Hardware Architectures for Embedded Speaker Recognition Applications: A Survey

HASNA BOURAOUI, University of Tunis El Manar, Tunisia
CHADLIA JERAD, University of la Manouba, University of Carthage, Tunisia
ANUPAM CHATTOPADHYAY, Nanyang Technological University, Singapore
NEJIB BEN HADJ-ALOUANE, University of Tunis El Manar, Tunisia

Authentication technologies based on biometrics, such as speaker recognition, are attracting more and more interest thanks to the elevated level of security offered by these technologies. Despite offering many advantages, such as remote use and low vulnerability, speaker recognition applications are constrained by the heavy computational effort and the hard real-time constraints. When such applications are run on an embedded platform, the problem becomes more challenging, as additional constraints inherent to this specific domain are added. In the literature, different hardware architectures were used/designed for implementing a process with a focus on a given particular metric. In this article, we give a survey of the state-of-the-art works on implementations of embedded speaker recognition applications. Our aim is to provide an overview of the different approaches dealing with acceleration techniques oriented towards speaker and speech recognition applications and attempt to identify the past, current, and future research trends in the area. Indeed, on the one hand, many flexible solutions were implemented, using either General Purpose Processors or Digital Signal Processors. In general, these types of solutions suffer from low area and energy efficiency. On the other hand, high-performance solutions were implemented on Application Specific Integrated Circuits or Field Programmable Gate Arrays but at the expense of flexibility. Based on the available results, we compare the application requirements vis-à-vis the performance achieved by the systems. This leads to the projection of new research trends that can be undertaken in the future.

CCS Concepts: • **Computer systems organization** → **Embedded hardware**; • **General and reference** → *Surveys and overviews*; • **Hardware** → Digital signal processing

Additional Key Words and Phrases: Embedded hardware, speaker recognition, acceleration, classification algorithms and implementations

ACM Reference Format:
Hasna Bouraoui, Chadlia Jerad, Anupam Chattopadhyay, and Nejib Ben Hadj-Alouane. 2017. Hardware architecture for embedded speaker recognition applications: A survey. ACM Trans. Embed. Comput. Syst. 16, 3, Article 78 (April 2017), 28 pages.
DOI: http://dx.doi.org/10.1145/2975161

1. INTRODUCTION

Speaker recognition is generally used in relation to speaker identification or speaker verification techniques and approaches based on information contained in the acoustic signal. The aim is to recognize a person from his/her voice. The field of applications related to speaker recognition is very wide and encompasses domestic applications, military applications, and automotive sector and access control applications [Singh et al.

Authors' addresses: H. Bouraoui, University of Tunis El Manar, Tunisia and TU Dresden, Germany; email: hasna.bouraoui@tu-dresden.de; C. Jerad, University of la Manouba and University of Carthage, Tunisia; email: chadlia.jerad@ensi-uma.tn; A. Chattopadhyay, Nanyang Technological University, Singapore; email: anupam@ntu.edu.sg; N. B. Hadj-Alouane, University of Tunis El Manar; email: nejib_bha@yahoo.com.

DOI: http://dx.doi.org/10.1145/2975161

2012]. These applications could be deployed in an embedded systems. The embedded systems design makes it imperative for hardware implementation platform choice to be appropriate and to help fulfill application constraints.

Speaker recognition techniques generally uses a three-stage process:

(1) It starts with the acoustic analysis of the speech signal for features extraction.
(2) Then a speaker modeling stage is used.
(3) Finally, the decision to accept or reject the speaker is done in the last stage.

Speaker recognition applications are characterized by two main constraints, which are the heavy computational burden and the hard real-time constraints. Additional constraint that is related to memory access demands may depend on the type of the performed recognition. In order to make these applications more popular, the aforementioned challenges should be resolved. There are many techniques for accelerating processes, each one with a focus on one or more particular aspect(s)/metric(s) and thus leading, in general, to a diverse set of points of the design space.

In this article, we give a survey of the existing solutions of embedded speaker recognition applications. Since speech recognition and speaker recognition have a number of steps in common, we included speech recognition acceleration in the scope of the survey. Indeed, certain works can be used for both types of applications. To the best of our knowledge, there is only one work reported in the literature that surveyed speaker recognition applications with a focus on implementation technologies [Alee et al. 2013]. However, the study that we conduct goes much deeper and takes into consideration a greater number of selected articles in the scope of this work. Since it is very hard to benchmark or compare the different existing systems, in several scenarios, we resorted to a qualitative classification instead of a quantitative one. Additionally, several discussions and possible future directions are proposed based on the aforementioned classification. Thus, this work can be seen as a piece contributing to the determination of the algorithm-architecture co-exploration.

The article is organized as follows. In Section 2, speaker recognition is motivated and introduced. The algorithmic process and its analysis are detailed in Section 3. A general overview of hardware implementation technologies for embedded systems is given in Section 4. The scope of the survey and the classification of existing solutions in the literature are given in Section 5. The classification is done based on performance, flexibility, and the tradeoff between them. The following section discusses the overall general trend and gives insights about possible future directions. Finally, Section 7 concludes the article.

2. SPEAKER RECOGNITION AS A PROMISING BIOMETRIC

Since the mid-2000s, we have witnessed the emergence of several applications necessitating individual authentication: financial applications, access control (security), transmission of personal data, teleconferencing applications, and automotive sector. These applications typically use authentication technologies based on biometrics. Biometric authentication [Jain et al. 2004] is the automatic recognition of a person using distinguishing characteristics, that is, physical (biological) or personal behavioral traits, which are automatically quantifiable. These characteristics need to be robust and distinctive and can be used to identify or verify the claimed identity of an individual. Some biometric technologies are relatively simple and low in complexity despite providing highly accurate recognition. They offer many advantages over traditional authentication methods such as the use of passwords or keys and access cards that are highly vulnerable to theft and falsification.

A large amount of biometric information has been used in various applications and areas, including fingerprint, hand, iris, retina, face, voice, veins, and signature

[Yampolskiy and Govindaraju 2008; Delac and Grgic 2004; Bowyer et al. 2013; Bharadwaj et al. 2014]. In our case, we are interested in speaker recognition. Speaker recognition is a technique for automatically recognizing a person communicating from their voice characteristics. The system thus designed is based on certain criteria, taking into account the physical structure of the speech of the individual and the characteristics of its behavior (movement of the mouth, pronunciation, vocal tract, etc.). Thus, speaker recognition refers to the automated method of identifying or confirming the identity of an individual based on his voice. As argued in a recent publication [Fazel and Chakrabartty 2011], speaker recognition represents an interesting biometric for two main reasons:

—First, it can be used remotely.
—Second, it relies on very common acquisition equipment, since microphones are now embedded on most handheld personal devices.

Biometrics are a real alternative to passwords and other credentials for secure access controls. Compared to authentication systems using an object or a password, biometric information is more fluid and provides answers in terms of similarity percentage, while 100% is never achieved. This variation of the identification results of an individual is more related to the quality of the biometric information capture and the modification of the biometric features of individuals, which are generally stable over time.

In our case, we are interested in speaker recognition biometric technology. Each person has her/his own voice that can be captured by a microphone recording. The sounds are characterized by their frequency, intensity, and tone. Speech post-processing takes into account distortions related to the used equipment and can analyze a bad sound such as a telephone or radio transmission, while also fatigue, stress, or cold can cause changes in the voice. Speaker recognition has the advantage of being well accepted by the user, regardless of culture. Furthermore, in the case of a secure telephone transaction, the voice is the only available information. The number of applications continues to grow everyday. This technology is often used in environments where the voice is already captured, such as call centers and telephony, where it is the easiest and convenient biometric to use. Furthermore, the most important advantage of such technology is that it is unable to imitate the voice of a person and it can be used remotely.

2.1. Speaker Recognition Branches

The speaker recognition discipline has many branches, which are either directly or indirectly related. In general, it manifests itself in six different ways, as we can see in Figure 1. Homayoon Beigi [Beigi 2011] categorizes these branches into two different groups, Simple and Compound. The first group, that is, simple speaker recognition, includes branches that are self-contained. The second group (i.e., compound branches) covers those utilizing one or more of the branches of the first group along with eventual additional techniques.

