TotalRecall: Visualization and Semi-Automatic Annotation of Very Large Audio-Visual Corpora

Rony Kubat
MIT Media Lab
20 Ames St. E15-486
Cambridge, MA
kubat@media.mit.edu

Philip DeCamp
MIT Media Lab
20 Ames St. E15-441
Cambridge, MA
decamp@media.mit.edu

Brandon Roy
MIT Media Lab
20 Ames St. E15-441
Cambridge, MA
bcroy@media.mit.edu

Deb Roy
MIT Media Lab
20 Ames St. E15-488
Cambridge, MA
dkroy@media.mit.edu

ABSTRACT

We introduce a system for visualizing, annotating, and analyzing very large collections of longitudinal audio and video recordings. The system, TotalRecall, is designed to address the requirements of projects like the Human Speechome Project [18], for which more than 100,000 hours of multitrack audio and video have been collected over a twenty-two month period. Our goal in this project is to transcribe speech in over 10,000 hours of audio recordings, and to annotate the position and head orientation of multiple people in the 10,000 hours of corresponding video. Higher level behavioral analysis of the corpus will be based on these and other annotations. To efficiently cope with this huge corpus, we are developing semi-automatic data coding methods that are integrated into TotalRecall. Ultimately, this system and the underlying methodology may enable new forms of multimodal behavioral analysis grounded in ultradense longitudinal data.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Graphical user interfaces (GUI)

General Terms

Human Factors, Design

Keywords

multimedia corpora, video annotation, speech transcription, visualization, semi-automation

ICMI'07, November 12-15, 2007, Nagoya, Aichi, Japan.

1. INTRODUCTION

The rapidly increasing capacity and decreasing cost of data storage is allowing the generation of very large databases of multimedia content. These data warehouses are currently present in the entertainment, medical and security industries, but will soon find their way into personal applications. Making sense of these massive corpora leads to new technical and user interface design challenges.

Inexpensive and very large capacity storage has also opened new doors in the behavioral sciences, allowing observational corpora to be collected at unprecedented scales. The Human Speechome Project (HSP)[18] motivates the work presented here. The goal of HSP is to study early language development though analysis of audio and video recordings of the first two to three years of a child's life. The home of the family of one of the authors (DR) with a newborn has been outfitted with fourteen microphones and eleven omnidirectional cameras. At the time of this writing, approximately 10 hours of audio and video have been captured on a daily basis from multiple cameras and microphones over the past 22 months. The corpus thus far consists of over 75,000 hours of audio and 35,000 hours video. This data provides many new opportunities to understand the fine-grained dynamics of language development. We plan to study the child's early words by tracing back to the contexts in which they were used by adults interacting with the child.

Video is recorded at approximately 15 frames per second, 1 megapixel resolution from cameras with fisheye lenses embedded in the ceiling. 16 bit, 48 KHz audio is recorded from ceiling mounted boundary layer microphones. To be useful for analysis, a large portion of speech that the child hears needs to be detected, tagged with a written transcript, and labeled with the appropriate speaker. Initial annotation of video content will focus on detecting and identifying all people in the home, with emphasis on head orientation providing insight into their shared attention. The video annotations provide information about the context in which words are used in natural child-caregiver interactions.

HSP entails some unique challenges for coding:

- HSP gathers far more data than strictly necessary for language development research. Audio and video is gathered in rooms without the infant present, when the child is asleep, and when non-linguistic activity occurs. Gathering this data is necessary, nevertheless, to ensure maximal coverage. The approximately 200 hours of multi-track audio and video recorded each day can be distilled into about three or four hours of continuous, pause-free speech relevant to the language research.

- Due to the unconstrained vocabulary and topic of natural speech in the home, the highly spontaneous nature of informal and child-directed speech, and the pragmatic necessity of placing microphones at a distance (ceiling mounted), current speech recognition technology is woefully inadequate for producing accurate speech transcriptions. As a result, human transcribers are being employed. Given the enormous quantity of speech to be transcribed, the efficiency of human transcribers is of paramount importance.

There are several existing tools that speech and language researchers currently use to code raw voice data with metadata tags such as transcription, prosody, speaker ID, etc[13, 2, 3, 12, 16, 17, 19, 14][1]. Coding using existing tools is tedious and time consuming. Furthermore, these tools are designed to work with small raw datasets, on the order of minutes or hours. Current coding tools are not designed to work smoothly with very large corpora such as HSP. We present a new software system, TotalRecall, designed to efficiently cope with very large corpora and to maximize the productivity of the human data annotator. Our strategy is to combine human and machine analysis in ways that leverage complementary strengths. TotalRecall increases speech transcription productivity by leveraging signal processing techniques to automatically classify sound segments, remove silences and pauses, and chunk speech into short, easily-transcribable sections. The idea is to use automatic methods to preprocess audio recordings and prepare easy-to-transcribe “soundbites” that a human transcriber may rapidly transcribe without being distracted by secondary tasks such as finding speech, selecting the best channel to listen to, and so forth.

