proc. ICIP*, 2005, pp. 1252–1255.

[5] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer vision for music identification," in *Proc. Computer Vision and Pattern Recognition*, 2005.

[6] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, 2002.

[7] C. Jacobs, A. Finkelstein, and D. Salesin, "Fast multiresolution image querying," in *Proc. SIGGRAPH*, 1995.

[8] B. Gold and N. Morgan, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. John Wiley & Sons, Inc., 1999.