Simple speaker recognition branches include speaker verification, speaker identification, and speaker classification. Speaker verification is based on a system that authenticates if a person is who she or he claims to be. Speaker identification assigns an identity to the voice of an unknown speaker. However, the classification aims to classify similar audio signals into individual bins such as gender, age, and events (music, gunshots, screams). Nevertheless, the compound branches of speaker recognition are segmentation, detection, and tracking. The segmentation goal is to divide the audio utterance into parts covering streams of different speakers, music, and several background conditions. The detection branch uses segmentation as well as identification, because it aims at detecting a specific speaker in a stream of audio. The tracking

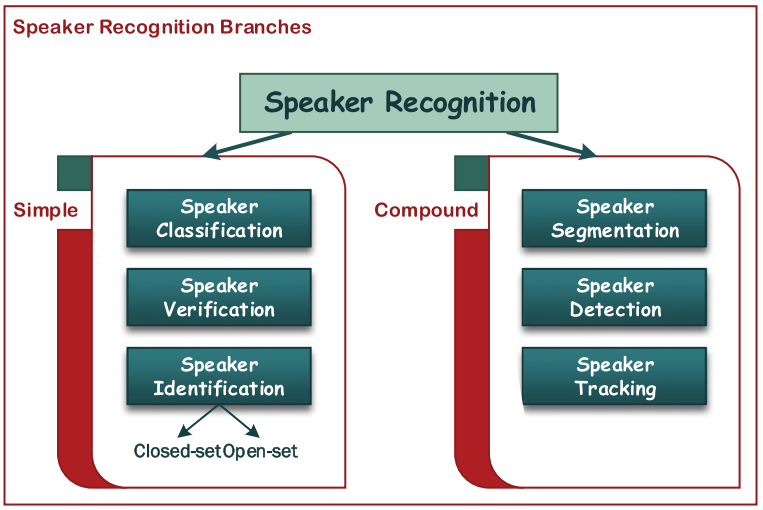


Fig. 1. Speaker recognition branches.

approach consists of continuous detection, as speakers are tracked across the audio stream.

From the previously mentioned branches, the focus is rather made on two main popular branches: speaker verification [Naik 1990] and speaker identification. As mentioned in Beigi [2011], the latter can be divided into two types: a closed-set identification and an open-set identification. This technique makes it possible to use the speaker's voice to verify his/her identity and control access to services such as voice dialing, banking by telephone, telephone shopping, database access services, information services, voice mail, security control for confidential information areas, and remote access to computers.

Furthermore, speaker recognition may be used based on various modalities, which are related to the language, context, and other means. Thus, it can be classified into two categories. The first category includes the systems that are independent of the content of the pronounced sentence, that is, text-independent speaker recognition. In this mode, the speaker can pronounce any phrase to be recognized. There is no constraint on the message that the speaker has to say or the language he/she can use. It requires voice characteristics of the speaker, regardless of the spoken word. The second category includes systems that are executed based on required text, that is, text-dependent speaker recognition. In this case, a list of previously recorded words is needed (vocal signature).

2.2. Commercial Prototypes

Applications related to voice biometrics technology aims to be increasingly deployed in civilian applications. They are growing continually and aim to provide security as well as fraud avoidance. Many commercial applications of biometrics are based on knowledge, such as pin numbers and passwords, or on tokens, such as badges and ID cards [Saquib et al. 2010]. These applications may be divided into three clusters [Jain et al. 2004]: commercial applications dedicated to electronic data security or physical access, government applications related to social security and several kinds of licenses, and forensic applications such as terrorist identification or criminal exploration. Further details are given in Table I.

Table I. Commecial Prototypes

Commercial prototypes	Description	Commercial applications	Government applications	Forensic applications
Agnito (Leader for voice ID products) ▹Spain	-Voice and speech recognition -Aims to prevent crime, identify criminals, and provide evidence in court	-Physical access control, time and attendances, healthcare biometrics -Financial transactions -Mobile authentication [agn 2015a]	-Voice surveillance [agn 2015b]	-Justice and low enforcement -Biometric criminal identification -Automatic Speaker Identification System (ASIS) [voi 2015a]
Speech Technology Center (SCT) ▹Russia	-Technologies for compact and large-scale biometric solutions	-Financial transactions -Solution for real-time speaker identification for phone calls, mobile devices [voi 2015c]	-Systems designed for large city and national system deployments [voi 2015b]	-Justice/law enforcement, logical access control
Nuance ▹Italy	-Biometric sensors and detectors that aims to the use of resources and customer interactions based on better accuracy, reliability, and ease of use.	-Decreasing the number of calls handled by contact center agents based on intelligent call routing technology [aut 2015]	—	—
DAON ▹USA	-Biometric solutions such as fingerprint readers, iris recognition, as well as voice and speech recognition.	-Border control and airports systems and other logical access control systems [log 2015]	-Dedicated to smart cards and signature [ide 2015]	—
Voxomos ▹India [vox 2015]	-A novel voice solution that makes Internet information available as speech in native languages.	-Focused on speech innovations and integration of voice, touch, text, and image to lead the next-generation user-experience of the Web. Mobile, telecom, education, as well as healthcare applications.	—	—

3. OVERVIEW OF SPEAKER RECOGNITION PROCESS

3.1. Generic Process

Generally, the speaker recognition biometric follows a well-defined two-phase process, as illustrated in Figure 2. The first one, which is the enrollment phase (also known as the training phase) aims at determining the parameters of a statistical model of a specific speaker. The obtained model is then stored in a database. During the second phase, which is recognition, the aim is to authenticate an unknown utterance of a speech sample by comparing it to the trained speaker specific model. As already mentioned in Section 2.1, the recognition phase could be either the identification or the verification.

Both processes include two main modules. The first one is a feature extraction module. The last consists of extracting the parameters of the speech signal to obtain a parametric representation of the speaker's vocal tract. The second one is classification [Mller 2007]. This task is performed through the execution of a pattern matching process followed by a decision. In the case of identification, the closest speaker identifier is

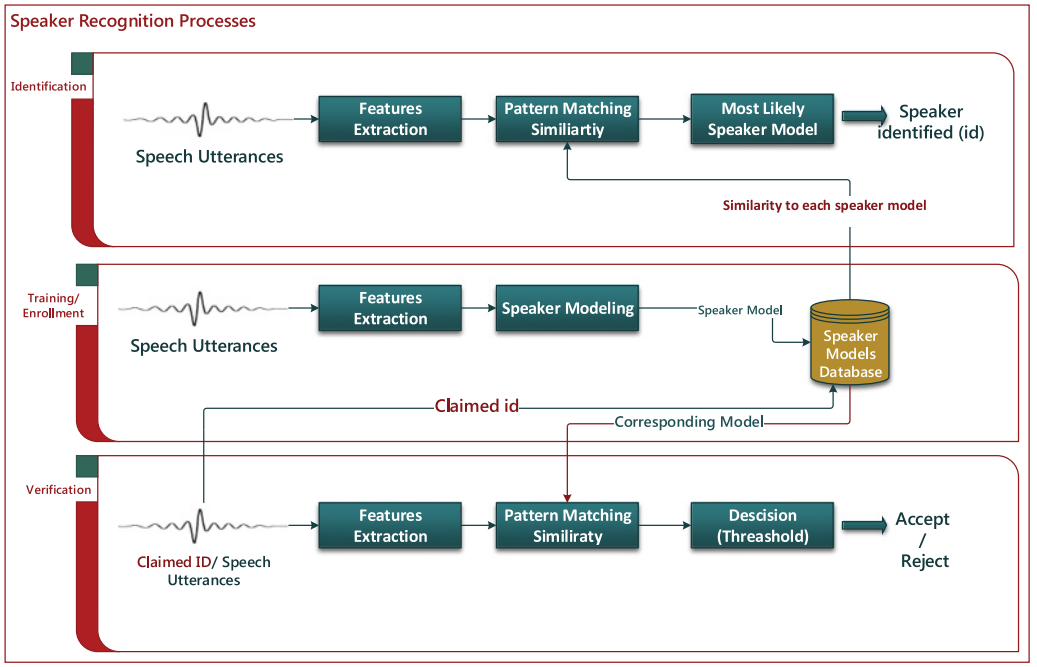


Fig. 2. Speaker recognition process.

returned, while in the case of verification, a threshold is considered to reject or accept the identity of the speaker.

3.1.1. Feature Extraction. The speech signal is complex, as it combines different types of information classified by their level of representation. The information, called low level, is readily available from the digital signal analysis of the word. They include information related mainly to physical traits of the individual (physiological and morphological factors). Information called high level, such as the linguistic or emotional state of the speaker, are much more complex to characterize. This information is related to the sociocultural factors of the individual. Six levels of information hierarchy are identified:

(1) The level of sound: the parameters are linked to the analysis of the spectral envelope of the signal.
(2) Prosodic level: refers to the melody of the speech utterance.
(3) Phonetic level: the distinction between different identifiable sounds of a given language.
(4) The idiolect level: refers to linguistic particularities of an individual.
(5) The dialogic level: defines how an individual communicates, like his or her speaking time in a conversation.
(6) The semantic level: the meaning of speech features.

