Collaborative Real-Time Speaker Identification for Wearable Systems

Mirco Rossi
Wearable Computing Lab., ETH Zurich, Switzerland
mrossi@ife.ee.ethz.ch,
http://www.wearable.ethz.ch

Oliver Amft
Signal Processing Systems, TU Eindhoven, The Netherlands
amft@tue.nl,
http://w3.ele.tue.nl/en/sps

Martin Kusserow
Wearable Computing Lab., ETH Zurich, Switzerland
kusserow@ife.ee.ethz.ch,
http://www.wearable.ethz.ch

Gerhard Tröster
Wearable Computing Lab., ETH Zurich, Switzerland
troester@ife.ee.ethz.ch,
http://www.wearable.ethz.ch

***Abstract*—We present an unsupervised speaker identification system for personal annotations of conversations and meetings. The system dynamically learns new speakers and recognizes already known speakers using one audio channel and speech-independent modeling. Multiple personal systems could collaborate in robust unsupervised speaker identification and online learning. The system was optimized for real-time operation on a DSP system that can be worn during daily activities.**

The system was evaluated on the freely available 24-speaker Augmented Multiparty Interaction dataset. For 5 s recognition time, the system achieves 81% recognition rate. Collaboration between four identification systems resulted in a performance increase of up to 17%, however even two collaborating systems yield an performance improvement. A prototypical wearable DSP implementation could continuously operate for more than 8 hours from a 4.1 Ah battery.

I. Introduction

Identifying a speaker during meetings or conversations allows to annotate communication, determine social relations, capture interesting moments in daily life. Moreover, it can enable many further applications, such as recognising speech and analysing interactions. While stationary systems are used to recognize speakers in meeting rooms, mobile and wearable systems can enable speaker annotations without being constrained to particular locations. Systems such as the body-worn Sociometer [1] revealed the potential of identifying speakers in several applications. Moreover, a personal annotation of social contacts and conversations allows to search, select, and retrieve information from databases that could potentially capture all audio and visual information perceived during daily life [2]. Several projects aim to capture this multimodal information, including MyLifeBits [3] and Interest-Based Life Logging [4].

A wearable speaker identification system requires, however, to cope with a number of constraints that have not been adequately considered to date. Firstly, the system must be personal and autonomous, not dependent on tight collaboration with systems of others, since ad-hoc conversations may involve individuals without a compatible system. Secondly, the system must be able to detect and learn new speakers as conversations may involve new collaborators, friends, and strangers. Finally, the processing on a wearable system must be performed efficiently and in real-time to provide immediate response options.

We present in this work the design and implementation of an unsupervised speaker identification for wearable systems. In our approach, a speaker is modeled dynamically from voice data and subsequently identified. While in this way our system can be used in standalone mode, without collaboration, we particularly foresee a collaboration feature, to jointly decide whether a speaker is known. As the system learns new speakers without supervision, the collaboration can moreover help to improve system robustness.

In particular this work makes the following contributions:

1) We present an unsupervised, text-independent speaker identification system using only one microphone. We study its performance using a freely available dataset that had not been investigated for speaker identification before. The results demonstrate that our system can cope even with large meeting sizes of 24 speakers.
2) We evaluate the collaboration of multiple personal systems in six meetings of four speakers each, and discuss design choices for the collaborative setting. Our results confirm clear benefits for unsupervised systems to collaborate during the identification of new speakers. Performance is directly related to the number of collaborating systems.
3) We discuss the real-time system design with regard to constraints in wearable systems. To this end, we present and evaluate a complete implementation and deployment on a DSP platform prototype, show that the system can operate in real-time, and a wearable identification system can be built.

Section II reviews related works with regard to the lack of wearable system solutions for speaker identification. Section III presents our system design approach, which is analyzed in standalone mode (Section IV) and in the collaborative setting (Section V). Section VI analyses a wearable device deployment. Finally, Section VII summarizes the work.

978-1-4244-5328-3/09/$25.00 ©2009 IEEE

II. Related Work

Automated speaker identification that could enable monitoring social interactions has been investigated from both application and technical perspectives for several years. These systems are either stationary installed in rooms to annotate meetings or - as it is aimed at in this work - the system can be worn as a daily personal accessory. The latter case allows to identify interaction partners, annotate conversations, and build a personal diary of social activities.

Several smart meeting rooms have been proposed, such as at Dalle Molle Institute [5] and at Berkely [6]. These rooms are equipped with microphone arrays, typically at the table center and lapel microphones for each participant. To identify a speaking person, the lapel microphone having the highest input signal energy is chosen. The approaches are by far not restricted to monitoring using acoustic means alone. Approaches have been made to combine sensor information from multiple sources, including vision and audio [7], [8]. An extensive review of these attempts is beyond the scope of this section however. As the systems are stationary their use is restricted to meetings and conversations held in the particular room. Wearable systems can capture conversations as they happen outside of these smart spaces.

An initial wearable system is the Sociometer developed by Choudhury and Pentland in 2002 [1]. This system can be attached to a person's shoulder. It includes an IR transmitter and receiver to communicate with persons nearby. A microphone was used to separate speech from non-speech segments. The Sociometer is used for different kinds of social network analysis and organizational behavior, including analysis of social behavior in a research group [9], modeling of group discussion dynamics [10], and prediction of shopper's interest [11]. As the speaker identification with the Sociometer is achieved through IR communication, only individuals wearing this system can be recognized.

