

# Mass personalization: social and interactive applications using sound-track identification

Michael Fink · Michele Covell · Shumeet Baluja

Published online: 21 December 2006  
© Springer Science + Business Media, LLC 2006

**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 a 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.

**Keywords** Mass personalization · Mass media · Real-time · Television · Audio-finger printing

## 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).*

---

M. Fink (✉)  
Center for Neural Computation, The Hebrew University of Jerusalem, Jerusalem 91904, Israel  
e-mail: fink@cs.huji.ac.il

M. Covell · S. Baluja  
Google Research, Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA

M. Covell  
e-mail: covell@google.com

S. Baluja  
e-mail: shumeet@google.com

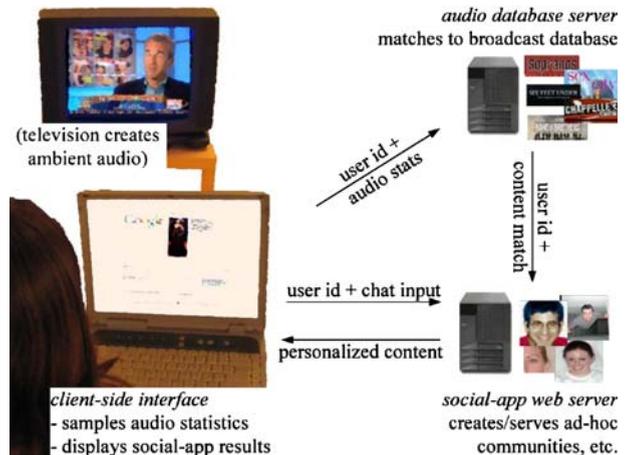
These characteristics of mass media contrast sharply with the World Wide Web. Mass-media channels typically provide limited content to many people; the Web provides vast amounts of information, most of interest to few. Mass-media channels are typically consumed in a largely anonymous, passive manner, 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*.

Upon request, our system tests whether the user’s ambient audio matches a known mass-media channel. If a match is found the user is provided with services and related content originating from the Web. As shown in Fig. 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 6,101, 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.

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 must either have a TV-tuner card installed in her laptop (which is rare), or have a camera pointed towards the TV screen (which is cumbersome). In contrast, non-directional microphones are built into most laptops and shipped with most desktops. Second, audio recording does not require the careful normalization and calibration needed for video sources (camera alignment, image registration, *etc.*). Third, processing audio is less computational demanding 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 attempt 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

**Fig. 1** Flow chart of the mass-personalization applications



of the infrastructure required to deploy these applications. The core technology needed for ambient-sound matching is described in 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 television more personalized, interactive and social. These applications are: 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., television or radio) in an effortless manner. As with proactive software agents [11], 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 Figs. 2 and 3).

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

**Fig. 2** An interface showing the dynamic output of the mass-personalization applications during a CNN documentary on actor Tom Cruise. Personalized information layers are shown as “wH@T’s h@T 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 material that is presented to the viewer is driven both by the broadcast program to which she is listening (as determined by the ambient audio identification) and by an interest profile she might provide

The screenshot shows a Mozilla Firefox browser window displaying a web page titled "wH@T's Layers". The page features several interactive elements:

- Top Left:** A "video surf" logo.
- Top Middle:** A line graph showing real-time popularity ratings for three channels: KPIX (3.1M), CNNH (2.2M), and KGO (1M).
- Top Right:** A "wH@T's Layers" section with a "Fashion" tab selected, displaying a photo of Nicole Kidman and a "Fashion Profile" listing her attire: "Nicole Kidman Wears Chanel No 5, Karl Lagerfeld (dress for less) & Diamonds".
- Middle Left:** A "ChaT.V." chat section indicating "5 members of your College Friends Group are currently watching CNNH (2005.07.01) 23:50". A chat message from "Joe Keshet" says: "How could he ever leave Nic!?"
- Middle Right:** A "Sponsored Links" section for Toyota, with the text: "Your next door Toyota. Hey we're just three blocks away. Great offers on 2006 models. StevensCreekToyota.com".
- Bottom:** A "My Video Library" section with video bookmarks for "CNNH 23:50", "Stienfeld 6101", and "CNN (7.1.05)".