Speaker characteristics can be categorized based on different features. According to the physical interpretations of these characteristics, it is possible to divide them into (i) spectral features in the short term, (ii) features of the vocal source, (iii) spectro-temporal features, (vi) prosodic features, and (v) high-level features. For further details, readers may refer to Kinnunen and Li [2010].

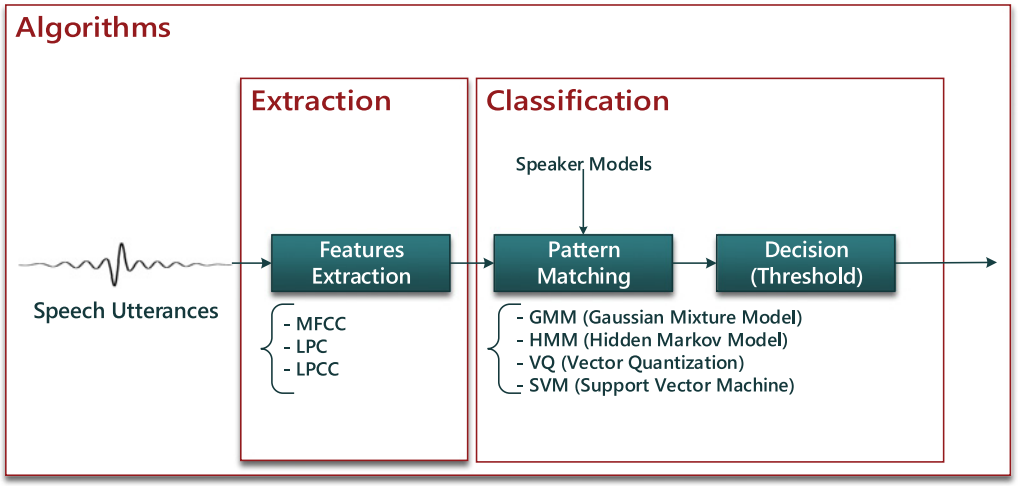


Fig. 3. Algorithms for speaker recognition process.

3.1.2. Pattern Matching. Whether to recognize the message uttered by a speaker or his/her identity, we need to model the entities needed to be automatically recognized later. In text independent speaker recognition, speaker recognition systems try to model the different pronunciations that a speaker may have made for the same patterns. Studying the speakers' speech in several pronunciations of the same pattern, such system can distinguish the characteristics of the speech signal variability allowing it to separate the speakers from each other.

In order to store dependent speaker characteristics, it is important to use algorithms and data types, which are able to capture common points between different representations of spectral patterns forming a given speaker model. It is also important to be able to adapt according to frequency variations and temporal scales of the speech signal. These patterns can be determined by speech segments. In the case of text-independent mode, the phonetic content of the segment is unknown, which is not the case in the text-dependent one. Therefore, these algorithms must be coupled with a measure that would give a value of distortion (or similarity) between the speaker model and a given pattern.

3.2. Algorithmic Background Analysis

Several approaches and algorithms were used to perform a speaker recognition process [Malode and Sahare 2012]. The most popular ones are summarized in Figure 3. In this subsection, we perform algorithmic analysis in terms of computational demands and memory access demands.

3.2.1. Computational Demands Analysis. A feature extraction phase extracts from the waveform the discriminatory information and constructs a representation that can be parametric or not. A wide range of algorithms can be used to perform this step. In the case of parametric representation, we mention Linear Predictive Coding (LPC) and the Linear Prediction Cepstral Coefficient (LPCC) [Tuzun et al. 1994]. However, in the non-parametric case (which is related to the human auditory non-linear frequency characteristics), we cite the Mel-Frequency Cepstrum Coefficient (MFCC) [Togneri and Pullella 2011], which is the most used. Indeed, MFCC performs better than other approaches in the recognition accuracy. For both aforementioned approaches, the Fast Fourier Transform (FFT) is used as one major step. With regards to other performed

Table II. Characteristics of Pattern-Matching Algorithms

Algo.	Pros	Cons	Complexity
DTW	-Easy to implement -Model randomly time wrapping	-Cannot scale well for large vocabulary -Not suitable for changing environments	$O(N^2V)$ N: sequences lenghts V: number of words
HMM	-High recognition rate -Reduced time and complexity -Suitable for large vocabulary	-Huge number of HMM parameters -Large quantity of training data	$O(N^2T)$ N: states number T: observation sequence length
GMM	-Optimal classification performed -Accuracy	Computational complexity	$O(M^2)$ M: Gaussian model dimension
SVM	-Improve the system robustness -Easy Training -Scalability for high dimensional data	Kernel functions are needed	$O(M^2)$ M: training set size
VQ	Reduced storage and computation costs	Potential loss of information due to quantization	$O(KN)$ K: input vector dimension N: code book size

steps, they have lower complexity than the computation of the FFT. Then feature extraction algorithms have complexity of $O(N.log(N))$, where N is the data size [Kumar 2015].

For the classification phase, there are also different well-known approaches for matching speaker recognition algorithms. Some of them are based on statistical pattern matching. This set includes Hidden Markov Model (HMM) [Rabiner 1989] and Gaussian Mixture Model (GMM) [Reynolds 1995]. The complexity of the HMM algorithm is $O(N^2T)$, where N is the number of states and T is the observation sequence length, while the complexity of the GMM classifier is quadratically dependent on the dimension of the Gaussian models. Within the template matching techniques, we cite Dynamic Time Wrapping (DTW), which is used in cases of text-dependent recognition, as it uses a limited dictionary [Maruti et al. 2012]. The complexity of DWT is $O(N^2V)$, where N is the length of the sequence and V is the number of words in the dictionary (or equivalent). Another well-known algorithm for pattern matching is Vector Quantization (VQ) [Makhoul et al. 1985]. This approach is commonly used in the compression of speech signals. It has a time complexity $O(K.N)$, where K is the input vector dimension and N is the codebook size. We find also the Support Vector Machine (SVM) algorithm [Campbell et al. 2006], which constructs a set of hyper-planes in a high-dimensional space for classification task. The complexity is of $O(M^3)$ with M being the training set size.

All these algorithms have advantages as well as disadvantages depending on the type of recognition. Table II summarizes some of their characteristics (advantages and disadvantages) and provides the corresponding complexity. However, and even though GMM is computationally heavy, the use of GMM is most common [Alee et al. 2013] due to the fact that it can be utilized for text-independent speaker recognition. In addition, GMM is based on a probabilistic framework and provides high-accuracy recognition.

3.2.2. Memory Access Demands Analysis. Depending on the type of recognition, whether it is identification or verification, the memory access demands vary widely. Indeed, in case of verification, the size of the stored models in the database has no impact. This is due to the fact that the new model is compared to only the one it is supposed to be. However, in the case of recognition, the given model should be compared to all the stored models in the database. Consequently, the larger it is, the higher the memory access demands and also the computational demands.

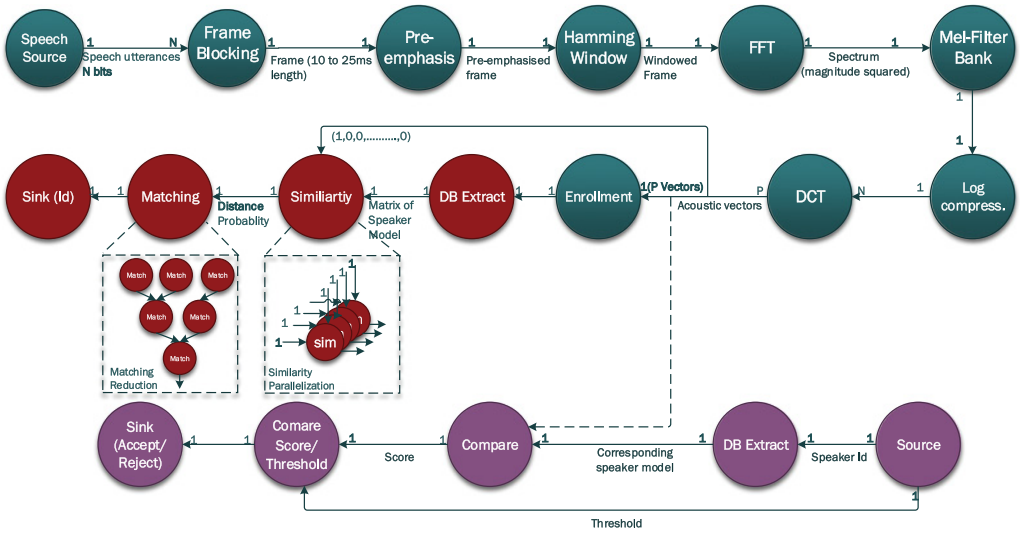


Fig. 4. Data flow graph of the speaker recognition process.

