Social- and Interactive-Television Applications Based on Real-Time Ambient-Audio Identification

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

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

Abstract
This paper describes *mass personalization*, a framework for combining mass media with a highly personalized Web-based experience. We introduce four applications for mass personalization: personalized content layers, *ad hoc* social communities, real-time popularity ratings and virtual media library services. Using the ambient audio originating from the television, the four applications are available with no more effort than simple television channel surfing. Our audio identification system does not use dedicated interactive TV hardware and does not compromise the user's privacy. Feasibility tests of the proposed applications are provided both with controlled conversational interference and with "living-room" evaluations.

1. Introduction

"Mass media *is the term used to denote, as a class, that section of the media specifically conceived and designed to reach a very large audience… forming a mass society with special characteristics, notably atomization or lack of social connections"*
(en. wikipedia.org).

These characteristics of mass media contrast sharply with the World Wide Web. Mass-media channels typically provide limited content to many people; the Web provides vast amounts of information, most of interest to few. Mass-media channels typically beget passive, largely anonymous, consumption, while the Web provides many interactive opportunities like chatting, emailing and trading. Our goal is to combine the best of both worlds: integrating the relaxing and effortless experience of mass-media content with the interactive and personalized potential of the Web, providing *mass personalization*.

Beyond presenting mass personalization applications, our main technical contribution is in creating a system that does not rely on future hardware or physical connections between TVs and computers. Instead, we introduce a system that can simply 'listen' to ambient audio and connect the viewer with services and related content on the Web. As shown in Figure 1, our system consists of three distinct components: a client-side interface, an audio-database server (with mass-media audio statistics), and a social-application web server. The client-side interface samples and *irreversibly* compresses the viewer's ambient audio to summary statistics. These statistics are streamed from the viewer's personal computer to the audio-database server for identification of the background audio (*e.g.*, 'Seinfeld' episode 6101, minute 3:03). The audio database transmits this information to the social-application server, which provides personalized and interactive content back to the viewer. Continuing with the previous example, if friends of the viewer were watching the same episode of 'Seinfeld' at the same time, the social-application server could automatically create an on-line *ad hoc* community of these "buddies". This community allows members to comment on the broadcast material in real time.

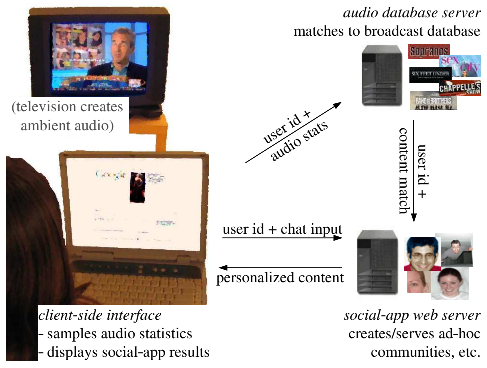
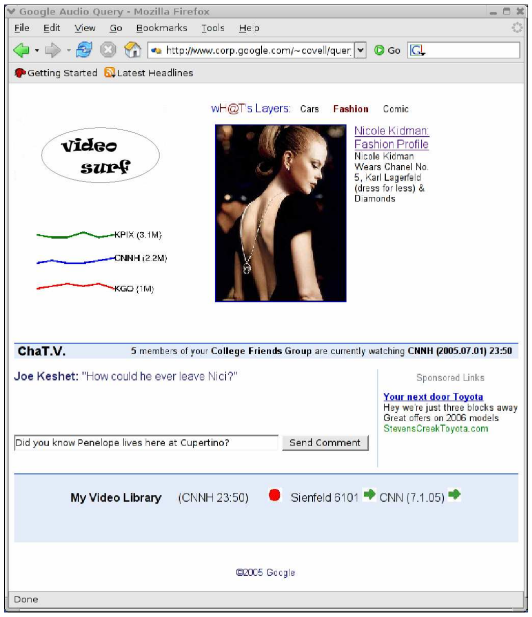

audio database server
matches to broadcast database
(television creates ambient audio)
user id + audio stats
user id + content match
user id + chat input
personalized content
client-side interface
- samples audio statistics
- displays social-app results
social-app web server
creates/serves ad-hoc communities, etc.

Figure 1: Flow chart of the *mass-personalization* applications.


Google Audio Query - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.corp.google.com/~covell/quer
Go
Getting Started
Latest Headlines
video surf
KPIX (3.1M)
CNNH (2.2M)
KGO (1M)
Send Comment
Done

wH@T's Layers: Cars **Fashion** Comic

Nicole Kidman: Fashion Profile
Nicole Kidman Wears Chanel No. 5, Karl Lagerfeld (dress for less) & Diamonds

ChaT.V. 5 members of your **College Friends Group** are currently watching **CNNH (2005.07.01) 23:50**

Joe Keshet: "How could he ever leave Nici?"