The works cited above impressively demonstrate the broad application potential of speaker identification. Nevertheless, these systems are limited by the prior knowledge and configuration required to operate them, such as the number and identity of speakers, and their location. Since those approaches did not use a speaker modeling, the monitoring devices depend essentially on exchanging information on the current speaker. However, the availability of speaker models would allow to use identification system while roaming between locations and continuously identifying speakers that have been modeled before. Subsequently, adding the capability to detect a new speaker allows to learn speakers dynamically and unsupervised.

Several procedures intended for unsupervised speaker recognition have been developed. Anliker [12] proposed an online speaker separation and tracking system based on blind source separation. The task of identifying speakers is largely facilitated by source separation, for which reason it had been used in many works. However, at least two microphones are required to perform a source separation. This property imposes extended processing and power consumption requirements, which contradict to the viability of a wearable system implementation.

Other algorithms that operate without speaker separation and, therefore, need one microphone only, have been proposed by Charlet [13], Lu and Zhang [14], Kwon and Narayanan [15], and Lilt and Kubala [16]. These works utilized different speech features including linear predictive cepstrum coefficients (LPCC), mel-frequency cepstrum coefficients (MFCC), and line spectrum pair (LSP). For modeling speaker these systems typically use Gaussian Mixture Models (GMMs). It is known that GMMs may not be stably derived from small training data sizes. For unsupervised operation during conversations, however, only small data amounts may be available to learn a new speaker online. In contrast, Vector Quantization (VQ) handles small training data sizes more effectively [17]. Nishida and Kawahara [18] combined GMM and VQ speaker modeling to overcome this training problem.

None of these single microphone systems investigated the benefit of collaborative speaker identification. Nonetheless, it can be expected that the reduced performance due to design choices for online and unsupervised operation could be compensated by ad-hoc collaboration of speaker identification systems. Anliker [12] addressed the case of collaborative information fusion with two and three systems performing source separation and speaker identification. However, the results did not show a clear improvement of the collaboration when compared to a stand alone system. This observation may be attributed to the specific source separation and identification procedure considered in his work.

While implementations of *speech recognition* systems can be found in the literature, e.g. [19], [20], only a small number of works address the real-time implementation of *speaker recognition* systems. E.g. Anliker [12] aimed at an online system, however, the solution was not adequate for real-time operation due to the processing complexity. Furthermore, Liang et al. [21] described the design and implementation of a pre-trained speaker recognition system using FPGAs and DSPs. McCowan and Moore [22] designed an algorithm for FPGA hardware as well. This system was capable to localize and segment a small number of speakers in the vicinity using a microphone array.

III. Unsupervised speaker identification system

Our unsupervised speaker identification system incorporates two operations: recognition and online learning. For recognition, the system identifies a speaker by matching phoneme models from a speaker database to the continuous audio stream. In addition, the system can identify new speakers that do not sufficiently match with the existing model database. These new speakers are automatically added

to the system using online learning. Figure 1 illustrates main components of our identification system for both operations.

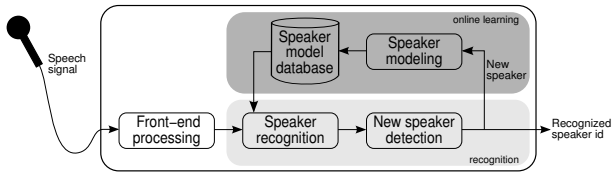

Figure 1. Concept of unsupervised speaker identification system supporting recognition and online learning operations.

Rejecting an utterance that does not belong to any existing speaker model in the database, is a core design element of an unsupervised speaker identification system. We study here three variants for discriminating between known and unknown speakers that yield different identification performance. In addition, these variants influence collaboration between systems as further analyzed in Section V.

A. Front-end audio processing

Front-end processing targets to extract speaker-dependent and text-independent features from the audio signal using pre-processing, feature extraction, and channel compensation.

Most speaker related information in speech is inside a frequency band of 0-4 kHz. To minimize system complexity, we chose an 8 kHz sampling and 16 bit quantization rate. During pre-processing we filtered the raw audio signal with a transfer function $H(z) = 1 - \alpha z^{-1}$, where $\alpha = 0.97$. This filter emphasizes higher frequencies bands and removes speaker independent glottal effects [23].

Subsequent to pre-processing, a feature vector $\mathbf{x} = (x_1, \ldots, x_N)$ was derived from the audio signal. Two frequently used voice modeling approaches have been evaluated: linear predictive cepstrum coefficients (LPCC) and mel-frequency cepstrum coefficients (MFCC). Both concepts capture phonetic speaker properties. Since phonemes are speech segment of about 20-30 ms [23] we used a sliding window with 30 ms length and 20 ms step size to derive these features.

We utilized a linear channel compensation approach to minimize device-dependent effects. We used here the short-term cepstral mean subtraction [23]. However, we applied it on sliding windows: $\widetilde{\mathbf{x}}^t = \mathbf{x}^t - \overline{\mathbf{x}}^t$, with $\overline{\mathbf{x}}^t = \frac{1}{T}\sum_{j=t-T}^{t} \mathbf{x}^j$. This corresponds to subtracting the feature vector average of the last T features from feature vector $\mathbf{x}^t$ generated at time t. T was set to the recognition epoch size, as described below.

B. Speaker modeling and matching

During recognition mode, feature vectors of a speech segment are compared with stored database models to identify a known speaker. While GMMs can outperform VQ in text-independent speaker recognition performance [24], they require a complex model learning phase using the expectation maximization algorithm. However, VQ can outperform GMMs at small amounts of training data and when fast modeling time is required [25]. With regard to our real-time speaker recognition and learning system, we chose VQ as a short training time was desired. In addition, algorithm complexity is a critical concern for the DSP implementation.

