Your Reactions Suggest You Liked the Movie: Automatic Content Rating via Reaction Sensing

Xuan Bao
Samsung Research

Songchun Fan
Duke University

Alexander Varshavsky
AT&T Research

Kevin A. Li
AT&T Research

Romit Roy Choudhury
Duke University

ABSTRACT

This paper describes a system for automatically rating content - mainly movies and videos - at multiple granularities. Our key observation is that the rich set of sensors available on today's smartphones and tablets could be used to capture a wide spectrum of user reactions while users are watching movies on these devices. Examples range from acoustic signatures of laughter to detect which scenes were funny, to the stillness of the tablet indicating intense drama. Moreover, unlike in most conventional systems, these ratings need not result in just one numeric score, but could be expanded to capture the user's experience. We combine these ideas into an Android based prototype called *Pulse*, and test it with 11 users each of whom watched 4 to 6 movies on Samsung tablets. Encouraging results show consistent correlation between the user's actual ratings and those generated by the system. With more rigorous testing and optimization, *Pulse* could be a candidate for real-world adoption. [1]

ACM Classification Keywords

H.5.0 Information Interfaces and Presentations: General

Author Keywords

Content Rating, Mobile Phones, Reaction Sensing, Context

INTRODUCTION

Online content ratings serve as "quality indicators" to help a user make more informed decisions. While these ratings have been effective, we believe that there is room for improving the value and experience with ratings. Our observations are two-fold: (1) Today's ratings are most often a simple number, such as a "4 star" for a Netflix movie, a 87% red-tomato by Flixster, or simply 23 Likes for videos in YouTube. These numbers may be viewed as a *highly-lossy compression* of the viewer's experience, that often leaves the new user asking for more. (2) Eliciting a carefully considered rating from users is difficult, partly due to the lack of incentives. Providing a brief review can take up a good amount of user's time. Once a user has watched the video, she may not be willing to make this time investment. We envision that content rating systems of the future will require minimal user participation and yet provide rich, informative ratings. Figure 1 shows an example – a movie thumbnail could not only have a star rating, but also a tag-cloud of user reactions, and even short clips indexed by these reactions (such as, all scenes that were hilarious).

This paper makes an attempt to realize this vision through a system called *Pulse*. The opportunity arises from the growing number of sensors that are entering the mobile platform, especially smartphones and tablets. We hypothesize that when users watch a movie on these devices, a good fraction of their reactions leave a footprint on various sensing dimensions. For instance, if the user frequently turns her head and talks – detectable through the front facing camera and microphone – one could infer the user's lack of attention to that movie. Other kinds of inferences may arise from laughter detection via the microphone, the stillness of the device from the accelerometer, variations in orientation from gyroscope, fast forwarding of the movie, etc. Pulse learns the mapping between the sensed reactions and these ratings. Later, the knowledge of this mapping is applied to users to automatically compute their ratings, especially when they do not provide one. The sensed information is also used to create a tag-cloud of reactions, expected to offer a "break-up" of the different emotions evoked by the movie. If one wishes, she may also be able to watch a set of short clips that pertain to any of these emotions. Pulse can provide them since it logs user reactions for each segment, across many users. The result is like a customized trailer [9], one per user reaction.

Figure 1. Envisioned movie ratings for the future – a conventional 5-star rating; a tag-cloud of user reactions; movie clips indexed by these reactions.

[1]The research leading to these results was done during Xuan's PhD in Duke University

UbiComp'13, September 8–12, 2013, Zurich, Switzerland.

http://dx.doi.org/10.1145/2493432.2493440

The core ideas in Pulse may generalize to a variety of applications: (1) The timeline of a movie can be annotated with reaction labels (e.g., funny, intense, warm) so that viewers could jump ahead to desired segments. (2) The advertising industry may use Pulse to offer free or subsidized movies in exchange for more targeted ads. A user who reacts to a particular scene could be presented with corresponding ads. (3) It may be feasible to create an automatic highlights of a movie, perhaps consisting of all action scenes. (4) Finally, Pulse may offer educational value to film institutes and mass communication departments – students can use reaction logs as case studies from real-world users.

Of course, translating Pulse to reality, and enabling these applications, entails a number of challenges. The viewer's head pose, lip movement, and eye blinks need to be detected and monitored over time to infer reactions [5]. The user's voice needs to be separated from the sounds of the movie (which may be audible if the user is not wearing headphones), and classified as either laughter or speech. Patterns in accelerometers and gyroscopes need to be identified and translated to user focus or distractions. Finally, the function that translates reactions to ratings needs to be estimated through machine learning, and the learnt parameters used to generate semantic labels as a summary about the movie [17, 23].

This paper incorporates these ideas into a Samsung tablet running the Android OS, and distributes these tablets to real users for evaluation. Results indicate that Pulse's final ratings are consistently close to the user's ratings (mean gap of 0.46 on a 5 point scale), while the reaction tag-cloud reliably summarizes the dominant reactions. The highlights feature also extracted the appropriate segments, while the energy footprint remained small and tunable. A small-scale user study generated an enthusiastic response to Pulse.

The main contributions may be summarized as follows.

- **We identify an opportunity to automatically rate content at a few different granularities.** Our approach requires minimal user participation and harnesses multi-dimensional sensing available on modern tablets and smartphones.

- **We design a practical system, Pulse, that senses user reactions and translates them to an overall system rating.** In addition, we process the raw sensor information to produce rating information at variable granularities – a tag-cloud and a reaction-based highlight.

- **We develop Pulse on Android based Samsung Galaxy tablets and evaluate it with 11 volunteers, each of whom watched 4 to 6 movies**. Results show that the average gap between human and system ratings is 0.46 (on a 5 point scale). The tag-cloud exhibits similarity to the user's true reactions, thereby capturing reasonably, the user's overall experience.

The rest of the paper expands on each of these contributions, beginning with a high level overview, and followed by design, implementation, and evaluation.

SYSTEM OVERVIEW

Pulse has been implemented on Android tablets and focuses specifically on movies and videos. Figure 2 envisions the high level architecture. This section briefly describes the three main modules, namely (1) Reaction Sensing and Feature Extraction (RSFE), (2) Collaborative Labeling and Rating (CLR), and (3) Energy Duty-Cycling (EDC).