**Fig. 3** As the documentary progresses, the information on the dynamic webpage is automatically updated. The Personalized information layers and the sponsored links now show new custom tailored content. Additional chat comments are delivered in real time and the current show's popularity ratings are continuously revised

The screenshot shows a Mozilla Firefox browser window with the address bar displaying 'http://www.corp.google.com/~covei/que'. The page content includes a 'video surf' logo, a line graph with three lines (red, green, blue) and data points (3.58), (1.54), and (1.14). A photo of Katie Holmes is displayed. Below the photo is a chat window with messages from Joe Keshet and Tami Padan. To the right of the chat are sponsored links for Toyota and a poll. At the bottom, there is a 'My Video Library' section with a video player showing a scene from Seinfeld.

piece of content is retransmitted. We evaluated how often content is retransmitted for the ground-truth data used in Section 5 and found that up to 50% (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 Figs. 2 and 3).

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. [5].

In the output of our prototype system, shown in the top right panels of Figs. 2 and 3, 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 each 5-s segment. While we did this summarization manually within our experiment, automatic summarization technologies [8] could be used to facilitate this process and bidding techniques 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 a real time 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 television context changes (by changing channels) a re-sampling of the ambient audio can guarantee that the relevant community is automatically updated as well. The viewer need never indicate what program is being watched; this is particularly helpful for the viewer who changes channels often, and is not aware of which show or channel she is currently watching.

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 Figs. 2 and 3). While the Personalized Information Layers allow only limited response from the viewer and are effectively scripted prior to broadcast according to annotations or auction results, the content presented within the peer community application is created by ongoing collaborative (or combative) 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. Additionally, these chat sessions are automatically synchronized even when the viewer pauses, rewinds or fast-forwards the program.

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 [10].

### 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 Figs. 2 and 3).

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 often 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 [3].

### 2.4 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 [9]. 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 program identity and temporal index into that content can be saved using Synchronized Multimedia Integration Language (SMIL) [1]. Using the SMIL format has the advantage of being standardized across many platforms, allowing for freer distribution without compromising the rights of the copyright owners.

The program identity can be indicated using a pointer to the redistribution service the content owner requests. Alternatively, due to its compact representation, the actual identifying fingerprint could be written directly into the SMIL file. This SMIL-encoded bookmark can either be used to retrieve the entire program, a segment of the program as determined by the content owner (for example from the beginning of a scene or news-story), or simply the exact portion of the program from which the fingerprint was created. As with other bookmarks, the reference can then easily be shared with friends or saved for future personal retrieval.

Figure 2 shows an example of the selection interface under “My Video Library” at the 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.

As an added benefit, the start time for the bookmarked data can be adjusted to allow for the delay inherent in the viewer's reaction time. If the “record” button was pressed a minute

or so after the start of a new segment, the content owner can provide index for the start of the segment as the “in” edit point saved in the SMIL file, with the exact time at which the viewer pressed the record button being saved as the first “seek point” (in case that really was the time she wanted).

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. [7] 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-s 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, since 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 representations (32-bit descriptors) are saved. This allows as much as a year of broadcast fingerprints to be stored in 11 GB of memory, uncompressed, or (due to the redundant temporal structure) in about 1 GB of memory with compression.

The audio database was implemented on a single-processor, 3.4 GHz Pentium 4 workstation, with 3 GB of memory. The audio-database server received a query from the viewer every 5 s. As will be described in Section 4, each 5-s 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-application server receives the viewer/content-index matching information, with the confidence score, originating from the audio-database server. The social-application server also maintains client-session-specific state information, such as:

1. The current and previous match values and their confidence
2. A viewer preference profile (if available)
3. Recently presented chat messages (to provide conversational context)
4. 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 and her audio-match content.

## 4 Audio fingerprinting

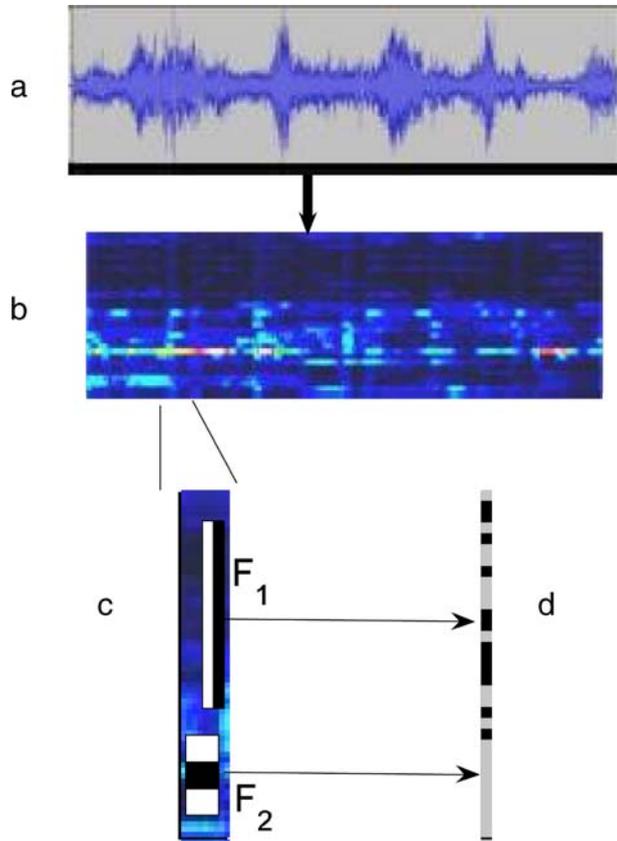
The main challenge for our system is to accurately match an audio query to a large database of audio snippets, in real-time and with low latency. High accuracy requires discriminative audio representations that are resistant to the distortions introduced by compression, broadcasting and client recording. This paper adapts the music-identification system proposed by Ke et al. [7] to handle TV audio data and queries. Other audio identification systems are also applicable (e.g., [13]) but the system by Ke et al. [7] 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., 5-s 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 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. [7] 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