With VQ, speaker models are formed by clustering a set of training feature vectors $\{\mathbf{x}_i\}_{i=1}^{L}$ in K non-overlapping clusters. Each cluster is represented by a code vector $\mathbf{c}_i$ of the cluster centroid. A set of code vectors (codebook) $C = \{\mathbf{c}_i\}_{i=1}^{K}$ serves as speaker model during recognition.

Several clustering algorithms can be used to derive a codebook, however, with marginal performance differences [26]. For this work we used the Generalized Lloyd algorithm (GLA) [27], which has low complexity compared to the other known algorithms. The modeling procedure parameters (codebook size K, number of feature vectors L) determine system complexity. These have been further evaluated in Section IV.

To identify a speaker during recognition we used the quantization distortion between a set of test feature vectors $X = \{\mathbf{x}_i\}_{i=1}^{M}$ and a speaker codebook C. The quantization distortion d_q of $\mathbf{x}_i$ with respect to C was defined as $d_q(\mathbf{x}_i, C) = \min_{\mathbf{c}_j \in C} d(\mathbf{x}_i, \mathbf{c}_i)$. Here $d(\mathbf{x}_i, \mathbf{c}_i)$ is a distance measure defined between two feature vectors for which we used the Euclidean distance. The average of all individual distortions was used as matching metric of a speaker model during recognition (Eq. 1).

$$D(X, C) = \frac{1}{M}\sum_{i=1}^{M} d_q(\mathbf{x}_i, C) \tag{1}$$

Speaker identification is done by calculating the mean distortion of every code of every codebook stored in the system's database. The speaker is then identified with the best matching speaker model C_{best}, which is the codebook with the smallest D.

The recognition performance is proportional to the length of a recognition epoch, hence, the number of feature vectors M considered for each recognition. Nevertheless, long epochs will prevent the system to identify rapid speaker changes in conversation and meetings. We evaluate M in Section IV.

C. New speaker detection

In unsupervised open-set operation, a speaker may be initially unknown to the system. Consequently, we developed a procedure that determines whether the analyzed observation belongs to a known or unknown speaker. For this purpose we defined a decision function shown in Eq. 2.

$$f_d(X, C_{best}) = \begin{cases} 1, & \text{if score}(X, C_{best}) \geq \Delta \\ 0, & \text{else} \end{cases} \tag{2}$$

X is the set of feature vectors of the tested person, C_{best} is the best matching speaker model, score(X, C_{best}) is a score function, and Δ is a threshold. If the score of a tested speaker is equal or larger than Δ, the tested speaker is classified as the best matching speaker C_{best}. However, if score(X, C_{best}) is smaller than Δ, the observation will be classified as unknown speaker.

We analyzed three variants for the score function, one of these is the impostor cohort normalization (ICN) [28], [29]. The two alternatives were developed for this work and compared to ICN in Section IV.

1) The score function corresponds to the negated best matching speaker model distortion (compare Eq. 1):

$$\text{score}_{\text{score}}(X, C_{best}) = -\text{D}(X, C_{\text{best}}), \qquad (3)$$

where C_{best} is the model of speaker C_{best}.

2) The score function corresponds to the negated $\text{D}(X, C)$, normalized by distortions of a set of other speaker models ("impostor speakers"):

$$\text{score}_{\text{ICN}}(X, C_{best}) = -\frac{\text{D}(X, C_{best}) - \mu_I}{\sigma_I}, \qquad (4)$$

with mean μ_I and standard derivation σ_I of the impostor distortions. This score function corresponds to the impostor cohort normalization (ICN).

3) The score function corresponds to the feature vectors in X with minimum distance to the best matching speaker model C_{best}, normalized by the total number of feature vectors in X:

$$\text{score}_{\text{win}}(X, C_{best}) = \frac{N_{win}}{N_{all}}. \qquad (5)$$

D. Online learning procedure

When an unknown speaker had been detected, as described in Section III-C above, this new speaker is enrolled in the system using online learning. All feature vectors that have been collected during recognition and new speaker detection, are reused to derive the new speaker model.

For a real-time system, timing constraints exist between recognition, new speaker detection, and online learning. Since an identified new speaker is instantly enrolled, the new speaker detection epoch defines the training set size. With regard to our real-time system, training time and recognition time should be as short as possible. However, learning a new model needs more data than the recognition task. Thus, speaker recognition can be performed in shorter epochs than new speaker detection. Figure 2 illustrates theses timing relations.

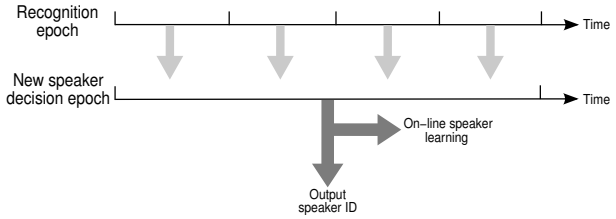

Figure 2. Illustration of timing relations between recognition, new speaker detection, and online learning in the real-time system.

IV. Standalone system evaluation

To confirm the robust system operation and to select parameters for efficient online performance, we initially evaluated the system in standalone operation. In particular, we analyzed system performance for LPCC and MFCC with different coefficient counts, the number of centroids to model speakers, the effect of training and recognition times, and the three score metrics for our new speaker detection. These parameters influence complexity and performance of the resulting system regarding both online learning and recognition, and hence determine viability of real-time operation on a DSP system.