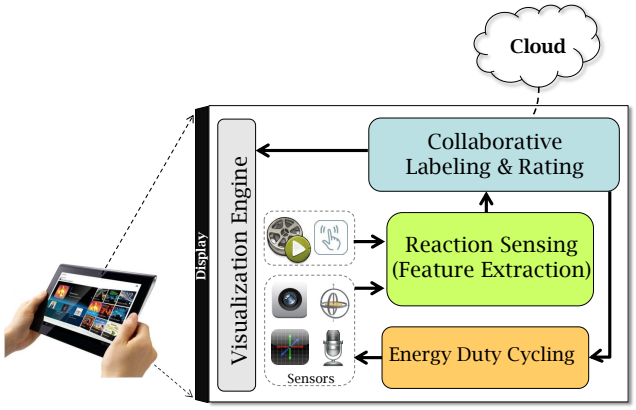


Figure 2. Architectural overview of Pulse.

(1) Reaction Sensing/Feature Extraction (RSFE)

When a user watches a video via the Pulse media player, all relevant sensors are activated, including the (front-facing) camera, microphone, accelerometer, gyroscope, and available location sensors. The raw sensor readings are forwarded to the RSFE module, which is tasked to distill out the features from them. These features inlude visual features collected through the front-facing camera, acoustic features extracted from embedded microphone, motion features (acceleration, rotation) captured by motion sensors, and control operations (e.g., fast forward) detected by the media player. RSFE collects all these features and forwards them to the collaborative labeling and rating (CLR) module.

(2) Collaborative Labeling and Rating (CLR)

Content storage and streaming, especially with movies and videos, is moving towards the cloud based model. The ability to assimilate content from many cloud users naturally offer insights into behavior patterns of a collective user base [21] – Netflix, Amazon, Hulu, are examples of service providers that leverage this approach to provide recommendation and personalization. Pulse is also positioned to benefit from access to the cloud. In particular, Pulse employs *collaborative filtering* methods where ratings are used across users to help improve accuracy. With more labeled data from users, Pulse will improve in its ability to learn and predict user ratings.

Sensing user reactions and exporting to the cloud raises privacy concerns [12], especially with face detection. However, we observe that none of the raw sensor readings need to be shared. Upon approval from the user, only ratings and semantic labels (or any subset of them with which the user is comfortable) can be exported. In the degenerate case, Pulse uploads the final star rating and discards the rest. This mimics today's systems, except that the rating will be determined automatically.

(3) Energy Duty-Cycling (EDC)

When the tablet is connected to a power-outlet, the EDC module is not necessary. In fact, we find that Pulse's additional energy consumption due to sensing is marginal compared to the energy consumed by the tablet's display and CPU, while playing the movie. However, when running on smartphones, EDC's task is to minimize the energy consumption due to sensing. As mentioned earlier, the key idea is to sense each user during non-overlapping time segments, and then "stitch" the user reactions to form the overall rating. The evaluation section presents measurement results.

Figure 3 shows how the different sub-modules lead up to the final rating. The RSFE module processes the raw sensor readings and extracts features to feed to CLR. The CLR module processes each (1 minute) segment of the movie to create a series of "*semantic labels*" as well as "*segment ratings*". Techniques such as collaborative filtering, Gaussian process regression (GPR), and support vector machines (SVM) are employed to address different types of challenges. Finally, the segment ratings are merged to yield the final "star rating" while the semantic labels are combined to create a tag-cloud. Thus, from the raw sensor values to the final star rating, Pulse distills information at various granularities to generate the final summary of the user's experience. We begin the technical discussion with the RSFE module.

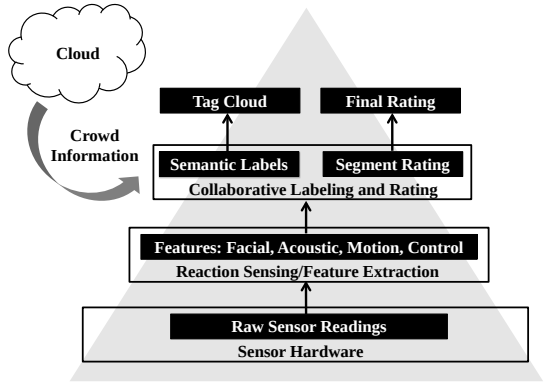


Figure 3. The RSFE and CLR modules distill raw sensor readings to a rating, tag-cloud, and video trailers

SYSTEM DESIGN: RSFE

We discuss the design of the Reaction Sensing and Feature Extraction module (RSFE).

Reaction Features: Visual

Pulse records visual information from the front camera to assess human reactions. Several prior efforts have attempted to achieve this using techniques involving face detection, eye tracking, and lip tracking [15]. However, our application presents a few unique challenges and opportunities compared to the traditional scenarios. First, the front facing camera on a mobile device usually does not capture the user's face from an ideal angle. In the case of our tablet, the top-mounted camera usually captures a tilted view of the face and eyes, requiring us to compensate for a rotational bias. Second, due to relative motion between the user and the tablet, the user's face may frequently move out of the camera view, either fully or partially. This derails contour matching methods, making continuous face detection difficult. Third, practical issues such as users wearing spectacles adds to the complexity. Fortunately, however, the field of view of the tablet is usually limited, making it easier to filter out unknown objects in the background, and extract the dominant user's face. Also, for any given user, particular head-poses are likely to repeat more than others (due to the user's head-motion patterns).

Pulse employs a combination of face detection, eye tracking, and lip tracking, using techniques from contour matching, speeded up robust feature (SURF) detection [3], and frame-difference based blink detection algorithms [15]. The flow of operations is as follows:

1. Pulse continuously runs a contour matching algorithm on each frame for face detection.
2. If a face is detected, the system runs contour matching for eye detection as well as lip detection, and identifies the SURF image keypoints in the region of the face. These image keypoints may be viewed as small regions of the face that maintains similar image properties across frames.
3. Now, if a full face is not detected, Pulse still tracks keypoints similar to previously detected SURF keypoints – this allows detecting and tracking a partial face, which occurs frequently in real life.
4. Pipelined with the face detection, Pulse runs an algorithm to perform blink-detection and eye-tracking. The difference in two consecutive video frames are analyzed to identify a blink. Essentially, if the pixels that change across consecutive frames form two nearly-symmetric ellipses, then the pixels are likely to be the blink. For eye-tracking, contour matching-based techniques fail when users are wearing spectacles – blink-detection is effective here. In other words, even if the eyes are blurred by the spectacles, the blinks can approximate the eye positions.

Figure 4 shows an intermediate output of the algorithm. Here Pulse detects the face through the tablet camera, detects the eyes using blink detection, and finally tracks the keypoints.