In addition to speech transcription, we also seek to annotate salient aspects of human behavior in video. For example, when studying language acquisition by young children, one may want to look at the effects of caregiver proximity or joint attention: instances in which the caregiver and child both focus on the same object. To address these issues, TotalRecall contains functionality for annotating features of video, including person identification, location, and head orientation. Similar to audio, we have chosen a semi-automatic approach that combines manual annotation with automatic techniques such as a person tracker and head pose estimator.

Many of the key features of TotalRecall are motivated by one of the initial data coding tasks now underway in our lab: transcription of all speech heard and uttered by the child from ages 9-18 months. To perform this task, annotators must first locate and track the child’s location within the home, identify the audio channel which has the clearest signal (a conversation in the house will be picked up by microphones in many different rooms), find and mark the extents of speech within that channel, identify the speaker and finally transcribe the speech. TotalRecall helps automate these tasks.

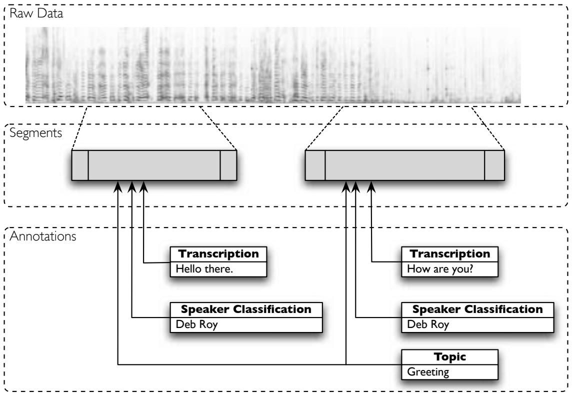


Figure 1: Three categories of TotalRecall data: raw, segments and annotations. Raw data is the recorded audio or video. Segments denote time extents within the raw data. Because TotalRecall handles many simultaneous channels of raw data, segments are also labeled with a channel ID (not shown in this illustration). Annotations are grounded to segments in a many-to-many relationship. In this case two utterances are linked to metadata tags identifying speaker and transcription. A topic tag has been grounded to both segments.

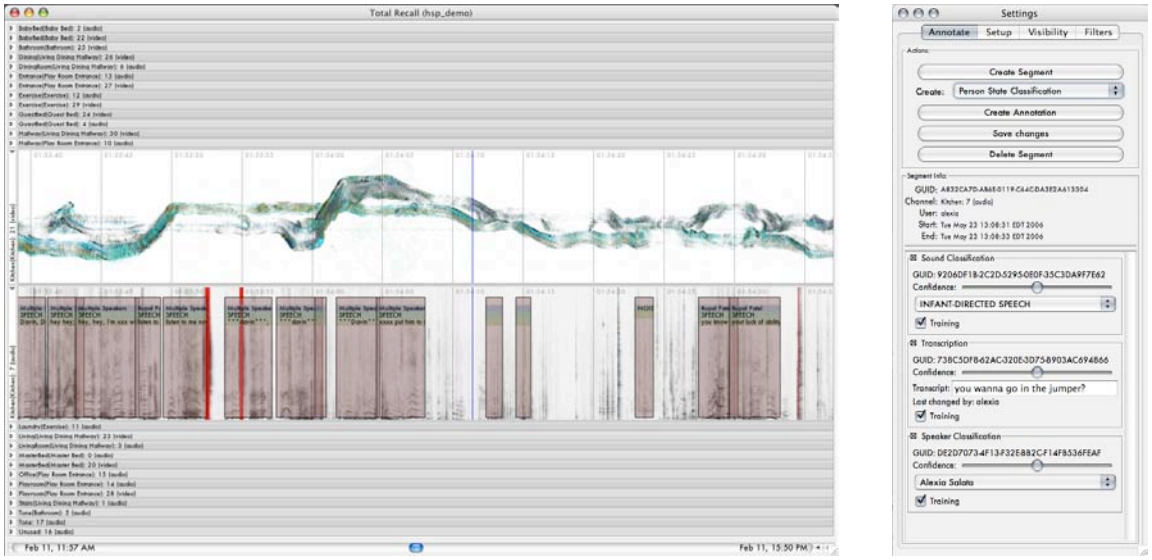
The next sections describe the infrastructure and algorithms that make this semi-automation possible. In the first section, we introduce the TotalRecall annotation model. Next we describe the TotalRecall interface and explain how it is optimized to cope with hundreds of thousands of hours of data via its visualizations and GUI. The algorithms used to process video and audio are described in sections 5 and 6, respectively. We describe how these algorithms increase the efficiency of the human annotator’s coding tasks. We conclude with a discussion of avenues for future work.