**Fig. 4** 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 (*whiteregions* – *blackregions*), and can be calculated using the integral-image technique. Also see Ke et al.: <http://www.cs.cmu.edu/~yke/musicretrieval>



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 Fig. 4). 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 [14].

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.

The 32-bit descriptor itself is then used as a key for direct hashing. The boosting procedure generates a descriptor that is 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 two). Since the 32-bit output statistics are robust to the audio distortions in the training data, positive examples (matching frames) have small hamming distances (distance measuring differing number of bits) and negative examples (mismatched frames) have large hamming distances.

#### 4.2 Within-query consistency

Once the query frames are individually matched to the audio database, using the efficient 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-s, 415-frame-long ‘Seinfeld’ query snippet,  $q$ , hits the 1,008th database descriptor ( $x_{1,008}$ ), this supports a candidate match between the 5-s query and frames 1,001 through 1,415 in the database. Other matches mapping  $q_n$  to  $x_{1,000+n}$  ( $1 \leq n \leq 415$ ) 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 use the model of interference from [7]: that is, 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, the frame transitions between ambient-only and interference-only states are treated 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. [7].

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

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

where  $\langle q_n, x_m \rangle$  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 yet strong acoustic interference, especially when using laptop-based microphones for sampling the ambient audio. Given that most conversational utterances are 2–3 s in duration [2], a simple exchange might render a 5-s 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  $M_h$  and  $C_h$  to refer to the best match for the previous time step (5 s ago) and its respective 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 s further along, and our confidence in this expectation is  $C_h - l/L$  where  $l=5$  s 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 process.

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:

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

where  $M_0$  is the match that is used by the social-application server for selecting related content and  $M_0$  and  $C_0$  are carried forward on to the next time step as  $M_h$  and  $C_h$ .

## 5 Evaluation of system performance

In this section, we provide a quantitative evaluation of the 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 3 days of one broadcast station and 1 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 1 min) of data.

We hand labeled the 4 days of video, marking the repeated material. This included most advertisements (1,348 min 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 min 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- and 10-s query snippets. Under these “clean” conditions, we jack-knifed the captured broadcast audio without added interference. We found that most of the false positive results on the 5-s snippets were during silence periods, during suspense-setting music (which tended to have sustained minor chords 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-s dialog, taken from

**Table 1** Performance results of 5- and 10-s queries operating against 4 days of mass media

	Query quality/length			
	Clean		Noisy	
	5 s	10 s	5 s	10 s
False-positive rate	6.4%	4.7%	1.1%	2.7%
False-negative rate	6.3%	6.0%	83%	10%
Precision	87%	90%	88%	94%
Recall	94%	94%	17%	90%

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

Kaplan's TOEFL material [12].<sup>1</sup> We scaled this dialog and mixed it into each query snippet. This resulted in 1/2 and 51/2 s of each 5- and 10-s 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.

## 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 these 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.

<sup>1</sup> 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."

In contrast with remote viewing, during seated viewing, we expect the ambient audio to be sampled by a 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 1 h 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 3.1).

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 h of broadcast footage, including the 1 h recorded to VCR tape. With this size audio database, the matching of each 5-s query snippet took consistently less than 1/4 s, even without the RANSAC sampling [4] used by Ke et al. [7].

During this experiment, the laptop was held on the lap of one of the viewers. We ran five tests of 5 min each, one for each of 2-foot increase in distance from the television set, from 2- to 10-feet. During these tests, the viewer holding the iBook laptop and a nearby viewer conversed sporadically. In all cases, these conversations started 1/2–1 min 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 s. 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 min 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-s) 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 min at a distance of 8 ft, 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  $L=0.75$ , 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.