To study memory access demands, it is important to use the appropriate formalism. Hence, we modeled the speaker recognition process by means of a dataflow graph to put the focus on the exchanged data between computing entities. Figure 4 illustrates this graph. The green part represents the front-end processing phase (feature extraction) of the speaker recognition application, which is also applicable for the speech recognition front end. Since MFCC is the most commonly used algorithm (as will be shown later in Section 5), it is used as a basis for detailing this phase and consists of describing different kernels of the algorithm. The rest of the graph presents the second part of the process, which concerns speaker identification (red part) and speaker verification (purple part).

In the part of the graph modeling the extraction phase, nodes present the kernels of the front end where edges model data transfer between these kernels. First, digitized speech utterances are divided into overlapping frames as it is illustrated on the arrow, from one speech input to M frames. Every frame is started every 10ms, and each frame lasts for 25ms. After that, each frame is passed through a filter that increases the energy of signal at higher frequency, thus emphasizing high frequencies. The output is a pre-emphasised frame that passes through a hamming window to smooth the signal in order to reduce spectral effects. This latter frame is then transformed from time into frequency domain by means of an FFT function. The mel-filter bank kernel is responsible for applying a triangular filter bank to approximate the frequency of the human ear. Usually, for a 16KHz sampling rate, a set of 40 mel-filters are used. Mel-weighted spectrum values are replaced by their natural logarithm through the log compression kernel. Finally, a DCT (Discrete Cosine Transform) function is applied in order to reduce spectral information that is acoustic vectors. All N frames are provided as input to the DCT kernel and transformed to P acoustic vectors. We can notice that all the previously mentioned steps could be seen as a Synchronous Dataflow [Lee and Messerschmitt 1987]. Generally, 40 element vectors are reduced to 13 element cepstral vectors. In other terms, 1s of utterances is transformed to approximately 100 acoustic vectors. Thus, and because the data size is small, the front-end part of the process uses less than 1% of the overall computation of the system. It is worth mentioning that floating point computations are used.

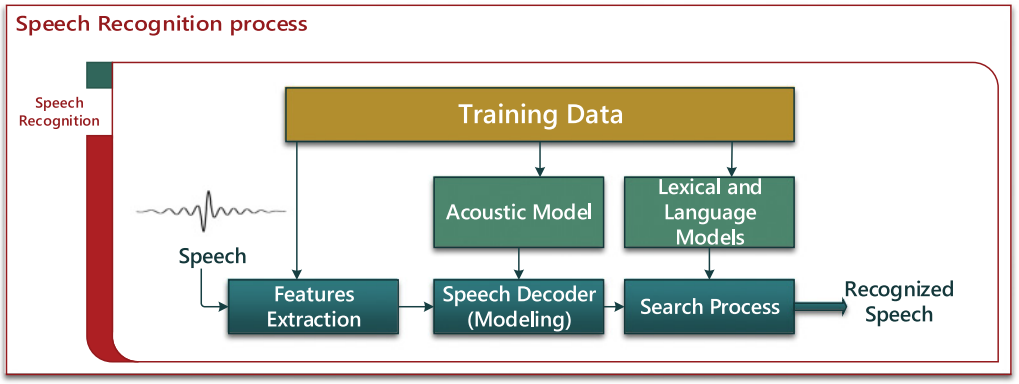


Fig. 5. Speech recognition process.

Regarding the pattern recognition phase within the identification process, we can notice that the P acoustic vectors could be considered as one big token passing through other steps. This token is in the form of a vector containing P elements. After the enrollment phase, the speaker models obtained from feature vectors are stored in the node of the database extract. After that, these feature vectors are passed once through a similarity computation node, which has also as input all the matrix in the database. As we can notice from the figure in the dotted square, similarity can be modeled as a parallel process, where the feature vector are compared to all the speakers in the database. A similarity is calculated for all the input models, in terms of distance or probability, depending on the algorithm used. Then distances or probabilities are passed through a matching node to pick up the closest speaker. In addition, the dotted square matching node could be modeled as a reduction dataflow, where we have to choose the minimum distance or the maximum probability likelihood every time.

For the verification branch, P features vector are passed to the comparison node. Thus with the user id as input from the source node, we fetch from the database node the speaker model corresponding to the declared id, and then the similarity score is computed. This score is compared to the threshold to make a decision to accept or reject the claimed id.

The study of the dataflow leads to the following conclusions. The first part that concerns the feature extraction could be appropriate for pipelining given its structure, whereas data-level parallelism can be considered in the back-end process. These remarks should be used for architecture improvements realization regarding memory as well as computational requirements.

3.3. Speaker Recognition vs. Speech Recognition

Speech recognition, also known as Automatic Speech Recognition, is a technique designed to recognize, in a series of sound signals, phonemes (minimum sound units) and sentences spoken by a speaker. The process is based on matching techniques to compare a sound wave to a set of samples made of words and phonemes. For this purpose, several approaches were used, such as acoustic-phonetic, pattern recognition, as well as artificial intelligence approaches (for further details, readers may refer to Rabiner and Juang [1993]).

As illustrated in Figure 5, speech recognition processes involve three steps. The first one is pre-processing, which consists of feature extraction. During this step, the speech signal (that is a set of acoustic waves) is transformed into a sequence of pre-phonetic symbols with no linguistic meaning but containing features values. The second step

is acoustic modeling, which compares the symbols with specific phonetic waveforms. Several well-known algorithms such as DTW, HMM [Fukunaga 1990], Neural Network, as well as VQ [Bourlard and Morgan 1994] were used for this purpose [Lee et al. 1990]. Finally, recognized speech is obtained after performing a search based on the lexical and language models. The final step consists of searching the corresponding speech, using the lexical and language models. Further details can be found in Rabiner and Schafer [2010].

Speech and speaker recognition processes have a lot of things in common. On the one hand, both use recordings of the human voice and thus use the anatomical features (i.e., age, gender, etc.). On the other hand, both have a phase related to feature extraction and pattern matching. Consequently, the approaches used are the same (MFCC, HMM, DWT, VQ, etc.). However, there is a big difference between both processes [Furui 2005]. For instance, speaker recognition could be independent of the language while speech recognition totally depends on it. Besides, the range of applications in speech recognition differ from those of speaker recognition [Peacocke and Graf 1990]. Further readings are provided in Lee et al. [2012].

Several works have been done in joint speech and speaker recognition approaches. For instance, BenZeghiba [2005] tried to combine an approach where both speaker and speech recognition are performed, leading then to diarization. In Reynolds and Heck [1991], the authors improved speech recognition by adding a specific speaker-dependent recognizer.

4. OVERVIEW OF HARDWARE ARCHITECTURE DESIGN SPACE OF EMBEDDED SYSTEMS

Generally, and based on the requirements of the speaker recognition systems, embedded systems solutions try to achieve a specific list of metrics and constraints in order to meet the system requirements. It is possible to divide the implementation technologies by their minimization of these following aspects:

—Number of clock cycles: This number affects the performance by influencing the system response time, in addition to energy consumption.
—Number of transistors (integration): This metric affects the area/size of the designed system and the energy consumption.
—Energy consumption.
—Production costs: Theese costs include time to market as well as non-recurring engineering costs (cost of the first product).

The problem of improving a metric can have a negative effect on another one. Simultaneous expertise in software and hardware is often required to optimize design metrics. In Platzner and Wehn [2010], the authors scaled the implementation platforms in terms of energy per Million of Operations per Second (MOPS) and MOPS per area, as illustrated in Figure 6. We can conclude that the performance decreases when the flexibility increases. In other words, the magnitude in performance and efficiency varies from one solution to another, either hardware or software, depending on the application in use.

To achieve the needed acceleration and the required metrics, there are many possible hardware platforms:

—General Purpose Processors (GPP): Although the most flexible, these processors exhibit very low performance (i.e., in terms of execution time and cost).
—Digital Signal Processors (DSP): This processor is optimized for signal processing operations. However, a programmable DSP has the following problems: high cost, large size, and high power consumption.