2. ARCHITECTURE AND THE ANNOTATION MODEL

Data within TotalRecall is divided into three broad categories: raw, transform and meta. *Raw* data is the original unmodified video and audio streams as recorded on-site. *Transform* data is fully automated transformations of the raw data. These transformations are frequently lossy. Examples of transform data include power spectral coefficients used to display spectrograms of audio and “video volumes” used to display movement patterns within video (see section 4.2). *Meta*data refers to annotations that involve human interpretation. Examples of metadata include speech transcriptions, speaker and activity classifications and motion tracks of people. Unlike raw and transform data, which are stored in a time-based file hierarchy, metadata is stored in a relational database.

Raw and transform data are divided into files, each of which contain one minute of data for one channel. A file tree rooted at the year and progressing down through month, day, hour and minute holds these files. TotalRecall automatically stitches each channel’s disparate data sources into a

[1] See [7] for a review of these and other annotation tools.



Create: Person State Classification

Segment Info:
GUID: A832CA7D-A86E-0119-C64CBA3E2A613304
Channel: Kitchen: 7 (audio)
User: alexia
Start: Tue May 23 13:08:31 EDT 2006
End: Tue May 23 13:08:33 EDT 2006

Sound Classification
GUID: 9206DF1B-2C2D-5295-0E0F-35C3DA9F7E62
Confidence:
INFANT-DIRECTED SPEECH
☑ Training

Transcription
GUID: 73BC5DF8-62AC-320E-3D75-8903AC694866
Confidence:
Transcript: you wanna go in the jumper?
Last changed by: alexia
☑ Training

Speaker Classification
GUID: DE2D7073-4F13-F32E-B82C-F14FB536FEAF
Confidence:
Alexia Salata
☑ Training

Figure 2: Screenshots of the TotalRecall system. In the left image, one video and one audio channel are being displayed. The video channel displays video volumes tracing two people moving in the kitchen. Below it, the accompanying audio track shows transcription and speaker ID annotations over spectrograms of the sound. The details window on the right shows additional information about the currently selected segment and its attached annotations.

single continuous stream. This allows TotalRecall to handle audio and video data of arbitrary length, with limits imposed only by storage capacity.

Annotations within TotalRecall are *grounded* to the raw data via many-to-many relationships. The intermediary we use are *segments*, globally unique locators to extents of time and particular audio or video channels. Each segment is a triplet: channel ID, start time and stop time, where the times are measured at millisecond resolution in UTC. The channel ID is an identifier of the microphone or camera with which the raw data was recorded. Figure 1 shows an example where a single annotation is grounded to two distinct segments and each segment is linked to three annotations.

TotalRecall is built to handle many millions of annotations across tens of channels and years of raw data. It is a collaborative tool, designed to support multiple users with different levels of data access (maintaining the privacy of the HSP subjects). Each annotation is tagged with the user who created it, as well as time(s) of creation and subsequent modification. Because annotations are grounded to segment intermediaries, changes to a segment's extent propagate to all linked annotations. Overall program efficiency is increased by storing metadata in a relational database. This allows sections of raw data to be coded by multiple annotators, and lets statistical inferences about the quality of annotations be made.

3. USER INTERFACE

The user interface of TotalRecall is organized into three primary windows. Figures 2 and 3 show screenshots of the TotalRecall system. In figure 2, the first window displays a synchronized timeline view of the multiple audio and video channels. Users view transform data in this window as well as view and edit annotations. The second window (the detail view) displays additional information about selected segments and holds the various configuration options for the program. The third window (figure 3) displays a single video channel at full resolution and ten video channels at reduced resolution and acts as a basic video player. This window is also used to make annotations to video.

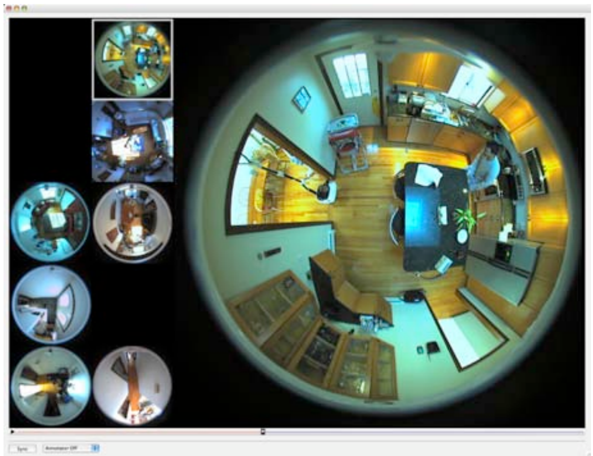


Figure 3: The TotalRecall video player, displaying low resolution video clips of all active channels on the left, and a single full resolution clip of the selected channel on the right.