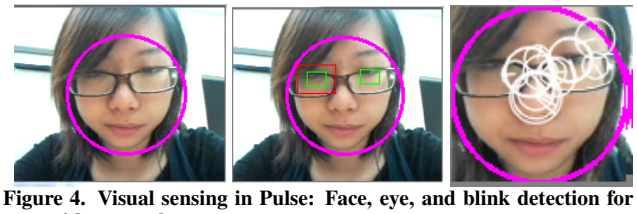


Figure 4. Visual sensing in Pulse: Face, eye, and blink detection for a user with spectacles.

Pulse draws out the following features: face position, eye position, lip position, face size, eye size, lip size, relative eye and lip position to the entire face, and the variation of each over the duration of the movie. We believe these features reasonably capture some of the reaction footprints useful for ratings [11].

Reaction Features: Acoustic

The Pulse video player activates the microphone and records ambient sounds while the user is watching the movie – this sound file is the input to our acoustic sensing sub-module. The key challenge is to separate the user's voice from the

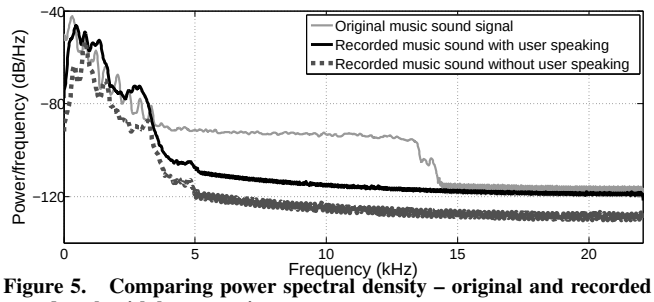


Figure 5. Comparing power spectral density – original and recorded soundtrack with human voice.

movie soundtrack, and then classify the user's voice as either laughter or speech. Since the movie soundtrack played on the tablet's speakers can be loud, separation is not straightforward. We describe Pulse's approaches as follows.

Voice Detection

Given that the human voice exhibits a well-defined footprint on the frequency band (bounded by $4kHz$), Pulse's first approach was to extract this band using a low pass filter and then perform separation [22]. However, the tablet already performs this filtering (to improve speech quality for phone calls). Figure 5 demonstrates this by comparing the Power Spectral Densities of the following: (1) the original movie soundtrack, (2) the sound of the movie recorded through the tablet microphone, and (3) the sound of the movie and human voice, recorded by the tablet microphone. Evidently, the recorded sounds drop sharply at around $4kHz$. At less than $4kHz$, the movie soundtrack with and without human voice are comparable, and therefore non-trivial to separate.

Pulse adopts two heuristic techniques to address the problem, namely (1) energy detection before and after speech enhancement and (2) per-frame spectral density comparison. We describe them here and show how they are applicable in different volume regimes.

(1) Energy Detection with Speech Enhancement:
Well established speech enhancement tools in literature can suppress noise and amplify the speech content in an acoustic signal. Pulse uses this to its advantage by measuring the (root mean square) signal energy before and after speech enhancement. For each frame, if the RMS energy diminishes considerably after speech enhancement, we regard this frame as noise. The simple intuition is that signals that contain speech will pass background noise suppression without being affected significantly; other noises should be reduced.

(2) Per-frame Spectral Density Comparison:
We observe that the power spectral density within $[0, 4]$ kHz is impacted by whether the user is speaking, laughing, or silent. In fact, the conversation from the movie can also impact this frequency regime. Figure 5 demonstrates an example case. Therefore, we compare the (per-frequency) amplitude of the recorded sound with the amplitude from the original soundtrack in each frame. If the amplitude of the recorded signal exceeds the soundtrack significantly, we deem that this video frame contains the user's voice.

Heuristic Selection based on Volume Regimes:
The two heuristics above perform differently depending on the volume of playback. Therefore, we use the energy detection heuristic when the playback volume is low and choose spectral density comparison for high-volume scenarios. Figure 6(a) reports their performance when the tablet volume is high – the dark horizontal lines in the top window represents the time windows when the user was actually speaking. The dark horizontal lines in the other two windows represent system detected speaking. Evidently, the second heuristic – per-frame spectral density comparison – exhibits better discriminative capabilities. This is because at high volumes, the human speech gets drowned by the movie soundtrack, and speech enhancement tools become unreliable. However, in low-volume cases, the soundtrack power is still low while the human voice is high, thereby allowing energy detection to identify the voice. Figure 6(b) shows this situation.

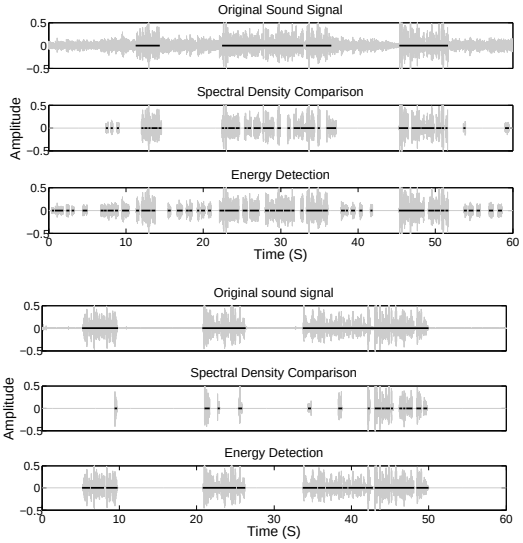


Figure 6. Comparison of voice detection. Top: High Volume; Bottom: Low Volume.

Laughter Detection

Pulse assumes that acoustic reactions during a movie are either speech or laughter – so, once human voice is detected, it needs to classified to one of the two categories. We use a support vector machine (SVM) and train it on the *Mel-Frequency Cepstral Coefficients* (MFCC) as the principle features. In sound processing, Mel-frequency cepstrum is a representation of the short-term power spectrum of a sound, and are commonly used in speech recognition [14]. To reduce false positives, Pulse performs a simple outlier detection. If a frame is suspected as laughter, but the 4 preceding and following frames are not, then these outlier frames are eliminated. Figure 7 reports results showing high accuracy and few false positives.

Reaction Features: Motion

Accelerometer and gyroscope readings are also likely to contain information about the user's reactions. The mean of the sensor readings over the playback of the entire movie may capture the typical holding position/orientation of the device, while variations may be indicators of potential events.

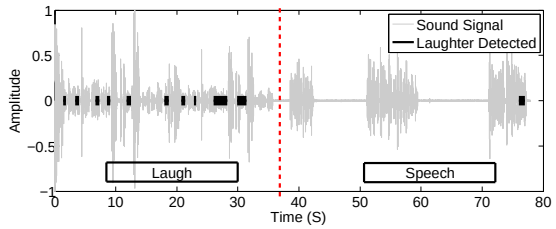


Figure 7. Discriminating laughter and speech from voice signals recorded by a tablet microphone.