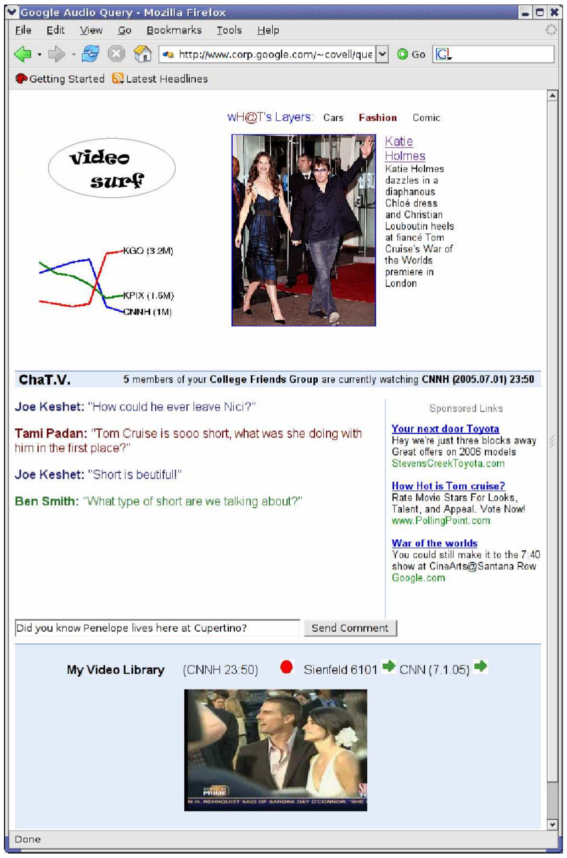
Did you know Penelope lives here at Cupertino?

Sponsored Links

Your next door Toyota
Hey we're just three blocks away
Great offers on 2006 models
StevensCreekToyota.com

My Video Library (CNNH 23:50) ● Sienfeld 6101 CNN (7.1.05)

©2005 Google


Google Audio Query - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.corp.google.com/~covell/que
Go
Getting Started
Latest Headlines
video surf
KGO (3.2M)
KPIX (1.6M)
CNNH (1M)
Send Comment
Done

wH@T's Layers: Cars **Fashion** Comic

Katie Holmes
Katie Holmes dazzles in a diaphanous Chloé dress and Christian Louboutin heels at fiancé Tom Cruise's War of the Worlds premiere in London

ChaT.V. 5 members of your **College Friends Group** are currently watching **CNNH (2005.07.01) 23:50**

Joe Keshet: "How could he ever leave Nici?"

Tami Padan: "Tom Cruise is sooo short, what was she doing with him in the first place?"

Joe Keshet: "Short is beutiful!"

Ben Smith: "What type of short are we talking about?"

Did you know Penelope lives here at Cupertino?

Sponsored Links

Your next door Toyota
Hey we're just three blocks away
Great offers on 2006 models
StevensCreekToyota.com

How Hot is Tom cruise?
Rate Movie Stars For Looks, Talent, and Appeal. Vote Now!
www.PollingPoint.com

War of the worlds
You could still make it to the 7:40 show at CineArts@Santana Row
Google.com

My Video Library (CNNH 23:50) ● Sienfeld 6101 CNN (7.1.05)

Figure 2: A hypothetical interface showing the dynamic output of the mass-personalization applications. Personalized information layers are shown as "wH@T's Layers" (top) and as sponsored links (right-side middle). Ad-hoc chat is shown under "ChaT.V." (left-side middle). Real-time popularity ratings are presented as line graphs (left top) and Video bookmarks are under "My Video Library" (bottom).

The viewer's acoustic privacy is maintained by the irreversibility of the mapping from audio to summary statistics. Unlike the speech-enabled proactive agent by Hong et al. (2001), our approach will not "overhear" conversations. Furthermore, no one receiving (or intercepting) these statistics is able to eavesdrop, on such conversations, since the original audio does not leave the viewer's computer and the summary statistics are insufficient for reconstruction. Further, the system can easily be designed to use an explicit 'mute/un-mute' button, to give the viewer full control of when acoustic statistics are collected for transmission.

Although we apply our techniques to television, we do not use the visual channel as our data source. Instead, we use audio for three pragmatic reasons. First, with visual data, the viewer either must have a TV-tuner card installed in her laptop (which is rare), or must have a camera pointed towards the TV screen (which is cumbersome). In contrast, non-directional microphones are built into most laptops and desktops. Second, audio recording does not require the careful normalization and calibration needed for video sources (camera alignment, image registration, *etc*.). Third, processing audio takes less computation than processing video, due to lower input-data rates. This is especially important since we process the raw data on the client's machine (for privacy reasons), and would like to keep computation requirements at a minimum.