The timeline interface of TotalRecall borrows from the visual vocabulary of multitrack audio and video editors in

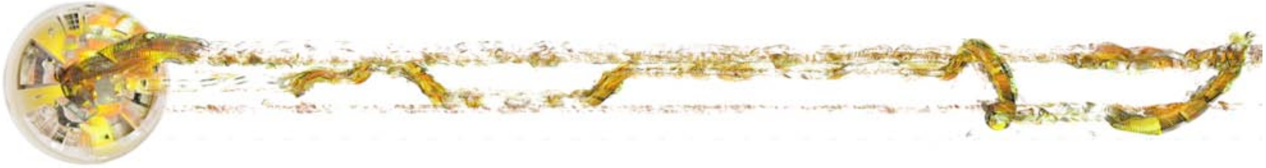


Figure 4: Detailed view of a video volume, showing two people moving. The video frame is displayed for clarity.

order to minimize training time of human annotators. Like these editors, the user can "zoom" the temporal resolution displayed though a continuum of ranges. The timeline view spans seven orders of magnitude of temporal resolution: at widest, a TotalRecall user can examine a year's activity, and at the highest resolution, she can examine sub-second intervals. A fisheye view[9] of the timeline allows users to examine and annotate a particular moment in time while being aware of the annotations which make up the context. A scroll bar at the bottom of the window allows the annotator to change the view backward and forward in time. Scrolling in this way is done nonlinearly. As the user strays farther from the initial position, the view zooms out. We have found it qualitatively easier to maintain a clear sense of the period of time under examination in this way.

4. VISUALIZATIONS

The visualizations for metadata and transform data are shown in the timeline window. Visualizations are viewable across all levels of temporal zoom. The visibility of data visualizations can be filtered. For example, the user can choose that only segments containing transcripts by a specific speaker are visible. Visibility for transform data is coarse (on/off).

4.1 Metadata: Segments and Annotations

The basic metadata of temporal extents (*segments* as described in section 2) are displayed as rectangles. Within these rectangles are strips for the annotations which are grounded to it. Each strip shows a summary of that annotation's data; for example speaker ID, transcript, etc. Segments are selected by clicking on them or by keyboard navigation. When selected, details about the segment appear in the detail window. There, less-frequently needed metadata such as confidence score, modifying user and globally-unique ID, are displayed. The size of the segment rectangle limits the number of simultaneous annotations that can be seen at once in the timeline view. This limitation does not exist in the detail view.

4.2 Transform data

In the timeline window of TotalRecall, video is displayed using a video volume technique similar to [6]. Visualizing video in this manner allows an annotator to rapidly find activity in the house and mark the video channel that shows the infant. One day's video—approximately 50 hours of raw data—can be so annotated in approximately 2 hours.

The video volume visualization works by first applying an alpha mask to each frame of video such that pixels which contain motion or activity are made opaque and pixels which remain static are made transparent. Specifically, the alpha value for each RGB pixel is computed as the scaled, absolute distance between itself and the corresponding pixel in the previous frame of video. Each processed frame is then placed on a horizontal timeline such that each frame overlaps the previous frame almost entirely, causing time and horizontal space to intermix on the horizontal axis, while vertical space occupies the vertical axis. The resulting image displays people as worm-like patterns that give an immediate summary of the locations of people throughout the video. This technique relies on a fixed camera position. Figure 4 shows a detailed view of a video volume.

Similarly, audio is displayed visually using spectrograms, enabling users to locate, and often visually identify, different types of audio activity.

The timeline window can display one additional video visualization, called "actigrams." Actigrams show the level of motion occurring in multiple, hand-coded regions of interest, such as "motion near the refrigerator" or "motion at the dining table". While actigrams may provide a human annotator with a very coarse summary of activity, they are specifically designed to be easily readable by machine and have been used as input for the activity recognition system described in [8]. This system uses actigrams to identify and annotate such activities as a person making a cup of coffee or getting a drink of water.

5. VIDEO ANALYSIS

TotalRecall contains functionality to annotate the identity, location, and head orientation of people in video. Head orientation provides an estimate of gaze direction where eye tracking is not possible. These annotations are created in the video player component of TotalRecall. The annotator may specify person location in any given frame of a video stream by drawing a bounding box around a person and selecting an identity label from a popup menu. When annotating head orientation, the annotator first draws a bounding box around a head, which causes three images to appear on top of the video frame as shown in Figure 5. The first image shows a 3D head model, the second shows a magnified view of the real head, and the third overlays the 3D model on top of the real head. Using an off-the-shelf input device with six degrees of freedom, the annotator may adjust the position, scale, roll, pitch and yaw of the 3D head until it aligns to the real head.