Figure 8 shows an example where mean and variance (after some smoothing) appear well correlated to when users' ratings change. It is possible that users are performing micro-movements at the beginning or end of logical segments, and the sensors seem to be capturing them. Pulse attempts to gain insights from these motion signatures.

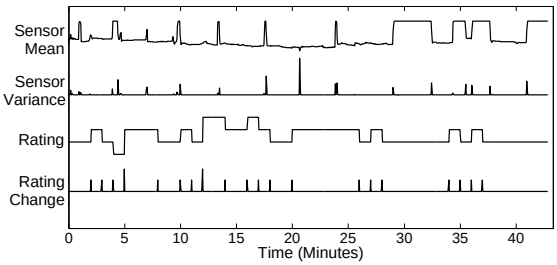


Figure 8. Motion correlates with rating changes.

Reaction Features: Touch Screen

Users tend to skip boring segments of a movie and, sometimes, may roll back to watch an interesting segment again. The information about how the user moved the slider can reveal the user's reactions for different movie segments. If Pulse observes a developing trend for skipping certain segments, or a trend in rolling back, the corresponding segments are assigned ratings proportionally (lower/higher).

SYSTEM DESIGN: CLR

This section describes the machine learning components in Pulse. The key goals are to model the sensed data and use the models to: (1) estimate segment ratings; (2) generate the final star rating from the segment ratings; (3) estimate semantic labels; (4) generate the tag-cloud from the semantic labels. To this end, Pulse requests multiple users to watch a movie, label different segments of the movie, and provide a final star rating as ground truth.

Ratings. *Segment ratings* are ratings for every short segment of the movie, necessary to compute the overall movie quality as well as to select enjoyable segments. A key challenge here is the ambiguity in how reaction features map to segment ratings. Laughter in a comedy movie may be a positive reaction, while laughter in a horror movie may mean the opposite. Some users may get excited and fidget in an intense scene, while others may watch it motionless. Pulse employs *Collaborative Filtering* and *Gaussian Process Regression* (GPR) to cope with such ambiguities (detailed later). To convert segment ratings to the *final rating*, Pulse uses a weighted averaging function.

Labels. *Semantic labels* are English labels assigned to each segment of the movie. CLR generates two types of such labels – *reaction labels* and *perception labels*. (1) *Reaction labels* are direct outcomes of reaction sensing, reflecting on the viewer's raw behavior while watching the movie (e.g., laugh, smile, focused, distracted, nervous, etc.). (2) *Perception labels* reflect on subtle emotions evoked by the corresponding scenes (e.g., funny, exciting, warm, etc.) While identifying reaction labels is straightforward, identifying perception labels is more challenging. Pulse employs a semi-supervised learning method combining Collaborative Filtering and SVM to predict perception labels. Then, Pulse aggregates all the predicted labels, counts their relative occurrences, and develops the tag-cloud description of the movie. The efficacy of prediction is quantified through cross-validation. The following subsection elaborates on the methodology and techniques.

Modeling and Prediction Challenges

We begin by describing our experimentation methodology, which will help explain the challenges we faced during modeling and prediction. Thereafter, we describe the solutions.

Experiment Methodology

To obtain labeled user data, we conducted a formative user study. We initially recruited 11 volunteers (4 females), aged 24–28. We provided volunteers with Android-based Samsung tablets pre-loaded with 6 movies (3 comedies, 2 dramas, and 1 horror), and asked them to watch only those movies they have not watched earlier. The volunteers were required to watch the movie using our Pulse video player, which activates and records sensor readings during playback. Because we needed data from natural settings, we let users watch movies at any place and time they chose; most users took the tablets home. We also provided a software tool that allowed users to rate the movie soon after they watched it. [2] This tool scans through the movie minute by minute (like fast-forwarding) and allows volunteers to rate segments on a scale from 1 to 5 (1 being "did not like", 5 being "liked"). Volunteers also labeled some segments with "perception" labels, indicating how they perceived the attributes of that segment. The perception labels were picked from a pre-defined set – some examples are "funny", "scary", "intense". Finally, volunteers were asked to provide a final (star) rating for the overall movie, again on a scale of 1 to 5.

Challenges

Pulse's goal is to model user behavior from the collected labeled data, and use this model to predict (1) segment ratings, (2) perception labels, and (3) the final (star) rating for each movie. Note that this is a high bar for Pulse – predicting human judgment, minute by minute, is quite difficult. The difficulty gets exacerbated by 3 types of heterogeneities, described next.

(1) Heterogeneity in users behavior: Some users watch movies attentively, while others are more fidgety. Such diversities are common among users, and particularly so when observed through the sensing dimensions. As a result, a naive universal model trained from a crowd of users is likely to

[2]To avoid affecting user's watching behavior, we asked users to provide ratings only after the finished watching the entire movie

fail in capturing useful behavioral signatures for any specific user. In fact, such a model may actually contain little information since the ambiguity from diverse user-behaviors may mask (or cancel out) all useful patterns. For example, if half of the users hold their devices still when they are watching a movie intensely, while the other half happen to hold their devices still when they feel bored, a generic model learned from all this information will not be able to use "stillness" as a discriminator between intensity and boredom. Thus, a good one-fit-all model may not exist. To confirm this, we created a regression model for estimating segment ratings using all available labeled data. Figure 9 plots the cross-validation results for the leave-one-video-out test, comparing the model's estimated segment ratings vs. the actual user ratings. The results show that the model's estimates fail to track the actual user ratings, and mostly converges on the *mean* rating of training data.

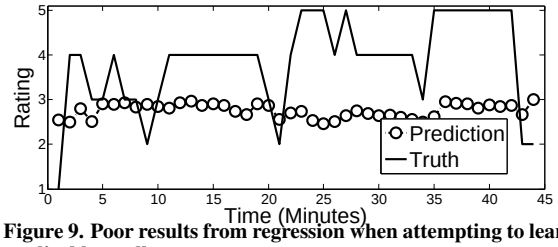


Figure 9. Poor results from regression when attempting to learn a model applicable to all users.

(2) Heterogeneity in environment factors: Even for the same user, her "sensed behavior" may differ from time to time due to different environmental factors. For instance, the behavior associated with watching a movie in the office may be substantially different from the behavior during a commute, which is again different from when at home. Figure 10 shows the gyroscope sensor data distribution from the same user watching two movies. The distribution clearly varies even for the same user, indicating that the way the user holds the device may not always be similar.