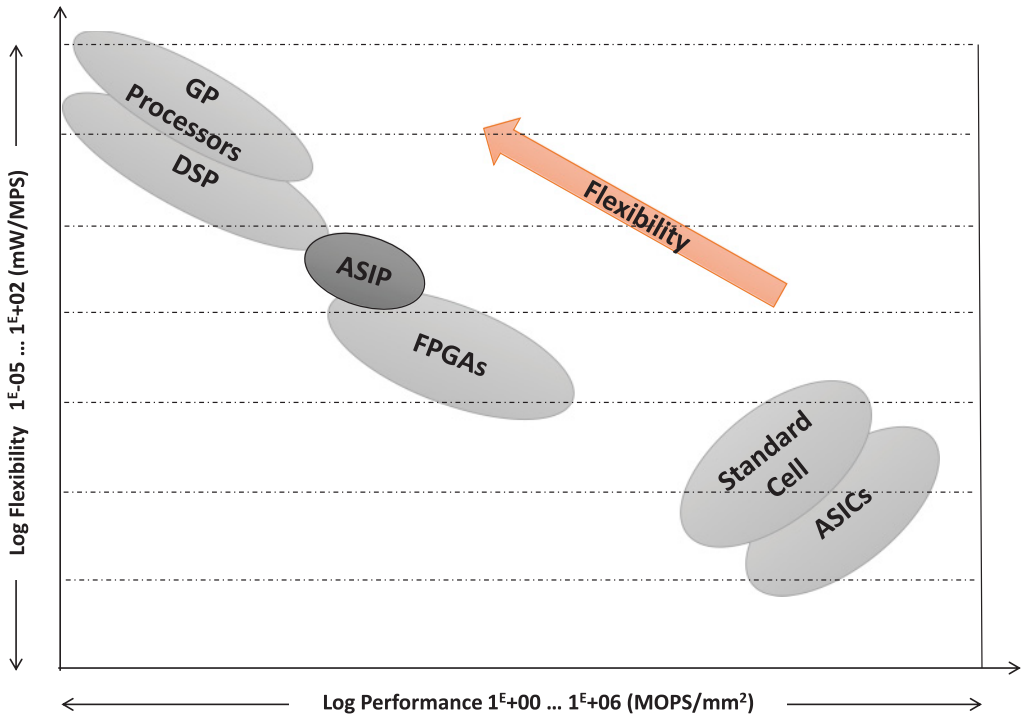


Fig. 6. Hardware architecture design space [Platzner and Wehn 2010].

—Application Specific Integrated Circuits (ASICs): These are customized in one way or another for a specific application. Despite offering the best performance, the high costs and lack of flexibility reduce considerably the use of ASIC-based solutions.
—Field Programmable Gate Arrays (FPGAs): The main advantages of FPGA technology are performance, time to market, cost, and reliability. In fact, the NRE (Non-Recurring Engineering) costs in custom ASIC are much higher than those of FPGA-based hardware solutions. Benefiting from the hardware parallelism, FPGAs offer superior processing power to that of DSP. In the same time, FPGA's flexibility in due to its reconfigurability.
—Application Specific Instruction set Processors (ASIPs): ASIP is a dedicated processor to a field of well-determined application or set of applications. It is intended to accelerate the most commonly used functions.

Figure 6 contains many intersections between these well-known hardware platforms. The one between the ASIP and FPGA design spaces is particularly interesting, as it highlights hybrid computing platforms, such as reconfigurable processors [Chattopadhyay 2013]. The strength of these platforms is their adaptability to reflects the changing technology landscape.

5. CLASSIFICATION

5.1. Scope of the Survey

As the embedded domain is characterized by real-time, cost, and energy constraints, efficient design of hardware solutions is required. If we check the various existing hardware solutions and see their relevance to speaker recognition applications, then we can identify three main clusters:

Table III. Hardware Accelerations Based on ASIC

Ref.	Speech/ Speaker Recognition	Algo. F. E.	Algo. P. M.	Tech. node	Performance			Architecture realization improvement
					Freq.	Power	Area	
[Pihl et al. 1996]	Speech	—	HMM	0.8μm	154MHz	835mW	16.5mm^2 Core: 8.4mm^2	Reduce Memory bandwith by half
[Han et al. 2003]	Speech	MFCC	HMM	0.35μm	20MHz	13.4mW	N/A	Speech IC to reduce the circuit complexity
[Nedevschi et al. 2005]	Speech	MFCC	HMM	0.18μm	N/A	N/A	2.56mm^2	-12× faster/SW solution -Reduce complexity, cost and energy consumption (UI specific recognition)
[Jia-ChingWang et al. 2014]	Speaker/ Verif.	LPC	SVM	90nm	up to 100MHz	8.12mW	7.9×7.9 mm^2 Core: 4.42×4.42 mm^2	-Whole process implemented -Large number of support vectors

—Efficient solutions but with no/low flexibility represent the first cluster. It includes ASICs as well as FPGA-based solutions.
—The second cluster includes flexible solutions with low performance such as DSP and GPP-based solutions.
—The third cluster considers the performance-flexibility tradeoff and brings forth specially tailored solutions.

We conducted studies on the different literature databases, such as Springer, googlescholar, IEEE (Institute of Electrical and Electronics Engineers), ACM (Association for Computing Machinery), and ScienceDirect, in order to select the most recent work on accelerating speech/speaker recognition applications. The early works related to speech recognition date to 1996, while the first works related to speaker recognition date to 2009.

In the following, the main features extracted from state-of-the-art works will be presented in tabular form. In all the upcoming tables, we denote "not supported" by "–" and "not available" by "N/A."

5.2. Performance Based Solutions

5.2.1. State of the Art. In the first cluster, we can pinpoint ASICs, an efficient customized hardware solution, as it has a small size, and small energy consumption with high speed. The ASIC has a major drawback in particular, unlike the printed circuit, which can be controlled at all points of the circuit using conventional apparatus; it cannot be tested in detail after its realization. In the case of non-functioning, diagnosis is difficult to do, especially if testability was not thought out at design time. Diagnosis can be made only through often-complex test programs, which are difficult to develop. Before starting the manufacture of an ASIC, it will be necessary to perform a detailed physical simulation. The ASIC has disadvantages as well as advantages, such as greater resistance to radiation, reduced costs, and a high integration rate (higher density).

Various previous research on hardware designs has described the implementation of the speech as well as the speaker recognition process based on ASICs. These works are listed in Table III, where the used algorithm feature extraction (Algo. F. E.), algorithm for pattern matching (Algo. P. M.), the technology node (Tech. node), and the

Table IV. Speaker Recognition Acceleration Based on FPGA

Ref.	Speaker Identif./ Verif.	Algo. F. E.	Algo. P. M.	Tech. node	Performance				Architecture realization improvement
					Exec. time	Freq.	#LUT	#Models	
[Ramos-Lara et al. 2009]	Verif.	MFCC	SVM	90nm	4647.44μs	50MHz	6138(F.E.) 647(P.M.)	52	-Affordable prices -Reduced area in the FPGA (fixed-point format)
[Sarkar and Saha 2010]	Identif.	LFCC	VQ	130nm	N/A	100MHz	1496(F.E.)	131	-Reduce redundant frames at pre-processing -Low identification time
[Wang et al. 2011]	Identif.	LPCC	SVM	90nm	10s	50Mhz	N/A	NIST 2010/ 9	-Focus on training -Reduce com. time cost and bandwidth using data-packed mechanism (1.05 speedup w.r.t. data-unpacked mechanism) -HW/SW co-designed solution -89.9% identification rate
[EhKan et al. 2011]	Identif. text ind.	MFCC	GMM	150nm	90× faster/ SW impl.	48MHz	16317(F.E.) 34193(P.M.)	400	Large number of voice streams simultaneously in real time
[Li et al. 2012]	Identif. text dep.	MFCC	VQ	90nm	14.976ms	50MHz	N/A	301	–Speedup: 17.6 w.r.t. Matlab PC execution -Identification rate is 93.3%
[Ramos-Lara et al. 2013]	Verif.	MFCC	SVM	90nm	4647.44μs	50MHz	4218(F.E.) 1296(P.M.)	52	-Low cost -fast identity verification
[Ehkan et al. 2015]	—	MFCC	—	0.15μm	N/A	—	39452(F.E.)	—	-Parallelism and pipelining during the signal processing -Improvement of the memory requirements and computational usage

performance (in terms of frequency, power, and area), and the architecture realization impact on the algorithmic demands are given. In Jia-Ching Wang et al. [2014], the authors opted for a high-speed and low-power solution on an ASIC (recall here that ASICs are certainly efficient but not flexible), while Han et al. [2003] focused on increasing the speed and the area efficiency of the system, which was implemented on an ASIC in order to accelerate the pattern comparison and the decision parts of the speech recognition process. In addition, in Nedevschi et al. [2005], the authors concentrated on power consumption, high volume, and the memory bandwidth by parallelizing computation bottlenecks. Thus, they opted for a solution that is at the same time scalable, retainable, and flexible. This improvement enabled the possibility of changing the language as well as the algorithm used. However, in Pihl et al. [1996], authors tried to reduce the memory bandwidth needed for the speech recognition process without affecting the recognition performance.