A. AMI speaker corpus and evaluation procedure

To ensure reproducibility of all analysis results, we selected the freely available Augmented Multiparty Interaction (AMI) corpus [30] for our evaluation. This dataset provides more than 200 individual English speakers and contains ∼100 hours of conversation/meeting scenes recorded from ambient far-field microphones and close-talk lapel microphones, worn by each participant. Each meeting had four participants. Two meeting types were recorded and transcribed: actual ad-hoc meetings and scenario-based meetings, where people had been briefed to talk about a particular topic beforehand.

For the standalone system performance analysis, we extracted speech data from the original corpus to evaluate performance for a set of 24 speakers (9 female, 15 male). From each speaker 5 minutes of speech out of two different meetings were used and annotated with a speaker ID. We used audio data recorded from all individual lapel microphones for this purpose[1]. As the audio files were originally recorded with 16 kHz, we resampled it to 8 kHz. An anti-aliasing FIR filtering was performed prior to downsampling. A two-fold cross-validation was applied to partition the speech data into training and evaluation set.

B. LPCC/MFCC vector dimension

We analyze the performance of LPCC and MFCC performance on the AMI dataset. For speaker enrollment a training time of 20 s was applied and the number of centroids per model was set to K=16. In recognition mode an epoch time of 5 s was used.

[1]Described as lapel mix at the AMI website.

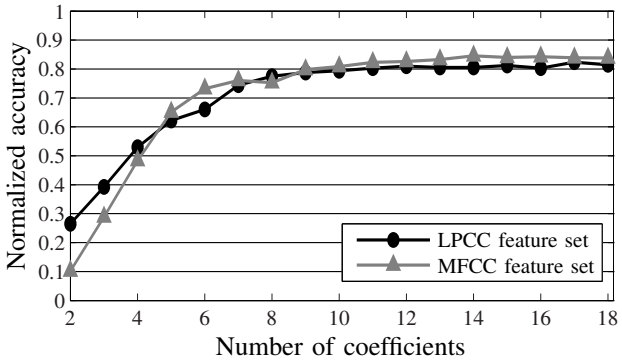

Figure 3. System performance for LPCC and MFCC modeling with different feature vector dimensions (cepstrum coefficients).

Figure 3 shows recognition performance of LPCC and MFCC algorithms for different feature vector sizes N (number of coefficients). We observed that both modeling approaches yield similarly good results. Increasing the number of coefficients, increases recognition accuracy. For more than 12 coefficients, however, performance only marginally increased. Consequently, lower cepstral coefficients carry most of the speaker individuality. These results for the AMI dataset confirm earlier performance reports [31], [32].

As the MFCC algorithm uses FFT, its complexity is larger than that of LPCC [33]. Since the performance of both methods was similar, we used LPCC in further analysis steps and set the number of coefficients to N=12.

C. VQ codebook dimension

Figure 4 shows the performance with regard to the codebook size K per speaker model. We observed that accuracy improves with more centroids per model, however, with more than 16 centroids, performance increases only marginally. Nevertheless, recognition complexity using the VQ method depends linearly on the number of centroids. For further analysis we set $K = 16$.

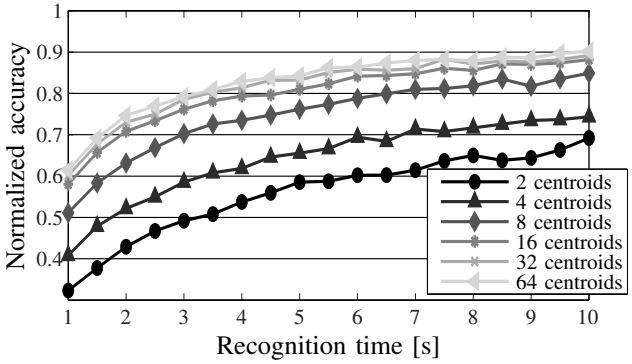

Figure 4. System performance with regard to the codebook size K (numbers of VQ centroids).

D. Training and recognition time

Due to the unsupervised online operation both learning and recognition must be performed with in size-constrained data. We analyzed the number of feature vectors needed to train (parameter L) and recognize a speaker (M). Figure 5 shows the system performance with regard to training and recognition time. The results confirm that below 5 s of recognition time system performance decreases rapidly. In contrast, only marginal improvements are obtained for more than 6 s of recognition time. With 10 s of training data per speaker, recognition accuracy was below 50%, while >70 s did not further improve performance.

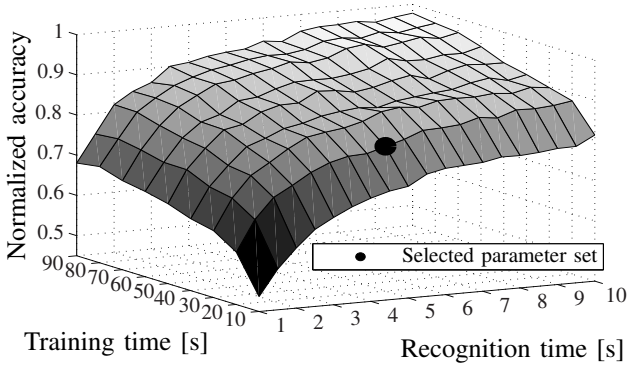

Figure 5. System performance trade-off with regard to training and recognition time. The selected parameter set (marked point) corresponds to an accuracy of 0.81.

As it is desirable to recognize a speaker in short speech segments, recognition time must be short. In addition, there is potentially only little training data available during conversations to learn a new speaker online. We selected 20 s training, and 5 s recognition time (see Figure 5) for system implementation and further evaluations.