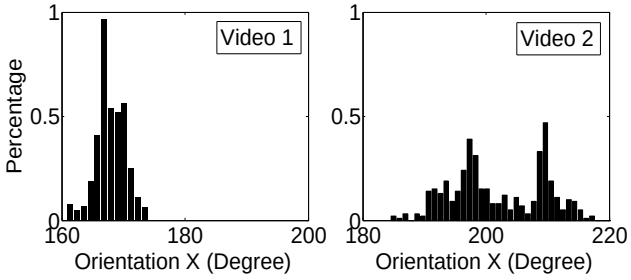


Figure 10. Orientation sensor data distribution

(3) Heterogeneity in user tastes: Finally, users may have different tastes, resulting in different ratings/labels given to the same movie scene. Some scenes may appear hilarious to one, and may not be so to another. Figure 11 shows the deviation in ratings given to the same scenes by 5 different users. Clearly, there is dissimilarity in taste.

Pulse's Learning Approach

The heterogeneities described above highlight the core challenge – we need to develop a model that will capture the unique taste/behavior of a user under different environments. One (brute force) approach would be to train a series of per-user models, each tailored to a specific viewing environment and for a specific genre of a movie. However, it is nearly impossible to enumerate all such environments, and worse, the user would have to provide ratings and labels for all combinations of movie genres and environments. This is impractical.

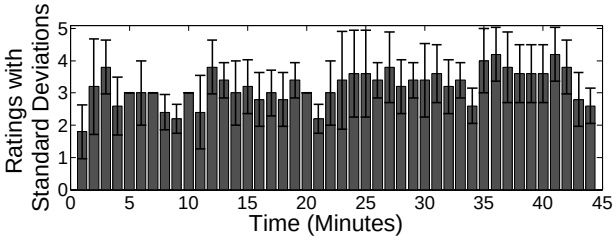


Figure 11. High Std. Dev. in ratings across users.

Pulse overcomes this problem by basing its solution on the following intuition. Although users exhibit heterogeneity overall, their reactions to certain parts of the movie are remarkably similar (or coherent). Therefore, we analyze the collective behavior of multiple users to extract only these coherent signals – i.e., segments for which most users exhibit agreement in their reactions. Similarly, for perception labels, Pulse also learns from segments on which most users agree. Collaborative filtering techniques [21] provide the ability to draw out these segments of somewhat "universal" agreement. We designed two separate semi-supervised learning methods – one for segment ratings and another for perception labels. For segment ratings, we combine *collaborative filtering* with *Gaussian Process Regression* (GPR). Using GPR, data from multiple sensing dimensions can be easily combined using the co-training procedure. For perception labels, we combine *collaborative filtering* with support vector machines (SVM) since this is essentially a multi-class classification problem.

When a new user watches a movie, Pulse uses the sensed data from *only* the "universally agreed" segments to train a customized model, which is then used to predict the ratings and labels of the rest of the user's segments. In other words, Pulse bootstraps using ratings that are agreeable in general, and by learning how the new user's sensing data correlates with these agreeable ratings, Pulse learns the user's "idiosyncrasies" (which is the most difficult aspects of automatic content rating). Now, with knowledge of these idiosyncrasies, Pulse can "extrapolate" to other segments of the movie (that users did not agree upon), and predict the ratings for this specific user [16]. Figure 12 illustrates our method. From the ratings of users A, B, and C, Pulse learns that minute 1 is intense (I) and minute 5 is boring (B). Then, when user D watches the movie, his sensor readings during the first and the fifth minutes are used as the training data to create a personalized model. This model is then used to predict the 2^{nd}, 3^{rd}, and 4^{th} segment ratings.

Figure 13 shows that Pulse's approach works reasonably well, with Pulse's estimated ratings tracking the actual user ratings. We will discuss additional results on segment rating and label prediction in the evaluation section.

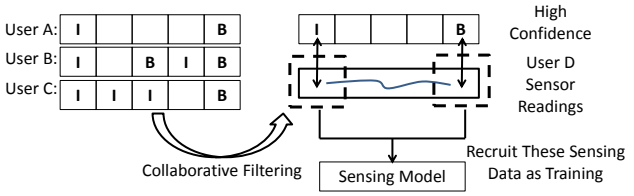


Figure 12. Pulse learns a custom model from high-confidence segments.

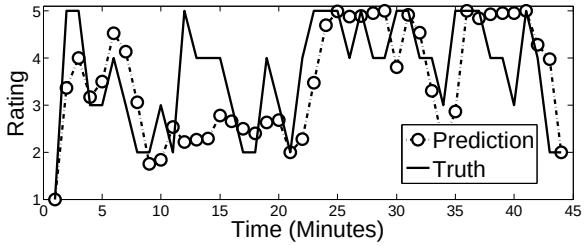


Figure 13. Collaborative filtering and GPR improve prediction – circles are the Pulse's predictions

Besides coping with inherent heterogeneity of users, we observed additional challenges emerging from (1) *time-scale of ratings* and (2) *sparsity of labels*. The first problem arises from the mismatch between the time-scale of sensed reactions (a laughter lasts a few seconds) and the time-scale of human ratings (one for each minute). As a result, the human labels we obtain are not necessary labeling the specific sensor pattern, but rather an aggregate of useful and useless patterns over the entire minute. This naturally raises the difficulty for learning the appropriate signatures. The situation is similar for labels. It is unclear exactly which part within the 1-minute segment was labeled as hilarious, since the entire minute may include both "hilarious" and "non-hilarious" sensor signals. To cope with this, we assume that each 3 second window in the sensing data has the label of the corresponding minute. In our prediction, once Pulse yields a rating/label for each 3-second entry, we aggregate them back to the minute granularity, allowing us to compute both prediction accuracy and false positives.

The second problem relates to how labels gathered in each movie are sparse (volunteers did not label each segment, but opted to label only scenes that seemed worthy of labeling). As a result, we found 65.9% of the segments unlabeled. This warrants careful adjustment of the SVM's weighting parameters to assign more importance to the positive samples – otherwise SVM may classify all segments as "none of the valid labels", and appear to achieve high accuracy (since much of the data indeed has no valid label). Precisely recognizing and classifying the few minutes of the labeled segments, from thousands of minutes of recordings, is an ambitious task. We designed under these constraints while ensuring we do not over-fit – the next section reports on the results.

EVALUATION

In this section, we demonstrate the feasibility of predicting (1) segment ratings, (2) final ratings, and (3) semantic labels, through multi-dimensional sensing.