In the next section, we describe four applications, aimed at supplementing television material with personal and social interactions related to the television content. Section 3 describes some of the infrastructure required to deploy these applications. We then describe the core technology needed for ambient-sound matching (Section 4). We provide quantitative measures of the robustness and precision of the audio matching component (Section 5.1) as well as the evaluation of the complete system (Section 5.2). The paper concludes with a discussion on the scope, limitations, and future extensions of this application area.

2. Personalizing Broadcast Content: Four Applications

In this section, we describe four applications to make TV more personalized, interactive and social: personalized information layers, *ad hoc* social peer communities, real-time popularity ratings, and TV-based bookmarks.

2.1 Personalized Information Layers

The first application provides information that is complementary to the mass-media channel (*e.g.*, TV or radio) in an effortless manner. As with proactive software agents (Rhodes *et al*., 2003), we provide additional layers of related information, such as fashion, politics, business, health, or traveling. For example, while watching a news segment on Tom Cruise, a fashion layer might provide information on what designer clothes and accessories the presented celebrities are wearing (see "wH@T's Layers" in Figure 2).

The feasibility of providing the complementary layers of information is related to the cost of annotating the database of mass-media content and the number of times any given piece of content is retransmitted. We evaluated how often content is retransmitted for the ground-truth data used in the evaluations presented in Section 5. We found that up to 1/2 (for CNN Headlines) of the content was retransmitted within 4 days, with higher rates expected for longer time windows.

Thus, if 'Seinfeld' is annotated once, years of reruns would benefit from relevant information layers. Interestingly, a channel like MTV (VH-1), where content is often repeated, has internally introduced the concept of pop-ups that accompany music clips and provide additional entertaining information. The concept of complementary information has passed the feasibility test, at least in the music-video domain.

In textual searches, complementary information providing relevant products and services is often associated via a bidding process (*e.g.*, sponsored links on Web search sites such as Google.com). A similar procedure could be adapted to mass-personalization applications. Thus, content providers or advertisers might bid for specific television segments. For example, local theaters or DVD rental stores might bid on audio from a movie trailer (see "Sponsored Links" in the center right panels of Figure 2).

In many mass-media channels, textual information (closed captioning) accompanies the audio stream. In these cases, the closed captions provide keywords useful for searching for related material. The search results can be combined with a viewer's personal profile and preferences (ZIP code and 'fashion') in order to display a web-page with content automatically obtained from web-pages or advertisement repositories using the extracted keywords. A method for implementing this process was described by Henzinger *et al*. (2003).

In the output of our prototype system, shown in the top right panels of Figure 2, we hand labeled the content indices corresponding to an hour of footage that was taped and replayed. This annotation provided short summaries and associated URLs for the fashion preferences of celebrities appearing on the TV screen during the corresponding 5-second segment. While we did this summarization manually within our experiment, automatic summarization technologies (Kupiec *et al*., 1995) could be used to avoid manual summarization, or bidding techniques described above could be used in a production system to provide related ads.

2.2 Ad-hoc Peer Communities

As evidenced by the popularity of message boards relating to TV shows and current events, people often want to comment on the content of mass-media broadcasts. However, it is difficult to know with whom to chat *during* the actual broadcast. The second application provides another venue for commentary, an *ad hoc* social community.

This *ad hoc* community includes viewers watching the same show on TV. We create this community from the set of viewers whose audio statistics matched the same content in our audio database. These viewers are automatically linked by the social-application server. Thus, a viewer who is watching the latest CNN headlines can chat, comment on, or read other people's responses to the ongoing broadcast. The group members can be further constrained to contain only people in the viewer's social network (i.e. on-line friend community) or to contain established experts on the topic.

Importantly, as the viewer's viewing context changes (by changing channels), the community is automatically changed by re-sampling the ambient audio. The viewer need never indicate what program is being watched; this is particularly helpful for the viewer who changes channels often, and is often not aware of the exact show or channel that is currently being viewed.

This application differs dramatically from the personalized information layers. This service provides a *commenting medium* (chat room, message board, wiki page or video link) where responses of other viewers that are currently watching the same channel can be shared (see "ChaT.V." in the center left panels of Figure 2). Personalized information layers allow only limited interaction by the viewer and are effectively scripted prior to broadcast according to annotations or auction results. In contrast, the content presented by this application is created by ongoing collaborative (or combative) efforts by the viewer and community responses.

As an extension, these chat sessions also have an interesting intersection with Personalized Information Layers. Program-specific chat sessions can be replayed synchronously with the program during reruns of that content, giving the viewer of this later showing access to the comments of previous viewers, *with the correct timing* relative to the program content.