While annotating a single frame in this manner takes only a few seconds, annotating one hour of continuous video would require at least several hours of labor. Several efforts are underway to increase the efficiency of these annotation tasks. First, a person tracker has been integrated into To-

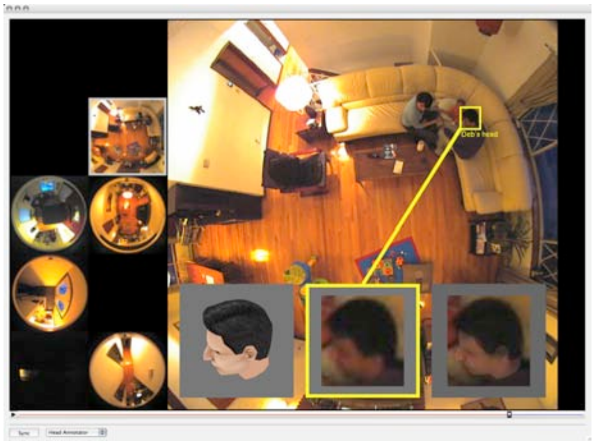


Figure 5: The head annotator allows the user to specify head pose by manipulating a 3D model with a 6-DOF input device.

talRecall to automate the location annotation task and to propagate identity labels throughout sections of video. Second, a head pose estimator is under development to automate the head annotation task.

5.1 Person Tracking

The prototype tracker is based on the mean-shift algorithm[5]. Typically, such a tracker is initialized by providing a single bounding box around a target. The target is modeled as the color distribution within this box. In each subsequent frame of video, a nearby region is located which contains a color distribution that best matches the target model.

In contrast to trackers that use only motion information, the mean-shift tracker provides excellent stability in cases where a person remains still for a long period of time, which is crucial when the video is taken from a home. A common enhancement is to continuously update the target model, which increases invariance to lighting and target orientation. The tracker in TotalRecall is further augmented to incorporate motion data from a fast foreground-background segmenter[15].

5.2 Head Pose Estimation

The HSP video poses several challenges for accurate head pose estimation: the cameras are mounted overhead and often only the hair and forehead are visible, a typical head is only 20x20 pixels in resolution, and the lighting conditions vary considerably. The current head pose estimator is designed to address these issues by using a textured 3D model-based approach similar to the system described in [4].

The estimation process begins with a single annotated head provided by the user, as described in section 5. The head annotation uses six parameters used to define head pose: three for angular orientation, two for position, and one for scale. For each subsequent video frame, the estimation algorithm can be summarized as a three step process. First, the system applies a mean-shift tracker to obtain the approximate region of the head. Second, a color classifier is run on each pixel in this region. The classifier uses a Gaussian mixture model to determine the probability that a given pixel is hair, skin, or neither. Based on the classification results, each pixel is transformed from an RGB vector to a vector that contains the probability of skin, probability of hair, probability of background, and the luminance of the original pixel. Last, a hill climbing search is performed over all six pose parameters. This search is performed by rendering a 3D model of the target head in a number of poses and finding the pose that maximizes the cross-correlation between the rendered head and the real head. For best accuracy, it is necessary to manually construct a 3D model for each person's head, which may not be feasible for data sets containing a larger number of people.

The head pose estimation system is currently under development and the initial results from this system are not yet available. Although we expect the system to make frequent mistakes, we believe it will be sufficient to automatically annotate short stretches of video and greatly reduce the number of annotations that must be made manually.

6. AUDIO ANALYSIS

There are three components of the audio analysis which help semi-automate the tasks of speech transcription and speaker identification. First, the best audio channel is selected from the 14 tracks based on signal power. This forms a "virtual channel," which is then processed into short speech segments by the speech detection algorithm. Finally, a speaker identification algorithm is applied to the segments labeled as speech.

6.1 Channel selection

The channel selection algorithm computes the power of the audio channels as a function of time. For each audio channel, we compute the RMS amplitude of a sliding 300ms window of samples. Choosing the channel with highest power at each window sometimes results in overly rapid switching between channels. Smoothing is performed using a dynamic programming cost minimization algorithm that assigns a fixed cost for switching channels and a cost for staying in the channel that is not the currently observed loudest. By changing the relative values of these costs, the degree of smoothing can be adjusted. The output of the channel selection algorithm is a set of segments.

One shortcoming of this approach is the inability to distinguish multiple simultaneous sound sources in different parts of the house (i.e., captured in different concurrent audio tracks). In such cases, only the louder source will be detected. Fortunately, it appears that such situations are relatively rare in the speechome corpus.