FPGAs offer another solution with high performance due to the fine-grained parallelism available in the configurable logic blocks. The FPGA configuration (or programming) is done on the field, thus allowing design modifications post-manufacturing, unlike for ASICs. Furthermore, an FPGA solution has advantages over ASICs such as lower development costs, which leads to a smaller NRE. It also offers simplicity of modifications, as well as shorter time to market than dedicated circuits. Although both of these fields, that is, speech recognition and speaker recognition, are close, most of the research works in the literature are concerned with the first area. Many dedicated hardware optimizations based on FPGA for these types of applications were presented through several research works, as shown in the Table IV (for speaker recognition–related works) and Table V (for speech rcognition–related works). In these tables, we give for each cited work the algorithm used for features extraction (Algo. F. E.), the algorithm for pattern matching (Algo. P. M.), the technology node (Tech. node), and

Table V. Speech Recognition Acceleration based on FPGA

Ref.	Algo. F. E.	Algo. P. M.	Tech. node	Performance			Architecture realization improvement
				Speedup	Freq.	Power	
[Vargas et al. 2001]	LPC	HMM	0.25μm	500× faster/ classic Viterbi	N/A	N/A	-Parallel Viterbi implementation -2.64ms/4 words -99.7% of exactitude rate
[Wang et al. 2002]	MFCC	HMM	0.35μm	—	—	—	-Decrease the required computational power and the memory size -4284 4 input LUT -1408 3 input LUT -Improves accuracy
[Yoshizawa et al. 2006]	MFCC	HMM	0.18μm	Speed: 45.5s	80MHz	421.5mW	-Noise robustness -56.9μs/word
[Bourke and Rutenbar 2008]	—	HMM	N/A	1.3	100MHz	196mW	Low-power
[Cheng et al. 2009]	MFCC	GMM	90nm	2.677 (6.64s⇒2.48s)	100MHz	N/A	Word recognition accuracy: 93.42%
[Lin and Rutenbar 2009]	N/A	HMM	N/A	10× faster	N/A	N/A	Large-vocabulary recognizer
[Vu et al. 2010]	MFCC	—	90nm	N/A	as low as 4.1 MHz	N/A	-Low-cost speech recognition systems -10% resource utilization
[Chen et al. 2011]	N/A	GMM	65nm	2.18 faster/ SW impl.	N/A	N/A	-Offloading computation-intensive parts -416 LUT
[Lakshmi and Rao 2013]	MFCC	HMM	90nm	4.7s ⇒1.49s	120MHz	N/A	93.33% of word accuracy
[Bapat et al. 2013]	N/A	HMM	180nm and 65nm	2	62.5MHz	210mW	-50% reduction in decoding latency -442,000 NAND2 gate
[He et al. 2013]	MFCC	GMM	40nm	3.02 and 2.25	62.5 MHz	54.8mW	-Parallel and pipelined architecture -Frequency and power reductions -3.02 speedup/bigram language model -3.25 speedup/trigram language model -Area: 3.86mm^2
[Buitrago et al. 2013]	N/A	HMM	90nm	23.08 faster/ SW impl (9714s⇒414s)	N/A	53.82mW	-Implementation of certain software routines as equivalent hardware models -8% increase in logic blocks

the performance and the architecture realization impact on the algorithmic demands. We note that the performance is given in terms of execution time, frequency, number of Look Up Tables (LUT), and the number of stored models for speaker recognition works, while it is given in terms of speedup, frequency, and power for speech recognition works.

Several research works treated the speaker recognition implementation on FPGAs as shown in Table IV. It is worth mentioning that the computation complexity of the speaker recognition process depends on several parameters influencing memory access demands. These parameters are the number of speakers in the database, the number of frame vectors, as well as their dimensionality. Besides, and depending on the type of recognition, whether it is identification or verification, the memory access demands vary widely.

For example, J. F. Wang et al. [2011] presented a hardware/software co-design for fast trainable recognition systems and optimized the training phase of the speaker recognition process. They opted for a hybrid solution while the training part was implemented on the hardware. SVM training parameters and speech features should be transferred from the software to the hardware. A HW/SW (Hardware/Software) communication block is used. In order to minimize the communication time cost and the bandwidth requirements between the software and the hardware, they used a packed data technology that minimizes transmission number. Furthermore a 50 RAMB16 is used.

In Ramos-Lara et al. [2013] and Ramos-Lara et al. [2009], the authors considered the speaker verification issue. They proposed a system implemented on an FPGA that carries out feature vector extraction. Contrary to the previous work, these authors implemented the whole process on the FPGA. Thus the feature vector extraction node is carried out in 285μs, while matching between this feature vector and the model stored in external SRAM (Synchronous Random Access Memory) is executed in 4.362μs (15 among 40 of the total used resources for 18-Kbit RAM (Random Access Memory)). Furthermore, EhKan et al. [2011] proposed a hardware system that is capable of processing 90 times more audio streams in real time than could be done in a standard computer. The feature vector as well as the model trained from the speech utterances are stored in an external memory (RAM) connected directly to the FPGA.

Last but not least, Li et al. [2012] as well as Sarkar and Saha [2010] used an FPGA for performing acceleration. Li et al. [2012] focused on accelerating the whole process by reducing the time consumption, while Sarkar and Saha [2010] focused on reducing the redundant frames in the feature extraction part and thus gaining a better performance of the system. Sarkar et al. implemented the whole process on dedicated hardware. The models of each speaker are stored in an external database (131 speakers).

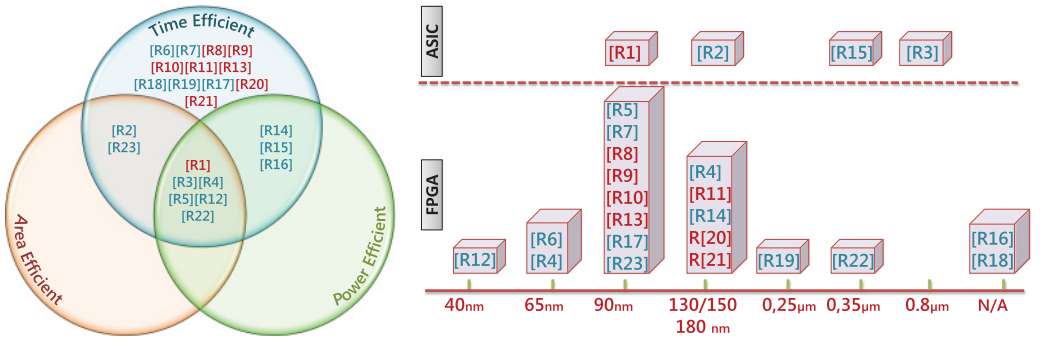
In Bapat et al. [2013], Bourke and Rutenbar [2008], Buitrago et al. [2013], Chen et al. [2011], Cheng et al. [2009], and Lakshmi and Rao [2013], the authors worked on the field of speech recognition, as shown in Table V. In Buitrago et al. [2013], Chen et al. [2011], and Lakshmi and Rao [2013], they tried to solve the issue of computation bottleneck and the energy problem by opting for a hardware/software co-design approach as a solution to accelerate the pattern classification phase or a speech recognizer engine such as Sphinx. For Bapat et al. [2013], they used a pure hardware solution and designed a co-processor in order to accelerate the speech recognition process.

Previous research on custom hardware also described also the speech recognition process using field programmable gate arrays such as in Lin and Rutenbar [2009], and in Vargas et al. [2001] they tried to achieve a high speed by reducing the researching time in the database and the processing time, respectively. He et al. [2013] were interested in reducing the frequency and thus the power consumption for the speech recognition. The architecture of the hardware solution for this article used parallelization as well as pipelining by implementing an ASIC.

Furthermore, Wang et al. [2002] aimed at decreasing the memory requirements of the system while reducing the computational complexity of the MFCC algorithm. In fact, after rescheduling the original MFCC algorithm, the number of needed computation operations in terms of addition/substraction and multiplication were reduced. Conversely, Yoshizawa et al. [2006] proved that the node with the higher priority for acceleration was the output probability calculation. In fact, the number of operations needed for such a kernel is about 335 million operations compared to 4 million for the Viterbi algorithm. Thus, Yoshizawa et al. [2006] achieved processing time, including data transfer, of 45.5ms, which is approximately 56.9μs/word in a 800-word vocabulary recognizer. The HMM training and the data transfer from external to internal memory took approximately 12.8s.