To enable this application, the social-application server simply maintains a list of viewers currently 'listening to' similar audio, with further restrictions as indicated by the viewer's personal preferences. Alternately, these personalized chat rooms can self assemble by matching viewers with shared historical viewing preferences (*e.g.*, daily viewings of 'Star Trek'), as is commonly done in "collaborative filtering" applications (Pennock *et al.* 2000).

2.3 Real-Time Popularity Ratings

Popularity ratings of broadcasting events are of interest to viewers, broadcasters, and advertisers. These needs are partially filled by measurement systems like the Nielsen ratings. However, these ratings require dedicated hardware installation and tedious cooperation from the participating individuals. The third application is aimed at providing ratings information (similar to Nielsen's systems) but with low latency, easy adoption, and for presentation to the viewers as well as the content providers. For example, a viewer can instantaneously be provided with a real time popularity rating of which channels are being watched by her social network or alternatively by people with similar demographics (see ratings graphs in top left panels of Figure 2).

Given the matching system described to this point, the popularity ratings are easily derived by simply maintaining counters on each of the shows being monitored. The counters can be intersected with demographic group data or geographic group data.

Having real-time, fine-grain ratings is more valuable than ratings achieved by the Nielsen system. Real-time ratings can be used by viewers to "see what's hot" while it is still ongoing (for example, by noticing an increased rating during the 2004 super bowl half-time). They can be used by advertisers and content providers to *dynamically* adjust what material is being shown to respond to drops in viewership. This is especially true for ads: the unit length is short and unpopular ads are easily replaced by other versions from the same campaign, in response to viewer rating levels.

2.3 Video "Bookmarks"

Television broadcasters, such as CBS and NBC, are starting to allow content to be (re-)viewed on demand, for a fee, over other channels (*e.g.*, iPoD video downloads or video streaming), allowing viewers to create personalized libraries of their favorite broadcast content (Mann, 2005). The fourth application provides a low-effort way to create these video libraries.

When a viewer sees a segment of interest on TV, she simply presses a button on her client machine, to "bookmark" that point in that broadcast. The current snippet of the ambient audio is recorded, processed and saved. This snippet provides a unique signature into the program being watched. This bookmark can either be used to retrieve the program for later viewing or to mark that specific portion of the program as being of interest. As with other bookmarks, the reference can then be shared with friends or saved for future personal retrieval.

Figure 2 shows an example of the selection interface under "My Video Library" at bottom of the second screen shot. The red "record" button adds the current program episode to her favorites-library. Two video bookmarks are shown as green "play" buttons, with the program name and record date attached.

The program material associated with the bookmarks can be viewed-on-demand through a Web-based streaming application, among other access methods, according to the policies set by the content owner. Depending on these policies, the streaming service can provide free single-viewing playback, collect payments as the agent for the content owners, or insert advertisements that would provide payment to the content owners.

3. Supporting Infrastructure

The four applications described in the previous section share the same client-side and audio-database components and differ only in what information is collected and presented by the social-application server. We describe these common components in this section. We also provide a brief description of how these were implemented in our test setup.

3.1 Client-interface setup

The client-side setup uses a laptop (or desktop) to (1) sample the ambient audio, (2) irreversibly convert short segments of that audio into distinctive and robust summary statistics, and (3) transmit these summary statistics in real-time to the audio database server.

We used a version of the audio-fingerprinting software created by (Ke *et al*., 2005) to provide these conversions. The transmitted audio statistics also include a unique identifier for the client machine to ensure that the correct content-to-client mapping is made by the social-application server. The client software continually records 5-second audio segments and converts each snippet to 415 frames of 32-bit descriptors, according to the method described in Section 4. The descriptors, not the audio itself, are sent to the audio server. By sending only summary statistics, the viewer's acoustic privacy is maintained: the highly compressive many-to-one mapping from audio to statistics is not invertible.

Although a variety of setups are possible, for our experiments, we used an Apple iBook laptop as the client computer and its built-in microphone for sampling the viewer's ambient audio.

3.2 Audio-database server setup

The audio-database server accepts audio statistics (associated with the client id) and compares those received "fingerprints" to its database of recent broadcast media. It then sends the best-match information, along with a match confidence and the client id, to the social-application server.

In order to perform its function, the audio-database server must have access to a database of broadcast audio data. However, the actual audio stream does not need to be stored. Instead, only the compressed representation (32-bit descriptor) is stored. This allows as much as a year of broadcast fingerprints to be stored in less than 1 GB of memory.

The audio database was implemented on a single-processor, 3.4GHz Pentium 4 workstation, with 3 GB of memory. The audio-database server received a query from the viewer each 5 seconds. As will be described in Section 4, each 5-second query was independently matched against the database.