6.2 Voice detection and segmentation

The sequence of segments from channel selection serves as the input to the speech detection and segmentation algorithm. The algorithm's output is a new set of speech segments and confidence estimates. The algorithm works by first downsampling the 48KHz input audio stream to 8Khz, partitioning the audio stream into 30ms frames, extracting a feature vector for each frame, classifying the feature vector as speech or non-speech, and then applying a smoothing and segmenting algorithm to the sequence of frame labels. Smoothing and segmenting for speech detection is similar to that of channel selection.

There are constraints on the minimum and maximum length of a speech segment. If a segment is too long, it is split into multiple segments by finding the "split points" that have both a minimum energy (we prefer to split at silence points) and yield a small number of new segments that satisfy the length constraints. A confidence score is also returned for the speech segment, based on the fraction of frames labeled as speech.

The feature vector used to classify each frame contains 13 MFCCs[10, 20], energy at two different frame widths, zero crossing rate, spectral entropy, maximum amplitude and the relative power between different frequency bands. In total, there are 19 feature dimensions. Each frame is 30ms long, with a frame shift of 15ms. The classifier is built with the Weka machine learning library[22], using boosted decision trees. Training proceeds by gathering a set of human verified speech and non-speech segments from the database, extracting the above-mentioned features for each segment, and training the algorithm using 5-way cross-validation. The notion of human verified segments includes speech annotations modified by a human, or speech annotations attached to a segment that has other human modified annotations.

6.3 Speaker identification

The speaker identification algorithm processes a set of speech segments, and outputs speaker annotations grounded to these segments. Feature extraction begins by first downsampling the audio stream to 16KHz. The features used for classification are a superset of the speech detection features in that MFCC deltas and double deltas are also extracted[10]. Components of the Sphinx-4[21] package are used for this extraction. Each 30ms frame is independently classified into one of the possible speaker classes. The final classification for the segment is the majority vote of all the frame level classifications. A confidence score is also returned which is the fraction of frame classifications with the final segment label. If the confidence is below a specified threshold, then no classification is returned.

7. ANNOTATION TASKS

TotalRecall is optimized for certain coding tasks. Primary among these is speech transcription. The previous section described the algorithms used to find and segment speech. Once these automatic steps are complete, a human annotator must still listen to the recordings and enter the actual transcription. The transcription task proceeds as follows: the annotator selects a speech segment and adds a transcription annotation, playing the audio only for that segment with a single keystroke. Once a transcription is entered, pressing the return key automatically advances to the next speech segment and begins to play it. Playback speed can be adjusted by the annotator with a pitch-shifting system that maintains intelligibility[11]. The primary time savings comes by automatically segmenting the audio into short spoken utterances and by limiting the amount of mouse movement required to create an annotation.

The task of labeling speech segments with speaker ID (a necessary step to train the speaker classifier algorithm) is optimized with a user interface similar to speech transcription. The TotalRecall user configures single keys to add a speaker ID annotation to a selected segment. When this key is pressed, the current segment is annotated and then the next speech segment is automatically selected and played.

Trial	Tool	Average time (mins)
1	CLAN	50.0
2	CLAN	44.5
3	Transcriber	32.6
4	Transcriber	27.0
5	TotalRecall	21.5
6	TotalRecall	20.5

Table 1: Average transcription times for transcribing five minute audio segments containing similar amounts of speech.

8. EVALUATION

A pilot evaluation of TotalRecall compared it to CLAN[13] and Transcriber[2]. In this evaluation, transcription and segmentation were performed completely manually in TotalRecall. A separate evaluation looked at manual segmentation time in TotalRecall to get a sense of the time demands of the segmentation task, which the automatic speech detection component eliminates. Further evaluations are needed to compare transcription in the semi-automatic system against other systems.

Audio from the Human Speechome Project was exported to several WAV files that could be processed with CLAN and Transcriber. Only audio containing dense speech in a single channel was used, to be fair to CLAN and Transcriber which do not support multi-channel audio. Both CLAN and Transcriber are split into a transcription panel and a audio waveform panel. Segments of audio can be highlighted, played, and associated to transcriptions. One difference between CLAN and Transcriber is that Transcriber does not allow overlapping segments. In CLAN, transcription typically proceeds by identifying candidate speech from the waveform display, highlighting and playing a segment of the audio, and refining the segment until the desired speech clip is highlighted. A transcription is then typed, and bound to the segment. In Transcriber, an annotator might perform a coarse segmentation of the speech first, and then transcribe and refine segment boundaries. Alternatively, an annotator might segment and transcribe simultaneously.