Metrics

We adopt three measures commonly used in information retrieval, namely, *Precision*, *Recall*, and *Fallout*. These metrics essentially are methods to compute overlaps (and non-overlaps) between two sets of items. Consider the case of segment rating. One set is the set of movie segments that the user truly enjoyed (i.e., segments manually rated as 4 or 5) – we call this the *Human Selected* set. The other set contains segments that Pulse believes the user enjoyed – called the *Pulse Selected* set. Then, the 3 metrics can be defined as:

$$Precision = \frac{|\{\text{Human Selected} \cap \text{Pulse Selected}\}|}{|\{\text{Pulse Selected}\}|}$$

$$Recall = \frac{|\{\text{Human Selected} \cap \text{Pulse Selected}\}|}{|\{\text{Human Selected}\}|}$$

$$Fall-out = \frac{|\{\text{Non-Relevant} \cap \text{Pulse Selected}\}|}{|\{\text{Non-Relevant}\}|}$$

Higher values of precision and recall are better; the converse for fallout. These metrics apply for the semantic labels as well, where one set is provided by humans, and the other generated by Pulse.

Summary of Results

1. **Segment Rating:** Predicted segment ratings closely follow users' segment ratings, with an average error of 0.7 on a 5-point scale. This is a 40% improvement over estimations based on only distribution or collaborative filtering (the improvement is more pronounced in terms of *recall*). More importantly, Pulse is able to capture enjoyable segments with an average precision of 71%, an average recall of 63%, and a minor fallout of 9%.
2. **Final Rating:** Pulse's overall star rating demonstrates an average error of 0.46 in the 5 point scale.
3. **Label quality:** On average, Pulse covers 45% of the perception labels with a minor average fallout of 4%. We also observe an order of magnitude improvement over a pure SVM-based approach and modest gains over collaborative filtering. The reaction labels capture the audience's reactions well. The tag-clouds were received with enthusiasm (a qualitative feedback). Detailed results follow.

Performance of Segment Rating

To quantify Pulse's accuracy at predicting segment ratings on the 5-point scale, we compared the results of four prediction algorithms: Random, Collaborative Average, Collaborative High Confidence and Pulse. Random predicts the scores randomly. Collaborative average uses the average of all user's segment ratings as the prediction. Collaborative High Confidence assigns the average user's score for only those segments that were given consistent ratings by most users and assigns the scale's average score (3 in our case) to other segments. Finally, Pulse, uses collaborative filtering results as a starting point and exploits sensing data as described in previous sections to provide a more accurate prediction. Figure 14 plots the mean prediction errors of the four algorithms as black bars. Pulse outperforms other algorithms, achieving 0.7 mean error. This result shows the value that sensing data may bring to automatic segment rating.

Often, people are interested in finding good movies (rated above 3) and may not care whether a movie is rated 1.4 or

2.4. This observation can be used to optimize Pulse further by treating ratings from 1 to 3 as the same. In doing so, we are essentially reducing the resolution of expressing that a movie is not worth watching to a single score. The mean prediction errors of the four algorithms, when the rating score is reduced to a 3-point scale, are shown as grey bars in Figure 14. Here, Pulse again outperforms other algorithms, achieving 0.25 mean error.

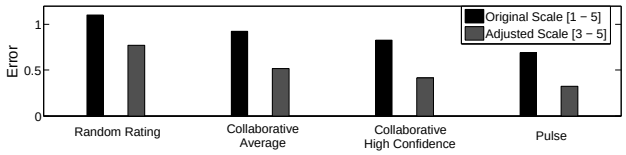


Figure 14. Mean Pulse segment rating error.

Segment Selection.

Pulse selects the "enjoyable" segments, i.e., ones rated as 4 and above, to generate a highlights of the movie. To evaluate whether Pulse's selection matches with the user's, we evaluate Pulse using precision, recall, and fallout. Figure 15 shows the average precision ranges from 57% to 80%, an average recall of 63%, and a minor fallout usually less than 10%. Pulse performed well on 2 comedy and 2 dramas, corresponding to the first four bars in each group. The performance was weaker in the remaining 2 movies (1 comedy and 1 horror).

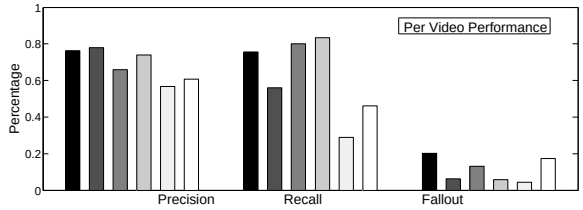


Figure 15. Precision/Recall/Fallout per video

Figure 16 shows the average per-user performance. Except for one outlier (the second user), the precision is above 50% with all recalls above 50%. Fallout ranges from 0 to 19%. Given the sparse labels we have, the accuracy we believe is reasonable – on average Pulse creates less than one false positive every time it includes five true positives. One may observe that the second user might be characterized as "picky" – the low precision, reasonable recall, and small fallout, suggest that she rarely assigns high scores. We note that all the above selections are personalized; a good segment for one user may be boring to another and Pulse can identify these inter-personal differences.

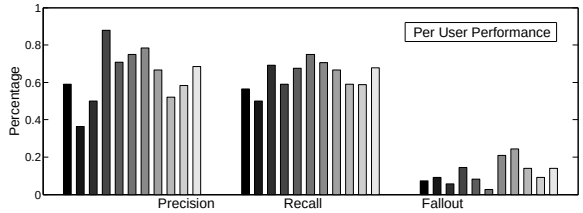


Figure 16. Precision/Recall/Fallout per user

Figure 17 illustrates the break-up of contributions from collaborative filtering and sensing. The four bars show the number of true positives, total number of positive samples (segments with ratings of 4 or 5), false positives, and total number of negative samples (segments with rating 1 to 3), respectively. As the figure illustrates, the contribution from sensing is substantial.

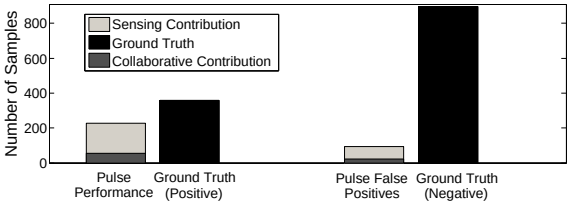
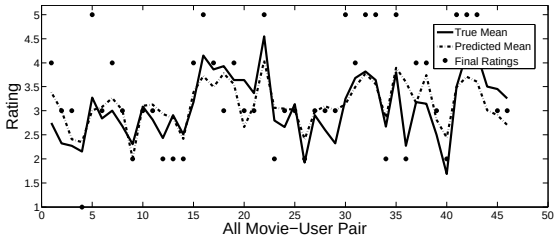


Figure 17. Break-up of contributions.