E. Score functions for new speaker detection

Initially unknown speakers are detected by the system using a score function as described in Section III-C. Using the system parameters chosen before, we compared the ICN score ($\text{score}_{\text{ICN}}()$) to both alternatives. Figure 6 shows the result for all three score functions using Receiver Operating Characteristic (ROC) analysis and 20 s training time. The area under the curve (AUC) was used to compare the score functions performance. score_D yields a low performance (AUC=0.71) compared to ICN with AUC=0.91. The best result was obtained for $\text{score}_{\text{win}}$, with AUC=0.94.

For our real-time system implementation it is important to avoid recalculating the best matching model that results from the 5 s recognition epochs for the total 20 s time frame. Hence, we applied a majority voting over the four frames obtained from recognition. With this scheme, AUC drops

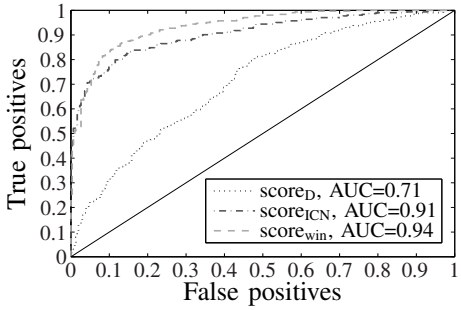

Figure 6. ROC performance for the new speaker detection using three score functions. Results for the classification of known and unknown speakers are shown. The curves were derived by varying a decision threshold on the score functions results. AUC measures the area under the curve.

from 0.94 to 0.93 for score$_{win}$. In return, complexity of the algorithm is greatly reduced.

V. Collaborative annotation of personal systems

We expect that fusing information from two or more speaker identification systems will increase the local system performance. In particular, we focus on a joint new speaker detection in collaborating systems.

A. Information fusion in the collaborative setting

Collaboration between individual systems requires an information exchange and fusion strategy which, in particular, for wearable systems, is constrained by wireless communication bandwidth and power consumption. While collaboration can be implemented by fusing information on several levels, including raw audio data, feature, speaker recognition, new speaker detection, and speaker model, only levels that provide a compressed form of information are viable in a wearable system. In our speaker identification approach, fusion at raw data, feature, and speaker model level would require each participating system to transmit net data rate of 128 kbit/s, 83.4 kbit/s, and 12.29 kbit/model, respectively. In contrast, information fusion at the level of speaker recognition and new speaker detection requires less than 64 bit per epoch, to transmit the decision. As recognition epoch and new speaker detection epoch may be typically larger than one second, this communication will require far lower bandwidth compared to the options summarized before.

An additional challenge in collaborations is the "compatibility" of exchanged information. For speaker identification, the channel properties may differ for individual recording systems, which would render the exchange of model or raw data more complex than using identities only. Moreover, when speaker IDs are exchanged, these would need to be compatible or known between the collaborating systems. Thus, we chose to exchange new speaker detection events.

In our approach, each system can subsequently merge the received detection information, e.g. by using a majority voting, or a more complex fusion scheme. In this work, we evaluated uniform majority voting and a weighted voting, based on the collaborating systems detection scores.

B. Evaluation of collaborative systems during meetings

In order to evaluate the benefit of collaboration among speaker identification systems, the six meetings of the AMI corpus were selected as evaluation dataset. These meetings were conducted with four participants each. Hence, two to four systems could collaborate to decide whether a speaker is known or unknown. Each system uses the lapel microphone of the owner as input for the identification and new speaker detection. Collaboration between the systems is performed by exchanging the individual detection information.

The meetings were annotated for performance analysis similar to the steps described for the standalone system evaluation in Section IV-A. A two-fold cross-validation was used to partition data of each speaker into training and evaluation set. In total 8 min audio data were available from each speaker.

In our evaluations we ensured compatible speaker IDs and new speaker detection information among all collaborating systems. We assumed that all participating systems were started at the same time and a new speaker model was learned only when a collective decision (using the voting schemes) resulted in an unknown speaker. These choices were made to identify benefits that the information fusion scheme could provide. Further synchronization protocols are needed to distribute unique speaker IDs among the collaborators.

Figures 7 and 8 show the collaborative new speaker detection performance for both information fusion strategies and all three score functions in comparison to the standalone system performance. While the analysis was performed for each meeting individually, the results show the average performance for all meetings. However, the standalone system performance shown here, does not correspond to the performance achieved in the standalone system evaluation (Sec. IV). There, 24 speakers were virtually assembled in one meeting.

Both fusion strategies indicate a continuous increase of detection accuracy with the number collaborating systems. Hence, collaboration provides a clear benefit in our unsupervised speaker identification approach. Moreover, the weighted voting can outperform the uniform voting scheme when the number of collaborating systems increases and the ICN score function is used.

The ICN metric (score$_{ICN}$) benefits the most from collaboration. Here, accuracy increased from 74.63% for the standalone system to 82.73% for the majority voting, and 87.49% for the score weighted voting. score$_{win}$, in contrast, did not improve similarly. We attribute this result to the

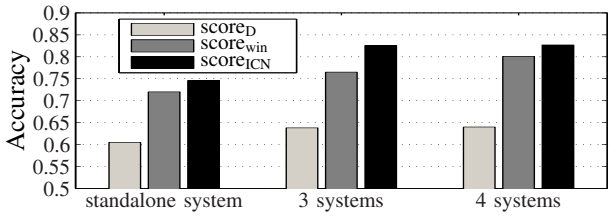

Figure 7. Performance for collaboration of multiple systems in comparison to the standalone system. The bars show results for a fusion scheme using *majority voting* and all three score functions.