3.3 Social-application server setup

The final component is the social-application server. The social-application server accepts web-browser connections (associated with client computers). Using the content-match results provided by the audio-database server, the social-application server collects personalized content for each viewer and presents that content using an open web browser on the client machine. This personalized content can include the material presented earlier: ads, information layers, popularity information, video "book marking" capabilities, and links to broadcast-related chat rooms and ad-hoc social communities.

For simplicity, in our experiments, the social-application server was set up on the same workstation as the audio-database server. The social-app server receives the viewer/content-index matching information, with the confidence score, from the audio-database server as the audio-database server determines those matches. It maintains client-session-specific state information, such as the current and previous match values and their confidence, the viewer profile (if available), recently presented chat messages (to provide conversational context), and previously viewed content (to avoid repetition). With this information, it dynamically creates web pages for each client session, which include the personalized information derived from the viewer profile (if available) and her audio-match content.

4. Audio Fingerprinting

For our system, the main challenge is accurately matching an audio query to a large database of audio snippets, in real-time and with low latency. High accuracy requires discriminative audio representations that are resistant to the expected distortions introduced by compression, broadcasting and client recording. This paper adapts the music-identification system proposed by (Ke *et al*., 2005) to handle TV audio data and queries. Other audio identification systems are also applicable (*e.g.*, Shazam Entertainment, 2005) but the system by (Ke *et al*., 2005) has the advantage of being compact, efficient, and non-proprietary (allowing reproduction of results).

The audio-identification system starts by decomposing each query snippet (*e.g.*, five-seconds of recorded audio) into overlapping frames spaced roughly 12 ms apart. Each frame is converted into a highly discriminative 32-bit descriptor, specifically trained to overcome typical audio noise and distortion. These identifying statistics are sent to a server, where they are matched to a database of statistics taken from mass-media clips. The returned

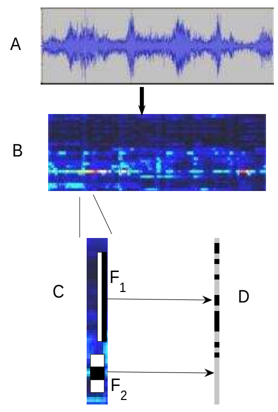


Figure 3: Audio (A) is converted into a spectrogram (B). The spectrogram frames (C) are processed by 32 contrast filters and thresholded to produce a 32-bit descriptor (D). Contrast filters subtract neighboring rectangular spectrogram regions (white regions -black regions), and can be calculated using the integral-image technique.

hits define the candidate list from the database. These candidates are evaluated using a first-order hidden Markov model, which provides high scores to candidate sequences that are temporally consistent with the query snippet. If the consistency score is sufficiently high, the database snippet is returned as a match. The next two subsections provide a description of the main components of the method.

4.1 Hashing Descriptors

Ke *et al*. (2005) used a powerful machine learning technique, called *boosting*, to find highly discriminative, compact statistics for audio. Their procedure trained on labeled pairs of positive examples (where q and x are noisy versions of the same audio) and negative examples (q and x are from different audio). During this training phase, boosting uses the labeled pairs to select a combination of 32 filters and thresholds that jointly create a highly discriminative statistic. The filters localize changes in the spectrogram magnitude, using first- and second-order differences across time and frequency (see Figure 3). One benefit of using these simple difference filters is that they can be calculated efficiently using the integral image technique suggested by (Viola and Jones, 2002).

The outputs of these filters are thresholded, giving *a single bit per filter* at each audio frame. These 32 threshold results form the only transmitted description of that frame of audio. This sparseness in encoding ensures the privacy of the viewer to unauthorized eavesdropping. Further, these 32-bit output statistics are robust to the audio distortions in the training data, so that positive examples (matching frames) have small hamming distances (distance measuring differing number of bits) and negative examples (mismatched frames) have large hamming distances.

The 32-bit descriptor itself is used as a hash key for direct hashing. The boosting procedure generates a descriptor that is itself a well-balanced hash function. Retrieval rates are further improved by querying not only the query descriptor itself, but also a small set of similar descriptors (up to a hamming distance of 2).

4.2 Within-query consistency

Once the query frames are individually matched to the audio database, using the hashing procedure, the potential matches are validated. Simply counting the number of frame matches is inadequate, since a database snippet might have many frames matched to the query snippet but with completely wrong temporal structure.

To insure temporal consistency, each hit is viewed as support for a match at a specific query-to-database offset. For example, if the eighth descriptor (q_8) in the 5-second, 415-frame-long 'Seinfeld' query snippet, q, hits the 1008th database descriptor (x_{1008}), this supports a candidate match between the 5-second query and frames 1001 through 1415 in the database. Other matches mapping q_n to x_{1000+n} ($1 \leq n \leq 415$) would support this same candidate match.