Performance of Final "Star" Rating

Pulse generates final ratings by thresholding the mean scores of per-minute segment ratings. This thresholding function essentially tries to learn how users map their mean scores for each segment to the final score. Figure 18(a) shows the means of predicted and true segment ratings (dashed and solid lines), as well as the true final rating. Pulse tracks the user's mean rating well. We observed that users are often conservative in rating the movie segments, but more generous with the final rating. Figure 18(b) shows Pulse's prediction of final ratings using a confusion matrix. Higher values concentrate around the diagonal, indicating desired performance. We are aware that we may have over-fitted our data with the thresholds, and intend to investigate this more carefully in future.



Pulse \ Truth	1	2	3	4	5
1	0	1	0	0	0
2	0	4	2	0	1
3	0	1	17	0	1
4	0	0	2	5	2
5	0	0	2	1	7

Figure 18. (a) Mean segment ratings and corresponding users' final ratings. (b) Confusion matrix.

Performance of Label Quality

Pulse associates semantic labels to each movie segment and eventually generates a tag cloud for the entire movie. This section evaluates the efficacy to predict labels. Recall that our semantic labels consist of *reaction labels* and *perception labels*; we evaluate them separately. As a ground truth, we intend to know the user's reactions and perceptions at every time point. However, providing this information (while the user is watching the movie) would have interfered with their viewing experience. Therefore, we asked the users to provide the perception labels to each movie segment after watching the movie. For reaction labels, we recruited two volunteers to view the video recording from the tablet camera, and label the viewer's reactions – we used this as ground truth [7].

Reaction Label Quality

Reaction labels capture users' actions while watching a movie (e.g., laugh, smile, etc.). The (limited) vocabulary is shown in Table 1. Figure 19 shows the comparison between Pulse's prediction and the ground truth – the gray portion is ground truth while the black dots denote when Pulse detects the corresponding labels. Although Pulse sometimes mislabels on a per-second granularity, the general time frame and weight of each label is reasonably well captured.

Table 1. Label Vocabulary

Label Category	Vocabulary
Perception	Funny, Intense, Warm
Reaction	Laugh, Smile, Shaking, Focused, Distracted, Speaking

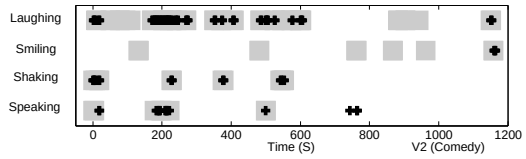


Figure 19. Reaction label prediction vs. groundtruth

Perception Label Quality

Perception labels represent a viewer's perception of each movie segment (e.g., funny, warm, intense). Figure 20 shows the performance of perception label prediction for each label, averaged over all users. These labels are hard to predict because (1) their corresponding behaviors can be subtle and implicit; (2) users provided these labels for few segments. Our performance is proportionally weaker: average precision is 50%, recall is 35%; however, fallout is satisfactory: 4%.

Figure 21 compares the performance between pure-SVM (using cross validation), collaborative filtering, and Pulse. From top to bottom, the figures show precision, recall, and fallout, respectively. Pulse demonstrates substantial improvement over SVM alone, but is comparable to collaborative filtering.

Tag Cloud and User Feedback

We attempted to visually summaries the results of Pulse using a tag cloud similar to Figure 1. The terms used within the tag-cloud combine perception and reaction labels, each weighted by its normalized occurrence frequency. We informally asked users who watched the episode (using Pulse) to comment on this tag-cloud. The feedback was resonantly enthusiastic, with comments like "very cool", and "certainly useful information with zero extra burden". Some users correctly pointed out that "a richer tag set is needed".

Power Consumption

We measured the power consumption of Pulse on the Samsung tablets and Nexus S smartphones, using the Monsoon Power monitor. Figure 22 compares Pulse's performance with conventional media players – with no active sensors. Given the high playback and display power on tablets, Pulse-based sensing adds only 16% more energy. The energy burden is higher on smartphones, and would call for duty cycling the sensors, perhaps to only sense the decisive segments. We leave this to future work.

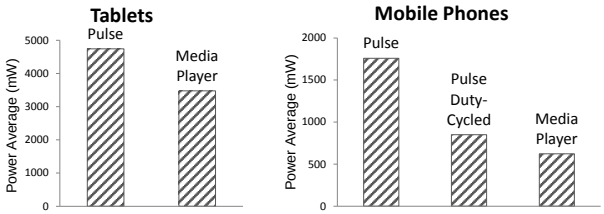


Figure 22. Power consumption comparison.

RELATED WORK

Pulse builds on work that falls roughly in two areas: activity inferencing and multimedia annotation. We discuss some of the related work here.

Activity Inference: The large number of sensors on mobile devices have been leveraged as a rich sensing platform. Accelerometers are useful beyond motion [1]; microphones often effective in detecting environments [13], and user's reaction [2, 10, 25]; front-facing cameras are valuable towards face/eye tracking in real-time video streams. Combined with machine learning and inferencing, these platforms are lending themselves to intent and context recognition [**?**]. The future is poised for more activities along these directions. Of course, such forms of continuous sensing causes substantial power drain. Existing proposals include offloading to the cloud [6] or duty cycling techniques [24]. In future, efforts such as Little Rock could offload sensing to DSP chips, allowing the CPU to sleep [18].

Multimedia Annotation: A powerful technique to annotating multimedia is to aggregate sensor data across multiple devices as a way of supersampling [8]. TagSense is one example, using sensor data from multiple devices, to annotate images [19]. Pulse uses a similar approach, but asynchronously aggregated across users. Recommender systems often annotate items using a set of known attributes – this maintains calibration across users, while capturing diversity in opinions [26]. Though these results tend to yield diverse results, they are resource intensive, require lots of time [20]. We hope to address some of these issues with a sensor based approach, available free in today's devices.

CONCLUSION

Advances in personal sensing and machine learning are empowering machines to better understand human behavior. This paper guides this opportunity into an application that automatically rates content on behalf of human users. The core idea is to leverage device sensors, such as cameras, microphones, accelerometers, and gyroscopes, to sense qualitative human reactions while she is watching a video;learn how these qualitative reactions translate to a quantitative value; and visualize these learnings in an easy-to-read format. Thus, when using our system, a movie automatically gets tagged not only by a conventional star rating, but also with a tag-cloud of user reactions, as well as highlights of the movie for different emotions.