Two annotators were instructed to transcribe speech only, ignoring speaker identity. In CLAN, entering the speaker code is part of the transcription syntax, so entering the true speaker identity did not seem to be more time consuming than typing a single code for all speakers. Each annotator practiced using the tool on a short trial run before performing their transcription task, which they timed using a stopwatch program. In TotalRecall, the same method was employed, focusing on a single channel of audio. Hot-keys for creating segments and transcriptions were defined, eliminating the need for excessive mouse movement. Table 1 and figure 6 compare the average annotator times for these transcription tasks. For each task, the individual annotator times were within a few minutes of each other. This evaluation shows that TotalRecall is, by itself, a faster transcription tool than both CLAN and Transcriber.

An evaluation of manual speech segmentation (without transcription) was conducted for TotalRecall in order to determine how time is required by a human annotator to segment speech. Two ten minute blocks of single channel HSP audio were segmented in TotalRecall by human annotators. They took roughly 22 and 24 minutes each. Time checks

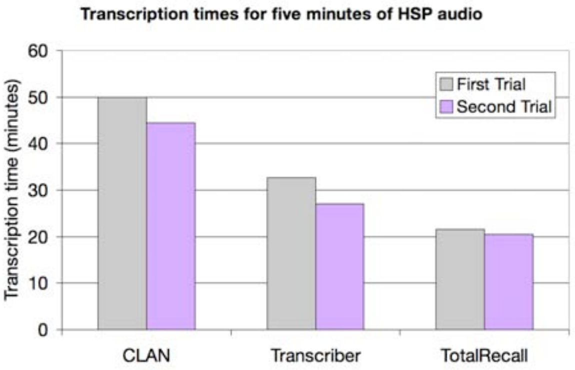


Figure 6: Graph of transcription times from table 1.

at intermediate points during the segmentation showed that segmentation time to progressed at a steady rate. This suggests that segmenting a block of audio in TotalRecall takes approximately twice as long as the actual audio duration. Assuming this segmentation rate, about 10 minutes of the transcription time for TotalRecall in table 1 is due to segmentation. Although there may be subtleties to transcribing automatically segmented speech, such as segmentation errors introduced by system, automatic segmentation should still provide significant time savings over purely manual segmentation and transcription.

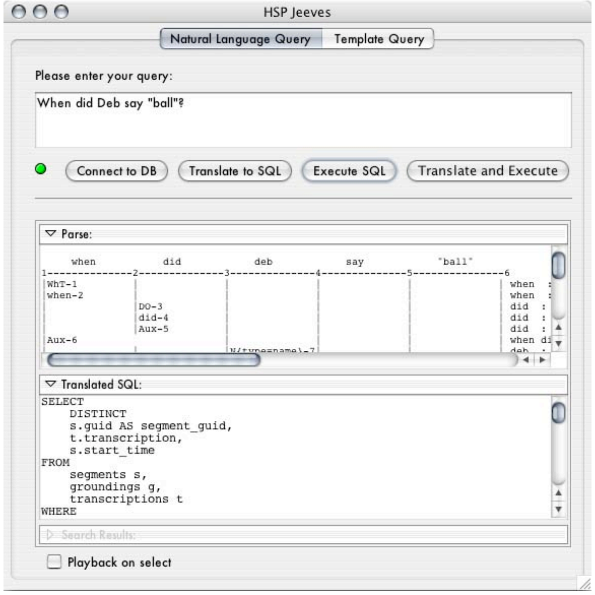
9. BROWSING AND SEARCHING

As the amount of data in TotalRecall grows, finding annotations can become difficult. By using a relational database to store annotations, queries can be arbitrarily complex and efficiently executed. Nevertheless, exposing the relational back end to the user requires them to be familiar with the database schema as well as the query language. TotalRecall massively reduces the complexity of making queries by providing a natural-language query tool. Input sentences are parsed and the parse tree evaluated with lambda functions associated with a hand-written grammar. These evaluations translate the input query into valid SQL[1]. Figure 7 shows a screenshot of the query tool.

10. CONCLUSIONS

The key contributions of TotalRecall are its user interface and integration with signal processing algorithms designed to semi-automate annotation tasks. The interface maximizes the efficiency of annotators and allows them to browse a continuum of multichannel recordings and metadata. Specific coding tasks such as speech transcription and speaker identification are highly optimized by: using video volumes to locate human activity, processing audio to automatically select the best signal to annotate and pre-segment utterances, adjusting playback speed and minimizing the user's need to switch between mouse and keyboard. The annotation model used by TotalRecall is flexible, extensible, scalable and efficiently queried.

One of our long term goals is to study the development of specific words by tracing all contexts in which a word was used by either child or caregivers. Our plan is to use TotalRecall to transcribe all child-directed and child-generated speech in the corpus, and then use the speech transcripts as an index into co-occurring video. Speech transcripts will focus our efforts to annotate selected portions of video using the video analysis methods described in this paper. Building on these speech and video coding efforts, we will investigate the role of various cross-modal behavioral interactions in early language acquisition.



Please enter your query:

When did Deb say "ball"?