In addition to temporal consistency, we need to account for frames when conversations temporarily drown out the ambient audio. We model this interference as an exclusive switch between ambient audio and interfering sounds. For each query frame i, there is a hidden variable, y_i: if $y_i = 0$, the *i-th* frame of the query is modeled as interference only; if $y_i = 1$, the *i-th* frame is modeled as from clean ambient audio. Taking this extreme view (pure ambient or pure interference) is justified by the extremely low precision with which each audio frame is represented (32 bits) and is softened by providing additional bit-flip probabilities for each of the 32 positions of the frame vector under each of the two hypotheses ($y_i = 0$ and $y_i = 1$). Finally, we model the between-frame transitions between ambient-only and interference-only states as a hidden first-order Markov process, with transition probabilities derived from training data. We re-used the 66-parameter probability model given by (Ke *et al*., 2005).

Our final model of the match probability between a query vector, q, and an ambient-database vector at an offset of N frames, x^N, is:

$$P(q|x^N) = \prod_{n=1}^{415} P(\langle q_n, x_{N+n} \rangle \mid y_n)\, P(y_n|y_{n-1}),$$

where $<q_n, x_m>$ denotes the bit differences between the two 32-bit frame vectors q_n and x_m. This model incorporates both the temporal consistency constraint and the ambient/interference hidden Markov model.

4.3 Post-match consistency filtering

People often talk with others while watching television, resulting in sporadic but strong acoustic interference, especially when using laptop-based microphones for sampling the ambient audio. Given that most conversational utterances are two to three seconds in duration (Buttery and Korhonen, 2005), a simple exchange might render a 5-second query unrecognizable.

To handle these intermittent low-confidence mismatches, we use post-match filtering. We use a continuous-time hidden Markov model of channel switching with an expected dwell time (i.e. time between channel changes) of L seconds. The social-application server indicates the highest-confidence match within the recent past (along with its "discounted" confidence) as part of the state information associated with each client session. Using this information, the server selects either the content-index match from the recent past or the current index match, based on whichever has the higher confidence.

We use $\boldsymbol{M}_\text{h}$ and $\boldsymbol{C}_\text{h}$ to refer to the best match for the previous time step (5 seconds ago) and its log-likelihood confidence score. If we simply apply the Markov model to this previous best match, without taking another observation, then our *expectation* is that the best match for the current time is that same program sequence, just 5 seconds further along, and our confidence in this expectation is $\boldsymbol{C}_\text{h} - l/L$ where $l = 5$ seconds is the query time step. This discount of l/L in the log likelihood corresponds to the Markov model probability, $e^{-l/L}$, of not switching channels during the l-length time step.

An alternative hypothesis is generated by the audio match for the *current* query. We use M_0 to refer to the best match for the current audio snippet: that is, the match that is generated by the audio fingerprinting software. C_0 is the log-likelihood confidence score given by the audio fingerprinting software.

If these two matches (the updated historical expectation and the current snippet observation) give different matches, we select the hypothesis with the higher confidence score:

$$\{M_0, C_0\} = \begin{cases} \{M_h,\ C_h\text{-}l/L\} & \text{if } C_h\text{-}l/L > C_0 \\ \{M_0,\ C_0\} & \text{otherwise} \end{cases}$$

where $\boldsymbol{M}_0$ is the match that is used by the social-application server for selecting related content and $\boldsymbol{M}_0$ and $\boldsymbol{C}_0$ are carried forward on the next time step as $\boldsymbol{M}_\text{h}$ and $\boldsymbol{C}_\text{h}$.

5. Evaluation of System Performance

In this section, we provide a quantitative evaluation of our ambient-audio identification system. The first set of experiments provides in-depth results with our matching system. The second set of results provides an overview of the performance of an integrated system running in a live environment.

5.1 Empirical Evaluation

Here, we examine the performance of our audio-matching system in detail. 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 to provide disjoint query/database sets: 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. The marked material also included repeated programs (487 minutes worth), 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). We also marked as repeats those segments within a single program (*e.g.*, the movie "Treasure Island") where the only sounds were theme music and the repetitions were indistinguishable to a human listener, even if the visual track was distinct. This typically occurred during the start and end credits of movies or series programs and during news programs which replayed sound bites with different graphics.

We did *not* label as repeats: similar sounding music that occurred in different programs (*e.g.*, the suspense music during "Harry Potter" and random soap operas) or silence periods (*e.g.*, between segments, within some suspenseful scenes).

Table 1 shows our results from this experiment, under "clean" acoustic conditions, using 5-second and 10-second query snippets. Under these "clean" conditions, we jack-knifed the captured broadcast audio without added interference. We found that

Table 1: Performance results on 4 days of 5-second and 10-second queries operating against 4 days of mass media. False-positive rate = FP/(TN+FP); False-negative rate = FN/(TP+FN); Precision = TP/(TP+FP); Recall = TP/(TP+FN).