At this stage, Pulse is still a prototype with many limitations. On the technical side, the current label vocabulary

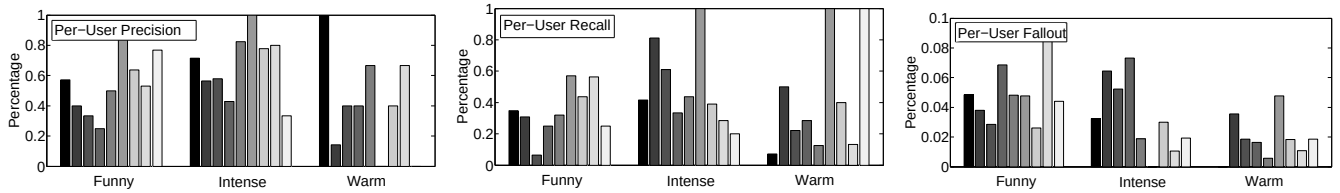


Figure 20. Performance for each label (averaged across users).

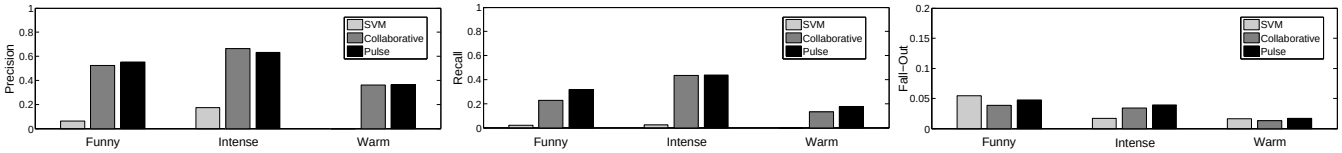


Figure 21. Performance comparison between SVM, collaborative filtering and our method (Pulse).

is still limited. We intend to explore additional optimizations in machine learning to improve performance, while taking advantage of more sensors that enter the tablet platform. Moreover, the current implementation does not focus on scenarios where multiple people watch movies together on mobile devices. New designs may be needed to accomodate such situations. On the social side, pulse may raise privacy concerns especially for exporting information to the cloud. Though we do not have a clear solution to this problem yet, Pulse should certainly place most of its functionalities locally on the user's device and potentially only needs to upload ratings in the end. Different methods for data fusion can also help anonymize the data.

With these limitations, we still believe there is value in building a sensing-based automatic rating system. With the universe of content growing at a rapid pace, the need for associating meta data to content will become increasingly relevant. Pulse is an early attempt towards this goal, with direct applications in recommendations systems and information retrieval [4].

REFERENCES

1. L. Bao et al. Activity recognition from user-annotated acceleration data. *Pervasive Computing*, 2004.
2. X. Bao and R. Roy Choudhury. Movi: mobile phone based video highlights via collaborative sensing. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010.
3. H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, 2006.
4. I. Cantador, M. Fernández, D. Vallet, P. Castells, J. Picault, and M. Ribière. A multi-purpose ontology-based approach for personalised content filtering and retrieval. *Advances in Semantic Media Adaptation and Personalization*, pages 25–51, 2008.
5. M. Cherubini, R. De Oliveira, N. Oliver, and C. Ferran. Gaze and gestures in telepresence: multimodality, embodiment, and roles of collaboration. 2010.
6. E. Cuervo, A. Balasubramanian, et al. MAUI: Making Smartphones Last Longer with Code Offload. In *ACM MobiSys*, 2010.
7. P. Ekman and W. Friesen. *Unmasking the face: A guide to recognizing emotions from facial clues*. 2003.
8. R. Honicky, E. Brewer, E. Paulos, and R. White. N-smarts: networked suite of mobile atmospheric real-time sensors. In *ACM NSDR*, 2008.
9. C. Jacob and S. Steglich. Conbrowse-contextual content browsing. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1–5. IEEE, 2010.
10. L. Kennedy and D. Ellis. Laughter detection in meetings. In *NIST Meeting Recognition Workshop*, 2004.
11. P. Lang, M. Greenwald, M. Bradley, and A. Hamm. Looking at pictures: Affective, facial, visceral, and behavioral reactions. *Psychophysiology*, 1993.
12. I. Li, J. Nichols, T. Lau, C. Drews, and A. Cypher. Here's what i did: sharing and reusing web activity with actionshot. In *ACM CHI*, 2010.
13. H. Lu, W. Pan, et al. SoundSense: scalable sound sensing for people-centric applications on mobile phones. In *ACM MobiSys*, 2009.
14. M. F. McKinney and J. Breebaart. Features for audio and music classification. In *ISMIR*, 2003.
15. T. Morris, P. Blenkhorn, and F. Zaidi. Blink detection for real-time eye tracking. *JNCA*, 2002.
16. T. Ouyang and Y. Li. Bootstrapping personal gesture shortcuts with the wisdom of the crowd and handwriting recognition. In *ACM CHI*, 2012.
17. M. Paolucci, G. Broll, J. Hamard, E. Rukzio, M. Wagner, and A. Schmidt. Bringing semantic services to real-world objects. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2008.
18. B. Priyantha, D. Lymberopoulos, and J. Liu. Littlerock: Enabling energy-efficient continuous sensing on mobile phones. *IEEE Pervasive Computing*, 2011.
19. C. o. Qin. Tagsense: a smartphone-based approach to automatic image tagging. In *ACM MobiSys*, 2011.
20. L. Rosenfeld and P. Morville. *Information Architecture for the World Wide Web*. O, 1998.
21. B. Sarwar et al. Item-based collaborative filtering recommendation algorithms. In *ACM WWW*, 2001.
22. J. Sohn, N. Kim, and W. Sung. A statistical model-based voice activity detection. *IEEE SPL*, 1999.
23. J. Teevan, E. Cutrell, D. Fisher, and Others. Visual snippets: summarizing web pages for search and revisitation. In *ACM CHI*, 2009.
24. Y. Wang, J. Lin, M. Annavaram, et al. A framework of energy efficient mobile sensing for automatic user state recognition. In *ACM MobiSys*, 2009.
25. X. Yang, C.-W. You, H. Lu, M. Lin, N. D. Lane, and A. T. Campbell. Visage: A face interpretation engine for smartphone applications. In *Mobile Computing, Applications, and Services*. Springer, 2013.
26. C. Yu, L. Lakshmanan, and S. Amer-Yahia. It takes variety to make a world: diversification in recommender systems. In *ACM EDBT*, 2009.