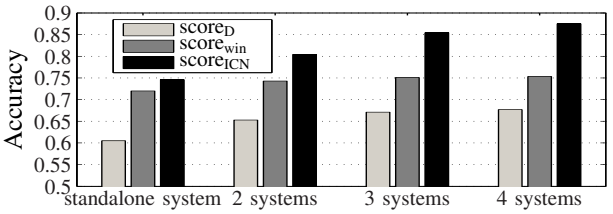

Figure 8. Performance for collaboration of multiple systems in comparison to the standalone system. The bars show results for a fusion scheme using *score weighted voting* and all three score functions.

system-dependent threshold of this function. Consequently, $\text{score}_{\text{win}}$ cannot be compared between systems as $\text{score}_{\text{ICN}}$.

VI. Standalone System Deployment on DSP

We implemented the speaker identification system on a wearable DSP system using Matlab-Simulink with the goal to identify speakers in standalone operation, as described in Section III.

While the performance results derived in Section IV were obtained in simulations using a desktop workstation, the results in this section refer to the actual wearable DSP system implementation. The Matlab algorithms and Simulink models remained the same for both evaluations. Hence, the recognition performance results presented above are valid for the DSP system as well.

The computational performance analysis presented in this section is based on run-time tests executed on our DSP system. Theoretical complexity analysis of the LPCC and VQ algorithms have been detailed in other works [33].

A. Implementation using Matlab-Simulink

The identification system was implemented in Matlab-Simulink. Predefined Simulink blocks from the library "Signal Processing Blockset" were used to facilitate system design. These included operations, such as "Autocorrelation LPC" and "LPC to Cepstral Coefficients". The designed solution was subsequently evaluated on a desktop workstation as presented above.

In a second step, DSP-specific interface blocks were added to the design. We used the library "Target for TI C6000" to generate executable code for the DSP from Simulink. Audio signal inputs, LEDs, Switches, memory operations, and special routines for the DSP board were controlled by blocks included in the library.

Simulink uses "Matlab Real-Time Workshop" to generate C code supported by the DSP platform. This code is then transfered to the development application (Code Composer Studio for the TMS320 DSP processor family from Texas Instruments), to build an executable for the intended DSP processor. The Real-Time Workshop build process loads the specified machine code to the board and runs the executable file on the DSP system. The hardware evaluation was performed using a Texas Instruments TMS320C6713 DSP clocked at $\sim$225 MHz with 16 MB of memory.

Nevertheless, we had to optimize the automatically generated code so that a sufficient processing performance of the system was achieved. The changes involved implementing an additional DSP routine as special block in Simulink.

B. System configuration

We selected a parameter set according to the evaluation results in Section IV), such that a trade-off between processing performance and recognition performance is achieved.

The speaker identification system was modeled with Simulink as a multirate system: every 20 ms a new 12-LPCC feature vector is extracted, every 5 s a speaker recognition is performed. The new speaker detection is done on a 20 s frame every 5 s. The decision is based on score_{win} with threshold $TH_{win} = 0.107$. If a speaker is classified as unknown a new 16-VQ speaker model is created based on the 20 s decision frame. Simulink separated these rates in three synchronous, periodically scheduled tasks with fixed priorities. The task with smallest period has the highest, whereas the task with the longest period has the lowest priority.

C. Optimization of the implementation

The automatically generated code was further optimized manually to achieve optimal system performance. In particular, we used the dedicated DSP function for calculating the squared sum of vector elements, according to $\text{vecsumsquared}(v) = \sum_i v(i)^2$. This function permits an efficient processing of the squared Euclidean distance, while Simulink did not provide a predefined block for this purpose. The optimized code was imported as S-function to the Simulink design to avoid manual changes after Simulink code generation. Performance improvement due to optimization is discussed in the next section and summarized in Table I.

D. Processing performance analysis

We analyzed real-time processing performance of the implementation on the DSP system and compared this result to the host workstation. Using the implementation generated by Simulink without optimization, as detailed above resulted in a online learning time of 25 s and real-time recognition of

up to 4 speakers without concurrent learning. These results are insufficient for the targeted real-time operation. Our analysis revealed that processing of the Euclidean distance was the limiting element. For every feature vector of 12 elements the distance to 16 centroids had to be determined, where 50 feature vectors were derived per second. This results in 9600 squaring operations per second.

Using the code optimization approach, clearly improved processing performance. The DSP system was able to recognize up to 150 speakers in real-time. We derived this result by virtually increasing the number of speaker models in the system's database. Furthermore, deriving a new speaker model for online learning required 5 s. The online learning could be initiated only after the 20 s of data have arrived. Hence in total 25 s were needed to enroll a new speaker until it could be identified from the database.

For comparison we evaluated the performance for a Intel Pentium 4, 3 GHz system. For this system a maximum of 70 speakers could be recognized in real-time. Table I summarizes the results.

System	**DSP** TI TMS320C67 unoptimized	**DSP** TI TMS320C67 optimized	**PC** Intel Pentium 4 (3 GHz)
Speakers in recognition	≤ 4	≤ 150	≤ 70
Learning time	25 s	5 s	16 s

Table I
PROCESSING PERFORMANCE OF THE IMPLEMENTED SYSTEM.

E. Wearable DSP prototype device

We analyzed the integration of the DSP system into a wearable device prototype to confirm the viability of our approach towards a personal speaker identification. For this purpose we designed a custom system, including the TI TMS320C67 DSP, audio interface, USB host connection, and power supply to attach a battery. Moreover, the system included 64 MB SDRAM memory and 16 MB flash. The system was designed to wear it attached to a belt. An external battery could be attached to the device.