Parse:

```
    when          did           deb           say          "ball"
1-------------2-------------3-------------4-------------5--------------6
|WhT-1        |             |             |             |              | when  :
|when-2       |             |             |             |              | when  :
|             |DO-3         |             |             |              | did   :
|             |did-4        |             |             |              | did   :
|             |Aux-5        |             |             |              | did   :
|Aux-6        |             |             |             |              | when di
```

Translated SQL:

```
SELECT
    DISTINCT
    s.guid AS segment_guid,
    t.transcription,
    s.start_time
FROM
    segments s,
    groundings g,
    transcriptions t
WHERE
```

☐ Playback on select

Figure 7: Screenshot of the TotalRecall natural language query tool.

11. ACKNOWLEDGMENTS

We would like to thank all the users of TotalRecall, especially Alexia Salata, for their feedback about TotalRecall. Jethran Guinness provided technical support with the disk storage array that underlies TotalRecall. This paper is based upon work supported under a National Science Foundation Graduate Research Fellowship.

12. REFERENCES

[1] J. Allen. *Natural language understanding (2nd ed.)*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 2 edition, 1995.

[2] C. Barras, E. Geoffrois, Z. Wu, and M. Liberman. Transcriber: development and use of a tool for assisting speech corpora production. *Speech Communication*, 33(1-2):5–22, 2001.

[3] P. Boersma. Praat, a system for doing phonetics by computer. *Glot International*, 5(9/10):341–345, 2001.

[4] X. L. C. Brolly, C. Stratelos, and J. B. Mulligan. Model-based head pose estimation for air-traffic controllers. In *ICIP (2)*, pages 113–116, 2003.

[5] V. Comaniciu and P. Meer. Kernel-based object tracking, 2003.

[6] G. Daniel and M. Chen. Video visualization. In R. M. Greg Turk, Jarke J. van Wijk, editor, *IEEE Visualization 2003*, pages 409–416, Seattle, Washington, USA, October 2003. IEEE Press.

[7] L. Dybkjær and N. O. Bernsen. Towards general-purpose annotation tools—how far are we today? In *Proceedings of the Fourth International Conference on Language Resources and Evaluation LREC'2004*, volume I, pages 197–200, Lisbon, Portugal, May 2004.

[8] M. Fleischman, P. Decamp, and D. Roy. Mining temporal patterns of movement for video content classification. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 183–192, New York, NY, USA, 2006. ACM Press.

[9] G. W. Furnas. Generalized fisheye views. In *CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 16–23, New York, NY, USA, 1986. ACM Press.

[10] H. Gish and M. Schmidt. Text-independent speaker identification. *IEEE Signal Processing Magazine*, October 1994.

[11] D. Henja and B. Musicus. The solafs time-scale modification algorithm. Technical report, BBN, July 1991.

[12] M. Kipp. Anvil – a generic annotation tool for multimodal dialogue. In *Proc. Eurospeech.*, 2001.

[13] B. MacWhinney. *The CHILDES Project: Tools for Analyzing Talk*. Lawrence Erlbaum Associates, Mahwah, NJ, 3^{rd} edition, 2000.

[14] K. Maeda, S. Bird, X. Ma, and H. Lee. The annotation graph toolkit: software components for building linguistic annotation tools. In *HLT '01: Proceedings of the first international conference on Human language technology research*, pages 1–6, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[15] A. Manzanera and J. Richefeu. A robust and computationally efficient motion detection algorithm based on sigma-delta background estimation. *Proceedings Indian Conference on Computer Vision, Graphics and Image Processing*, 2004.

[16] J. Milde and U. Gut. The tasx-environment: An xml-based corpus database for time-aligned language data. In *Proceedings of IRCS workshop of linguistic databases.*, 2001.

[17] D. Reidsma, N. Jovanović, and D. Hofs. Designing annotation tools based on properties of annotation problems. In *Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research*, 30 August - 2 September 2005 2005.

[18] D. Roy, R. Patel, P. DeCamp, R. Kubat, M. Fleischman, B. Roy, N. Mavridis, S. Tellex, A. Salata, J. Guinness, M. Levit, and P. Gorniak. The human speechome project. In *Proceedings of the 28th Annual Cognitive Science Conference.*, pages 2059–2064, 2006.

[19] K. Sjlander and J. Beskow. Wavesurfer - an open source speech tool. In *Proc. of ICSLP*, volume 4, pages 464–467, Beijing, Oct. 16-20 2000.

[20] G. Tzanetakis and P. Cook. Multifeature audio segmentation for browsing and annotation. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, October 1999.

[21] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel. Sphinx-4: A flexible open source framework for speech recognition. *Sun Microsystems Technical Report*, (TR-2004-139), November 2004.

[22] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, June 2005.