	Query quality / length			
	clean		noisy	
	5 sec	10 sec	5 sec	10 sec
False-pos. rate	6.4%	4.7%	1.1%	2.7%
False-neg. rate	6.3%	6.0%	83%	10%
Precision	87%	90%	88%	94%
Recall	94%	94%	17%	90%

most of the false positive results on the 5-second snippets were during silence periods, during suspense-setting music (which tended to have sustained minor cords and little other structure).

To examine the performance under noisy conditions, we compare these results to those obtained from audio that includes a competing conversation. We used a 4.5-second dialog, taken from Kaplan's TOEFL material (Rymniak, 1997)[1]. We scaled this dialog and mixed it into each query snippet. This resulted in 1/2 and 5 ½ seconds of each 5- and 10-second query being *uncorrupted* by competing noise. The perceived sound level of the interference was roughly matched to that of the broadcast audio, giving an interference-peak-amplitude four times larger than the peak amplitude of the broadcast audio, due to the richer acoustic structure of the broadcast audio.

The results reported in Table 1 under "noisy" show similar performance levels to those observed in our experiments reported in *subsection 5.2*. The improvement in precision (that is, the drop in false positive rate from that seen under "clean" conditions) is a result of the interfering sounds preventing incorrect matches between silent portions of the broadcast audio.

Due to the manner in which we constructed these examples, longer query lengths correspond to more sporadic discussion, since the competing discussion is active about half the time, with short bursts corresponding to each conversational exchange. It is this type of sporadic discussion that we actually observed in our "in-living-room" experiments (described in the next section). Using these longer query lengths, our recall rate returns to near the rate seen for the interference-free version.

[1] The dialog was: *(woman's voice)* "Do you think I could borrow ten dollars until Thursday?", *(man's voice)* "Why not, it's no big deal.".

5.2 "In-Living-Room" Experiments

Television viewing generally occurs in one of three distinct physical configurations: remote viewing, solo seated viewing, and partnered seated viewing. We used the system described in Section 3 in a complete end-to-end matching system within a "real" living-space environment, using a partnered seated configuration. We chose this configuration since it is the most challenging, acoustically.

Remote viewing generally occurs from a distance (*e.g.*, from the other side of a kitchen counter), while completing other tasks. In this cases, we expect the ambient audio to be sampled by a desktop computer placed somewhere in the same room as the television. The viewer is away from the microphone, making the noise she generates less problematic for the audio identification system. She is distracted (*e.g.*, by preparing dinner), making errors in matching less problematic. Finally, she is less likely to be actively channel surfing, making historical matches more likely to be valid.

In contrast with remote viewing, during seated viewing, we expect the ambient audio to be sampled by an laptop, held in the viewer's lap. Further, during partnered, seated viewing, the viewer is likely to talk with her viewing partner, very close to the sampling microphone. Nearby, structured interference (*e.g.*, voices) is more difficult to overcome than remote spectrally flat interference (*e.g.*, oven-fan noise). This makes the partnered seated viewing, with sampling done by laptop, the most acoustically challenging and, therefore, the configuration that we chose for our tests.

To allow repeated testing of the system, we recorded approximately one hour of broadcast footage onto VHS tape prior to running the experiment. This tape was then replayed and the resulting ambient audio was sampled by a client machine (the Apple iBook laptop mentioned in *subsection 3.12*).

The processed data was then sent to our audio server for matching. For the test described in this section, the audio-server was loaded with the descriptors from 24 hours of broadcast footage, including the one hour recorded to VCR tape. With this size audio database, the matching of each 5-second query snippet took consistently less than 1/4 second, even without statistical sampling (*e.g.*, the RANSAC method suggested by Fischler and Bolles, 1981).

Surf Dwell Time (sec)	Incorrect labels
1.25	0 %
1.00	22 %
0.75	22 %
0.50	14 %
0.25	18 %

Table 2. Match results on 30 minutes of in-living room data after filtering using the channel surfing model. The incorrect label rate before filtering was 80%.

During this experiment, the laptop was held on the lap of one of the viewers. We ran five tests of five minutes each, one for each of 2-foot increase in distance from the television set, from two- to ten-feet. During these tests, the viewer holding the iBook laptop and a nearby viewer conversed sporadically. In all cases, these conversations started 1/2 to 1 minute after the start of the test. The laptop-television distance and the sporadic conversation resulted in recordings with acoustic interference louder than the television audio whenever either viewer spoke.