Cheng et al. [2009] tried to achieve real-time requirements of such systems by reducing the memory bandwidth while maintaining the accuracy. In this work, the HMM parameters for the acoustic modeling phase are stored in two memory modules, SRAM and SDRAM (Synchronous Dynamic Random Access Memory), in the dedicated hardware. Thus, memory access demands are reduced thanks to the possibility of retrieving data quickly from the internal memory of the accelerator. However, the internal memory size should be 10 to 100s of KB, which may induce drawbacks. For that system, a transfer of 160 bytes/clock cycle of data to the hardware accelerator is needed.

Even though Graphics Processing Units (GPUs) do not appear in Figure 6, they were widely used as an acceleration platform. This is due to their high arithmetic power and



(a) (b)

Fig. 7. (a) Time efficient, power efficient, and area efficient classification of performance-based solutions. (b) Technology node scaled classification of performance-based solutions. Legend: R1: Jia-Ching Wang et al. [2014], R2: Nedevschi et al. [2005], R3: Pihl et al. [1996], R4: Bapat et al. [2013], R5: Buitrago et al. [2013], R6: Chen et al. [2011], R7: Cheng et al. [2009], R8: Wang et al. [2011], R9: Ramos-Lara et al. [2013], R10: Ramos-Lara et al. [2009], R11: EhKan et al. [2011], R12: He et al. [2013], R13: Li et al. [2012], R14: Yoshizawa et al. [2006], R15: Han et al. [2003], R16: Bourke and Rutenbar [2008], R17: Lakshmi and Rao [2013], R18: Lin and Rutenbar [2009], R19: Vargas et al. [2001], R20: Sarkar and Saha [2010], R21: Ehkan et al. [2015], R22: Wang et al. [2002], R23: Vu et al. [2010]. Red references refer to speaker recognition applications. Green references refer to speech recognition applications.

great memory bandwidth. Indeed, several research works treated GPU-based speaker, as well as speech, recognition acceleration. For instance, Liu [2009] focused on parallelizing algorithms for HMM training and classification, where they reached an 800× and 300× speedup for the forward algorithm and the Baum-Welch algorithm, respectively. Vaněk et al. [2011] focused on the use of very large acoustic models with real-time speech recognition. Other GPU accelerations of the speaker recognition process were presented in Azhari [2011] and Machlica et al. [2011]. Furthermore, in Fuqiu et al. [2012], a hybrid CPU (Central Processing Unit)-GPU platform is used to reach 63.8× speedup, whereas parallel computation techniques are used by Gaafar et al. [2014] to improve computational speed and recognition rate. Both works target speech as well as speaker recognition applications.

Due to their highly parallel structure, GPUs are effective for a wide range of image processing tasks and scientific computing applications. However, they are considered a multicore solution, whereas others (FPGA, ASIC) are still considered unicore. As we focus in this article on speaker recognition embedded applications, GPU-based implementations are discarded and will not be included in trend or future insights discussions.

5.2.2. General Trend Analysis. For all the previously mentioned works, different systems used hardware optimization for speech and speaker recognition applications. We notice that application-specific solutions for speaker recognition started appearing only in 2009. In addition, there was only one work among four dealing with ASIC. The key thrust in performance improvement was to minimize the execution time. Consequently, these works showed that these platforms are well suitable for real-time recognition. They even implemented the highest computationally demanding algorithms, such as GMM.

The aforementioned works are scaled in Figure 7 according to time, energy, area, and technology node. In both charts, red citations refer to speaker recognition applications, while green ones refer to speech recognition. Figure 7(a) extracts information to mention the intersection between the performance related to area, energy, and time efficiency. For instance, fast and low-power solutions are captured in the intersection

between the “Time” and “Power” circles, whereas low-power and low-area solutions are captured in the intersection between “Power” and “Area” circles. Any solution that optimizes all three metrics is in the intersection of the three circles. We note that only 5 works among 23 targeted optimizing simultaneously the the metrics of power, area, and time response. No much attention was paid to power and area, as the main focus was on the real-time aspect. Along with the area minimization, the implementation technology node differs from one work to another. Indeed, Figure 7(b) extracts information to mention the technology node used in each reference work.

5.2.3. Possible Future Directions. Despite the fact that ASIC solutions can offer better performance, the number of FPGA based ones are gaining more and more prominence. Indeed, the need for reconfigurability as an efficient solution for minimizing production and evolution costs is obvious.

Even though FPGA implementations are suitable platforms for reducing power and area, the focus was rather on real-time constraints. Now that embedded systems requirements are increasing in terms of power demands and miniaturization, more work should seriously target these metrics along with execution time demands. Indeed, the gap between power storage evolution and applications requirements can be narrowed with more focus on power consumption while using reconfigurable platforms.

5.3. Flexibility-Based Solutions

5.3.1. State of the Art. In the second cluster, we can find DSPs and GPPs. These solutions are known to be flexible, as they are programmable, but they are less efficient. On one hand, GPPs intend to cover the greatest number of potential application domains; their architectures have evolved over the years. Hence, they offer, in their current versions, so-called multimedia features for the treatment of data, which means that part of the architecture is dedicated to regular data processing of a large volume, such as sound, images, and video. On the other hand, a digital signal processor presents better performance compared to GPPs, even if they have much in common. DSP is a special type of microprocessor that incorporates a set of special functions. These functions are designed to make particularly efficient digital signal processing operations.

For example, it is possible to adjust a digital processing function in real time according to some criteria of signal changes. This could be an adaptive filter for the signal processing, hence providing the possibility of adaptive algorithms implementation. However, a programming delay may occur if there is a resource conflict or in the case of a rupture sequence. In addition, DSPs present an expensive solution with a high size and a lot of power dissipation.

Several works have been done based on DSP and one of them combined DSP with GPP, aiming at gaining flexibility as well as accuracy. For example, Lizondo et al. [2012] opted for a low-cost solution that was implemented on a dsPIC (the comercial name of Digital Signal Controller from Microchip) microcontroller chip and with the use of Matlab as a software tool. In the latter, they augmented the system accuracy, since the false acceptance rate and the false rejection rate became 8% and 12%, respectively, while optimizing the algorithm to enhance speed and memory usage. In order to gain in the memory usage of the feature extraction process, the authors opted for hard-coded subroutines in the program memory such as Hamming windows, filter-bank, and DCT kernels. In addition, Manikandan et al. [2011], Suryawanshi and Ganorkar [2014], and Hegde [2009] focused on accuracy of the system as well as efficiency. Manikandan et al. [2011] tried to minimize the memory requirements of the feature extraction kernel of the process by using the cochlear filters, which are considered robust in noisy environments. Thus the memory used is 511KB and 284KB for the MFCC and cochlear features, respectively. But these optimizations are at the expense of the computation cost.

Table VI. Speech/Speaker Recognition Acceleration Based on DSP

Ref.	Speech/Speaker Recognition	Algo. F. E.	Algo. P. M.	Accuracy	Extra features
[Kao and Rajasekaran 2000]	Speech	N/A	HMM	Embedded speech recognition system	-Flexibility (changing vocabulary) -Low cost
[Hegde 2009]	Speaker Ident./Verif.	MFCC	Discrete Cosine Transform	Efficiency as well as accuracy are achieved.	—
[Manikandan et al. 2011]	Speech	MFCC	SVM	-93.33% for MFCC features -89.67% for C. Filter banks.	-Designated to handheld devices -Cochlear Filter Banks algorithm added to F.E.
[Lizondo et al. 2012]	Speaker/ Verif.	MFCC	N/A	-False acceptance rate: 8% -False rejection rate: 12%	-Low cost -Increase performance
[Suryawanshi and Ganorkar 2014]	Speech	MFCC	DWT	-More than 90% -Precision between 70 and 80%	—

Furthermore, Suryawanshi and Ganorkar [2014] improved the overall performance of the system by means of highly accurate and precise results. Kao and Rajasekaran [2000] tried to merge the use of GPP with DSP to achieve flexibility by the use of a unlimited vocabulary and the possibility of suiting several recognition contexts. For that, and based on the computational requirements of the whole process, they proposed to divide it into a computation-intensive part, with low memory requirements implemented on a DSP, and a low computation grammar part, with large memory requirements implemented on a GPP. Thus the models are treated on the GPP and transmitted to the DSP while maintaining a minimum of interaction between both platforms.

More details are presented in Table VI, where the recognition type, the algorithm used for features extraction (Algo. F. E.), the algorithm used for pattern matching (Algo. P. M.), the accuracy, and extra features are given.