These very low values for the expected dwell times were possible in part because of the energy distribution within conversational speech. Most conversations include lulls and these lulls are naturally lengthened when the conversation is driven by an external

**Table 2** Match results on 30 min of in-living room data after filtering using the channel surfing model

Surf dwell time (s)	Correct labels
1.25	100%
1.00	78%
0.75	78%
0.50	86%
0.25	88%

The correct label rate before filtering was only 20%

presentation (such as the broadcast itself or the related material that is being presented on the laptop). Furthermore, in English, the overall energy envelope is significantly lower at the end of simple statements than at the start and, English vowel–consonant structure gives an additional drop in energy about 4 times per second. These effects result in clean audio about once each 1/4 s (due to syllable structure) and mostly clean audio capture about once per minute (due to sentence-induced energy variations). Finally, we saw very clean audio with longer durations but less predictable, typically during the distinctive portions of the broadcast audio presentation (due to conversational lulls while attending to the presentation). Conversations during silent or otherwise non-distinctive portions of the broadcast actually help our matching performance by partially randomizing the incorrect matches that we would otherwise have seen.

Post-match filtering introduces 1–5 s 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. [7] for audio associated with television. The precision levels are lower than in 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 channel surfing model ensures that 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 applications are provided while addressing five factors, we believe are imperative to any mass personalization endeavor:

1. Guaranteed privacy
2. Minimized installation barriers
3. Integrity of mass media content
4. Accessibility of personalized content
5. Relevance of personalized content

We now discuss how these five factors are addressed within our mass-personalization framework.

The viewer's privacy must be guaranteed. We meet this challenge in the acoustic domain by our irreversible mapping from audio to summary statistics. No one receiving (or intercepting) these statistics is able to eavesdrop on background conversations, since the original audio never leaves the viewer's computer and the summary statistics are insufficient for reconstruction. Thus, unlike the speech-enabled proactive agent by [6], our approach cannot "overhear" conversations. Furthermore, the system can be used in a non-continuous mode such that the user must explicitly indicate (through a button press) that they wish a recording of the ambient sounds. Finally, even in the continuous-case, an explicit 'mute' button provides the viewer with the degree of privacy she feels comfortable with.

Another level of privacy concerns surround the collection of "traces" of what each individual watches on television. As with web browsing caches, the viewer can obviate these concerns in different ways: first and foremost by simply not turning on logging; by explicitly purging the cache of what program material the viewer has watched (so that the past record of her broadcast-viewing behavior is no longer available in either server or client history); by watching program material without starting the mass-personalization application (so that no record is ever made of this portion of her broadcast-viewing behavior); by "muting" the transmission of audio statistics (so that the application simply uses her previously known broadcast station to predict what she is watching).

The second factor is the minimization of installation barriers, both in terms of simplicity and proliferation of installation. Many of the interactive television systems that have been proposed in the past, relied on dedicated hardware and on the accessibility to broadcast-side information (like a teletext stream). However, except for the limited interactive scope of pay-per-view applications, these systems have not achieved significant penetration rates. Even if the penetration of teletext-enabled personal video recorders (PVRs) increases, it is unlikely to equal the penetration levels of laptop computers in the near future. Our system takes advantage of the increasing prevalence of personal computers equipped with standard microphone units. By doing so, our proposed system circumvents the need for installing dedicated hardware and the need to rely on a side information channel. The proposed framework relies on the accessibility and simplicity of a standard software installation.

The third factor in successful personalization of mass-media content is maintaining the integrity of the broadcast content. This factor emerges both from viewers who are concerned about disturbing their viewing experience and from content owners who are concerned about modified presentations of their copyrighted material. For example, in a previously published attempt to associate interactive quizzes and contests with movie content, the copyright owners prevented them from superimposing these quizzes on the television screen during the movie broadcast. Instead, the cable company had to leave a gap of at least 5 min between their interactive quizzes and the movie presentation [15]. Our proposed application presents the viewer with personalized information through a separate screen, such as a laptop or handheld device. This independence guarantees the integrity of the mass media channel. It also allows the viewer to experience the original broadcast without modification, if so desired, by simply ignoring the laptop screen.

Maintaining the simplicity of accessing the mass personalization content is the fourth challenge. The proposed system continuously caches information that is likely to be considered relevant by the user. However, this constant stream is passively stored and not imposed on the viewer in any way. The system is designed so that the personalized material

can be examined by the viewer in her own pace or alternatively, to simply store the personalized material for later reference.