The design was split into a main board, containing the DSP and memories, and a interface board, containing power supply, and interfaces. Both board can be stacked to minimize the design size. Figure 9 shows both boards. In stacked format the system had an outline of 55x40x22 mm. In our initial investigation we used an existing battery design that provided 4.1 Ah capacity.

We performed an initial power consumption analysis of this system, where the DSP was clocked at ~197 MHz. When capturing audio at 8 kHz the system consumed

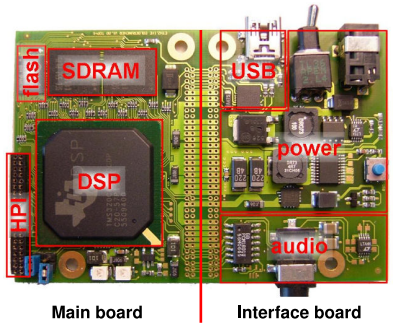

Figure 9. DSP integration for a wearable system. The design consists of two boards that can be stacked to achieve an outline of 55x40x22 mm.

928 mW. When the system executed additional processing algorithms in addition to audio capturing, and stored results to flash memory, consumption raised to 976 mW. However, when capturing audio at 48 kHz, 996 mW were required even without further processing. The latter result indicates that audio capture has an impact on consumption. Hence, processing two audio streams, as it would be required for source separation, or processing higher sampling rates increase the power consumption challenge for a wearable system. At standby the system consumed 308 mW. We expect that this standby consumption can be reduced by optimizing the power supply of the analyzed design. Although the current design did not feature a wireless interface to utilize the collaboration feature, we expect that the low bandwidth required will allow using energy-efficient communication techniques, such as ZigBee.

These consumption results cannot be compared to audio processing systems aiming at ultra low-power operation of 0.1W, such as the SoundButton [34]. In contrast, the design implemented here targets rapid prototyping, e.g. using the Matlab-Simulink toolchain. This concept allows to process complex algorithms such as the unsupervised speaker identification demonstrated in this work. Nevertheless, even at this current power consumption rate, the device had a measured battery operation time of 8.6 hours between recharges. This will be a sufficient runtime to further study the personal speaker annotation in various applications.

VII. CONCLUSIONS AND FUTURE WORK

We presented in this work an unsupervised real-time speaker identification approach intended for a personal wearable annotation system. The system provides recognition and an online learning functions that operate in parallel to identify speakers from a model database, detect unknown speakers, enroll new speakers, and optionally collaborate in new speaker identification.

We evaluated our design decisions regarding the real-time implementation on the freely available AMI dataset that has not been used for speaker recognition before. Our results

indicated an excellent performance of up to 81% recognition rate for 24 speakers and a recognition time of 5 s.

We investigated the collaboration of multiple systems for determining whether a new speaker was observed. The new speaker detection is a particularly critical function for unsupervised systems for robustly enrolling new speakers. The analysis showed that a collaborative operation can increase detection performance by up to 17% with four collaborating systems, demonstrating the clear benefit of collaborations. Moreover, our analysis revealed that not all scoring metrics scale equally well for collaborative operation. We found that the ICN metric was the most robust and provided the largest performance gains with an increasing number of systems.

Finally, we reported implementation results from deploying our speaker identification approach to a wearable DSP system using Matlab-Simulink. With manual optimizations, the implementation was able to process up to 150 speaker models on a DSP in real-time. Learning time for enrolling a new speaker was 5 s. Including the lead-time for new speaker detection, an unknown speaker could be enrolled (from initial voice samples to model in the database) within 25 s. Our subsequent evaluation of a wearable implementation prototype showed that the system could continuously operate for >8 hours using a 4.1 Ah battery. These results combined with the excellent recognition performance confirm the viability of our speaker identification approach on a wearable device.

We initially implemented the DSP system for standalone operation. By extending this platform with a low-power radio (e.g. Zigbee), we expect that the system would be viable for the collaborative mode as well. Information at the level of new speaker detection, speaker recognition, or speaker model could be exchanged without exceeding bandwidth capacities of low-power transceivers. With the increasing performance of novel smart phones, our speaker identification approach, using one microphone only, could be ported to such architectures in the future.

We assumed in this work that the analyzed audio data contains speech information only. We expect that a robust voice activity detection (VAD) can be added to the system to perform an a-priori speech segmentation.

While the system can operate robustly with the selected training time, a faster enrollment may be desirable. For this purpose the GLA algorithm would need to be replaced by another clustering approach that permits an incremental model creation. A weaker model could then serve to recognize the speaker during the first few seconds already. An alternative is to extend the collaboration concept by exchanging information from the participants' model database. If the new speaker had been enrolled by one collaborating system already, all participating systems could benefit from this model immediately. In a practical implementation of such an approach, however, additional considerations must be made, such as how to ensure uniqueness of speaker IDs between systems.

ACKNOWLEDGMENT

This work was supported by the EU project SENSEI, contract number 215923 (www.sensei-project.eu).

REFERENCES

[1] T. K. Choudhury and A. Pentland, "The sociometer: A wearable device for understanding human networks," in *Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW 2002)*, 2002, workhop on Ad Hoc Communications and Collaboration in Ubiquitous Computing Environments.

[2] N. Kern, B. Schiele, H. Junker, P. Lukowicz, and G. Tröster, "Wearable sensing to annotate meeting recordings," *Personal Ubiquitous Computing*, vol. 7, no. 5, pp. 263–274, October 2003.