The interference created by the competing conversation, resulted in incorrect best matches with low confidence scores for up to 80% of the matches, depending on the conversational pattern. However, we avoided presenting the unrelated content that would have been selected by these random associations, by using the simple model of channel watching/surfing behavior described in *subsection 4.2* with an expected dwell time (time between channel changes) of 2 seconds. This consistent improvement was due to correct and strong matches, made before the start of the conversation: these matches correctly carried forward through the remainder of the 5 minute experiment. No incorrect information or chat associations were visible to the viewer: our presentation was 100% correct.

We informally compared the viewer experience using the post-match filtering corresponding to the channel-surfing model to that of longer (10-second) query lengths, which did not require the post-match filtering. The channel-surfing model gave the more consistent performance, avoiding the occasional "flashing" between contexts that was sometimes seen with the unfiltered, longer-query lengths.

To further test the post-match surfing model, we took a single recording of 30 minutes at a distance of 8 feet, using the same physical and conversational set-up as described above. On this experiment, 80% of the direct matching scores were incorrect, prior to post-match filtering. Table 2 shows the results of varying the expected dwell time within the channel surfing model on this data. The results are non-monotonic in the dwell time due to the non-linearity in the filtering process: for example, between L=1.0 and 0.75, an incorrect match overshadows a later, weaker correct match, making for a long incorrect run of labels but, at L=0.5, the range of influence of that incorrect match is reduced and the later, weaker correct match shortens the incorrect run length.

Post-match filtering introduces one to five seconds of latency in the reaction time to channel changes during casual conversation. However, the effects of this latency are usually mitigated because a viewer's attention typically is not directed at the web-server-provided information during channel changes; rather, it is typically focused on the newly selected TV channel, making these delays largely transparent to the viewer.

These experiments validate the use of the audio fingerprinting method developed by (Ke *et al.*, 2005) for audio associated with television. The precision levels are lower than for the music retrieval application that they have described since broadcast television is not providing the type of distinctive sound experience that most music strives for. Nevertheless, the recall characteristic is sufficient for using this method in a living room environment.

6. Discussion

The proposed applications rely on personalizing the mass-media experience by matching ambient-audio statistics. The applications provide the viewer with personalized layers of information, new avenues for social interaction, real time indications on show popularity and the ability to maintain a library of the favorite content through a virtual recording service. These personalization applications can be modified in order to provide the degree of privacy each viewer feels comfortable with. Similarly, the applications can vary according to viewer-specific technical constraints, such as bandwidth and CPU time.

The paper emphasizes two contributions. The first is that audio fingerprinting can provide a feasible method for identifying which mass-media content is experienced by viewers. Several audio fingerprinting techniques might be used for achieving this goal. The proposed framework adapted the system proposed by (Ke *et al.*, 2005) due to its efficiency and accessibility. Once the link between the viewer and the mass-media content is made, the second contribution follows, by completing the mass media experience with personalized Web content and communities. These two contributions work jointly in providing both simplicity and personalization in the proposed applications.

The proposed applications were described using a setup of ambient audio originating from a TV and encoded by a nearby personal computer. As computational capacities proliferate to portable appliances, like cell phones and PDAs, the fingerprinting process could naturally be carried out on such platforms. For example, SMS responses of a cell phone based community watching the same show can be one such implementation. In addition, the mass-media content can originate from other sources like radio, movies or in scenarios where viewers share a location with a common auditory background (*e.g.*, an airport terminal, party, or music concert).

References

Buttery, P. & Korhonen, A. (2005). Large-scale analysis of verb subcategorization differences between child directed speech and adult speech. *In Proceedings of the Workshop on Identification and Representation of Verb Features and Verb Classes.*

Fischler, M. & Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381-395.

Henzinger, M., Chang, B., Milch, B., & Brin, S. (2003). Query-free news search. In *Proceedings of the International WWW Conference*.

Hong, J. & Landay (2001). J. A context/communication information agent. *Personal and Ubiquitous Computing*. 5(1):78-81.

Ke, Y., Hoiem, D., & Sukthankar, R. (2005). Computer vision for music identification. In *Proceedings of Computer Vision and Pattern Recognition*.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. In *Proceedings of. ACM SIG Information Retrieval*, pages 68-73.

Mann, J. (2005). CBS, NBC to offer replay episodes for 99 cents. *www.techspot.com/news/.*

Pennock, D., Horvitz, E., Lawrence, S., & Giles, C. L. (2000). Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In *Proceedings of Uncertainty in Artificial Intelligence*, pages 473-480.

Rhodes, B. & Maes, P. (2003). Just-in-time information retrieval agents. IBM *Systems Journal*, 39(4):685-704.

Rymniak, M. (1997). The essential review: Test of English as a foreign language. *Kaplan Educational Centers*.

Shazam Entertainment, Inc. (2005). www.shazamentertainment.com/.

Viola, P. & Jones, M. (2002). Robust real-time object detection. *International Journal of Computer Vision*.