Finally, the most important factor is the relevance of the personalized content. We believe that the proposed four applications demonstrate some of the potential of personalizing the mass-media experience. Our system allows content producers to provide augmented experiences, a non-interactive part for the main broadcast screen (the traditional television, in our descriptions) and an interactive or personalized part for the secondary screen. Our system potentially provides a broad range of information to the viewer, in much the same flavor as the text-based web search results. By allowing other voices to be heard, mass personalization can have increased relevance and informational as well as entertainment value to the end user. Like the web, it can broaden access to communities that are otherwise poorly addressed by most distribution channels. By associating with a mass-media broadcast, it can leverage popular content to raise the awareness of a broad cross section of the population to some of these alternative views.

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. 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 set and encoded by a nearby personal computer. However, 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, lecture, or music concert). In addition, 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. Thus, it seems that the full potential of mass-personalization will gradually unravel itself in the coming years.

**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. Bulterman DCA (2001) SMIL 2.0: overview, concepts, and structure. *IEEE Multimed* 8(4):82–88
2. 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*
3. Covell M, Baluja S, Fink M (2006) Advertisement replacement using acoustic and visual repetition. In: *Proceedings of IEEE multimedia signal processing*
4. Fischler M, Bolles R (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun ACM* 24(6):381–395
5. Henzinger M, Chang B, Milch B, Brin S (2003) Query-free news search. In: *Proceedings of the international WWW conference*.
6. Hong J, Landay J (2001) A context/communication information agent. *Personal and Ubiquitous Computing* 5(1):78–81
7. Ke Y, Hoiem D, Sukthankar R (2005) Computer vision for music identification. In: *Proceedings of computer vision and pattern recognition*

8. Kupiec J, Pedersen J, Chen F (1995) A trainable document summarizer. In: Proceedings of ACM SIG information retrieval, pp 68–73
9. Mann J (2005) CBS, NBC to offer replay episodes for 99 cents. <http://www.techspot.com/news/>
10. Pennock D, Horvitz E, Lawrence S, Giles CL (2000) Collaborative filtering by personality diagnosis: a hybrid memory- and model-based approach. In: Proceedings of uncertainty in artificial intelligence, pp 473–480
11. Rhodes B, Maes P (2003) Just-in-time information retrieval agents. IBM Syst J 39(4):685–704
12. Rymniak M (1997) The essential review: test of English as a foreign language. Kaplan Educational Centers, New York
13. Shazam Entertainment, Inc. (2005) <http://www.shazamentertainment.com/>
14. Viola P, Jones M (2002) Robust real-time object detection. Int J Comput Vis
15. Xinaris T, Kolouas A (2006) PVR: one more step from passive viewing. Euro ITV (invited presentation)



**Michael Fink** is a student in the Center for Neural Computation at the Hebrew University of Jerusalem. His doctoral research with Prof. Shimon Ullman focuses on settings that require learning new categories from few training examples. Humans seem to acquire a novel class from just a single example by utilizing representations extracted from previous classification tasks. Michael's research examines how similar transfer of representational knowledge can be implemented in a machine learning paradigm.



**Michele Covell** received her BS in electrical engineering from the University of Michigan and her MS and PhD from MIT in signal processing. She worked at SRI International from 1989-1993, in research on active (acoustic) noise control and at Interval Research Corporation from 1993-2000 in research on audio, image, and video processing. From 2000 to 2002, Michele worked on faster-than-real-time video analysis at YesVideo, Inc. She then returned to research, in streaming video and services in 3G telephony networks, at the Mobile Streaming Media group at HP Labs. Michele moved to Google, in the research group, in 2005. She has publications and patents in image-based graphics, speech-oriented audio processing, image analysis, video analysis, web-based video services, control-system modeling, and server resource modeling.



**Shumeet Baluja, Ph.D.**, is currently a Senior Staff Research Scientist at Google, Inc, where he works on a broad set of topics ranging from image processing and machine learning to wireless application development and user interaction measurement. Shumeet was formerly the Chief Technology Officer of JAMDAT Mobile, Inc., where he oversaw all aspects of technology initiation, development and deployment. Previously, Shumeet was Chief Scientist at Lycos Inc., where he led the Research and Applied Technology Group in the quantitative and qualitative analysis of user behavior, including data mining and trend analysis, advertisement optimization, and statistical modeling of traffic and site interactions. As Senior Vice President of R&D at eCompanies LLC, he spearheaded the creation of their wireless practice and was responsible for finding and evaluating new technologies and technology entrepreneurs. Shumeet has filed numerous patents and has published scientific papers in fields including computer vision and facial image processing, automated vehicle control, statistical machine learning, and high-dimensional optimization algorithms. Shumeet graduated from the University of Virginia with a Bachelor of Science, with high distinction, in 1991. He received his Ph.D. in computer science from Carnegie Mellon University in 1996.