[3] J. Gemmell, G. Bell, and R. Lueder, "Mylifebits: a personal database for everything," *Communications of the ACM*, vol. 49, no. 1, pp. 88–95, Jan 2006.

[4] M. Blum, A. Pentland, and G. Troster, "Insense: Interest-based life logging," *IEEE Multimedia*, vol. 13, no. 4, pp. 40–48, Oct.-Dec. 2006.

[5] "Smart meeting room, dalle molle institute," http://www.idiap.ch/scientific-research/smart-meeting-room.

[6] "Icsi smart meeting room, berkeley," http://www.icsi.berkeley.edu/Speech/mr/.

[7] Y. Chen and Y. Rui, "Real-time speaker tracking using particle filter sensor fusion," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 485–494, March 2004.

[8] D. Gatica-Perez, G. Lathoud, J.-M. Odobez, and I. McCowan, "Audiovisual probabilistic tracking of multiple speakers in meetings," *IEEE Transactions on Speech and Audio Processing*, vol. 15, no. 2, pp. 601–616, February 2007.

[9] T. K. Choudhury, "Sensing and modeling human networks," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.

[10] D. Olguin, P.A.Goor, and A. Pentland, "Capturing individual and group behavior withwearable sensors," in *Proceedings of AAAI Spring Symposium on Human Behavior Modeling*, 2009.

[11] T. Kim, O. Brdiczka, M. Chu, and J. Begole, "Predicting shoppers' interest from social interactions using sociometric sensors," in *Extended Abstracts of 27th Annual CHI Conference on Human Factors in Computing Systems (CHI 2009)*, 2009.

[12] U. Anliker, "Speaker separation and tracking," Ph.D. dissertation, Swiss Federal Institute OF Technology Zurich, 2005.

[13] D. Charlet, "Speaker indexing for retrieval of voicemail messages," in *Proceesings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, vol. 1, 2002, pp. 121–124.

[14] L. Lu and H.-J. Zhang, "Speaker change detection and tracking in real-time news broadcasting analysis," in *Proceedings of the tenth ACM international conference on Multimedia (MULTIMEDIA '02)*. New York, NY, USA: ACM, 2002, pp. 602–610.

[15] S. Kwon and S. Narayanan, "A method for on-line speaker indexing using generic reference models," in *Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH 2003)*, 2003.

[16] D. Lilt and F. Kubala, "Online speaker clustering," in *Proceedings of theIEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, vol. 1, May 2004, pp. I–333–6.

[17] T. Matsui and S. Furui, "Comparison of text independent speaker recognition methods using vq distortion and discrete/continuous hmms," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 1992)*, 1992.

[18] M. Nishida and T. Kawahara, "Speaker model selection using bayesian information criterion for speaker indexing and speaker adaptation," in *Proceedings of the NIST Rich Transcription Meeting Recognition Evaluation Workshop*, 2003.

[19] S. Nedevschi, R. Patra, and E. Brewer, "Hardware speech recognition for user interfaces in low cost, low power devices," in *Proceedings of Design, Automation Conference*, 2005.

[20] S. Melnikoff, S. Quigley, and M. Russell, "Implementing a simple continuous speech recognition system on an fpga," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM 2002)*, 2002.

[21] T. Liang, T. Zhang, G. J. Zhang, and X. Jun, "Design and implementation of speaker recognition system based on fpga and dsp," *CNKI Application of Electronic Technique*, September 2008.

[22] I. McCowan and D. Moore, "Small microphone array: Algorithms and hardware," IDIAP, Ecole Polytechnique Fdrale de Lausanne, Tech. Rep., 2003.

[23] J. R. Deller, J. G. Proakis, and J. H. Hansen, *Discrete-Time Processing of Speech Signals*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.

[24] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.

[25] M. Do and M. Wagner, "Speaker recognition with small training requirements using a combination of vq and dhmm," in *Proceedings of Speaker Recognition and Its Commercial and Forensic Applications*, 1998, pp. 169–172.

[26] T. Kinnunen, T. Kilpelainen, and P. Franti, "Comparison of clustering algorithms in speaker identification," in *Proceedings of the IASTED Int. Conf. Signal Processing and Communications*, September 2000, pp. 222–227.

[27] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design." *IEEE Transactions. on Communications*, vol. 28, pp. 84–95, 1980.

[28] R. A. Finan, A. T. Sapeluk, and R. I. Damper, "Impostor cohort selection for score normalisation in speaker verification," *Pattern Recognition Letters*, vol. 18, no. 9, pp. 881–888, 1997.

[29] A. M. Ariyaeeinia, J. Fortuna, P. Sivakumaran, and A. Malegaonkar, "Verification effectiveness in open-set speaker identification," in *Proceedings of the IEEE Vision, Image and Signal Processing*, vol. 153, no. 5, October 2006, pp. 618–624.

[30] E. funded AMI project, "The ami meeting corpus," http://corpus.amiproject.org/, 2008.

[31] T. Kinnunen, "Spectral features for automatic text-independent speaker recognition," Licentiate Thesis, University of Joensuu, Finland, 2003.

[32] J. Ajmera and C. Wooters, "A robust speaker clustering algorithm," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2003, pp. 411–416.

[33] E. Karpov, "Real-time speaker identification," Master's thesis, University of Joensuu, 2003.

[34] M. Stäger, P. Lukowicz, N. Perera, T. von Büren, G. Tröster, and T. Starner, "Soundbutton: Design of a low power wearable audio classification system," in *ISWC 2003: Proceedings of the 7th IEEE International Symposium on Wearable Computers*, Oct 2003, pp. 12–17.