5.3.2. General Trend Analysis. For the aforementioned research works, several systems aimed to implement on DSP/GPP platforms flexible speech or speaker recognition applications. Most of them used the dedicated hardware solution to solve the problem of the most computationally intensive part in the process. Thus, during the back-end search stage of recognition, the output probability calculation was the node of the process that has higher priority for acceleration. We can notice that two works, among five, target speaker recognition. Indeed, domain-specific processors such as digital signal processors are suitable for speech signal analysis. Therefore, the main contribution was on accuracy rather than on execution time. Indeed, it was possible to adjust the characteristics of the target application, such as increasing the database size (higher number of speakers), changing the language, and so on.

However, DSP-based designs provide a limited speed for data processing due to the use of special memory architectures that are able to fetch multiple data and/or instructions at the same time. Consequently, they are susceptible to arithmetic saturation.

5.3.3. Possible Future Directions. Several factors, such as time and health condition, make a person's voice change. Thus, utterances in training and testing can differ considerably. Highly variant input utterances create challenges for speaker recognition systems to attain efficiency. Indeed, flexibility as well as accuracy are necessary for such systems, and these technologies are still an active area of research. Due to the increasing need for speaker recognition applications, several flexible solutions need to be improved in order to increase their granularity and become more robust and accurate.

Despite the fact that DSP-based solutions provide better performance (in terms of execution time and memory demands) than GPP, they do not fulfill real-time requirements. In addition, energy consumption and area issues are only partially

revolved by these hardware architectures. Without loss of flexibility, these platforms need to be more aware of embedded domain constraints.

5.4. Heterogeneous Systems for Speaker Recognition

A hardware solution is considered efficient for low-power devices compared to general-purpose processor-based solutions. Some works tried to combine both advantages of the dedicated hardware as well as GPP by making a hybrid solution combining these two platforms. Consequently, a pattern matching kernel is more suitable to be mapped to a hardware platform, whereas less computationally heavy kernels would be mapped to GPP. This makes a hybrid solution combining both advantages.

Wang et al. [2011] implemented the speaker recognition process on a hybrid solution combining an FPGA and an ARM (Advanced RISC Machines) processor. Although they reached approximately 90% of the identification rate, they aimed to accelerate the training part of the process on the FPGA, although it is not considered the most computationally intensive part.

Vargas et al. [2001] achieved the real-time requirements for speech recognition by the limiting the state numbers in the HMM algorithm. A Viterbi node was implemented on the dedicated FPGA, and other operations, such as a signal analysis step, were left to a Motorola 56002. In addition Lee et al. [2007] proposeda high-performance co-processor with dedicated hardware for a distance calculation algorithm, which was considered the most computation-intensive operation. This part was implemented on a ASIC-based approach using a 0.18μm technology node, while other operations were left to the ARM7 processor. Li et al. [2009] tried to achieve the best tradeoff between GPP and dedicated hardware solutions. Hence, they considered a dedicated hardware co-processor for the output probability calculation, which is considered the most computation-intensive part. In addition, they used a microcontroller unit for other non-computation-intensive operations such as MFCC feature extraction, Viterbi decoding, and system control tasks. The co-processor was implemented using FPGA, and the communication between the MCU (Micro Controller Unit) and the dedicated hardware was assured by the SRAM (Static Random Access Memory) interface to facilitate control of the microcontroller.

For the aforementioned works, the authors tried to combine the advantages of using both GPP and a dedicated hardware design. Such a hybrid solution aims at achieving low power consumption while maintaining flexibility and performance of the system.

5.5. Performance-Flexibility Tradeoff

As mentioned in Section 4, ASIPs as well as reconfigurable processors represent a hardware solution to meet the efficiency-flexibility tradeoff [Chattopadhyay et al. 2008].

In fact, ASIPs [Henkel 2003] are customized to particular applications, thereby combining performance and energy efficiency of dedicated hardware solutions with flexibility. These processors are smaller and simpler than their general-purpose counterparts, able to run at higher clock frequencies, and are more energy efficient. The most powerful approach for designing ASIPs is based on the specification of the architecture by means of an Architecture Description Language (ADL), specifically aimed at defining programmable architectures at a higher level of abstraction compared to hardware description languages, like VHDL (VHSIC Very High Speed Integrated Circuit Hardware Description Language) and Verilog.

Reconfigurable processors are perhaps better to address this tradeoff. With advances in their design and use, they are attracting more attention from embedded hardware designers. In particular, we find reconfigurable ASIPs (rASIPs) [Jzwiak et al. 2010]. This class of architecture combines the programmability of ASIPs with the post-fabrication hardware flexibility of re-configurable structures.

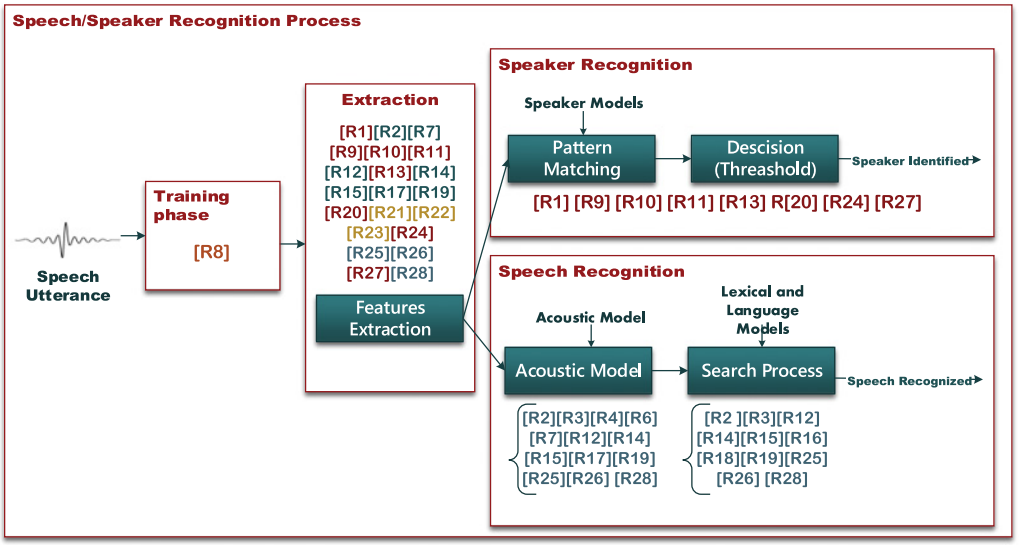


Fig. 8. Speech/speaker recognition process with different implementation works. Legend: R1: Jia-Ching Wang et al. [2014], R2: Nedevschi et al. [2005], R3: Pihl et al. [1996], R4: Bapat et al. [2013], R5: Buitrago et al. [2013], R6: Chen et al. [2011], R7: Cheng et al. [2009], R8: Wang et al. [2011], R9: Ramos-Lara et al. [2013], R10: Ramos-Lara et al. [2009], R11: EhKan et al. [2011], R12: He et al. [2013], R13: Li et al. [2012], R14: Yoshizawa et al. [2006], R15: Han et al. [2003], R16: Bourke and Rutenbar [2008], R17: Lakshmi and Rao [2013], R18: Lin and Rutenbar [2009], R19: Vargas et al. [2001], R20: Sarkar and Saha [2010], R21: Ehkan et al. [2015], R22: Wang et al. [2002], R23: Vu et al. [2010], R24: Lizondo et al. [2012], R25: Manikandan et al. [2011], R26: Suryawanshi and Ganorkar [2014], R27: Hegde [2009], R28: Kao and Rajasekaran [2000]. Red references: works on speaker recognition. Green references: works on speech recognition. Yellow references: works on accelerating extraction phase. Orange references: works on accelerating training phase.

6. DISCUSSION AND FUTURE INSIGHTS

Based on the classification and comparison of the considered works in this survey, there are several conclusions to be drawn. Indeed, many research works concerning the field of speaker recognition and speech recognition were presented. For all the previously cited implementations, many hardware solutions, either ASIC or FPGA based, were used for a possible optimization of computational needs as well as memory access demands targeting speech/speaker recognition applications.

In Figure 8, we gathered information to recapitulate which implementation attempted the acceleration of which algorithmic block and exactly which part of the speech/speaker recognition process. In this graph, references with red refer to works based on speaker recognition and that may target both the extraction part as well as the classification part of the process. Green refers to works focusing on the speech recognition process. In addition, yellow refers to implementations that attempt to accelerate only the extraction phase. Finally, orange refers to works accelerating the training phase that can be used not only for speech recognition but also for speaker recognition applications.

If we recall Figure 6, and add all the considered references according to the clusters mentioned earlier in the survey, then we can plot a flexibility performance chart as illustrated in Figure 9. This figure depicts the previously mentioned works and on which platform (DSP, GPU, ASIC, FPGA, ASIP) they are based. We note that references with red refer to speaker recognition applications, while green refers to speech recognition applications. Many conclusions can be drawn from these charts.

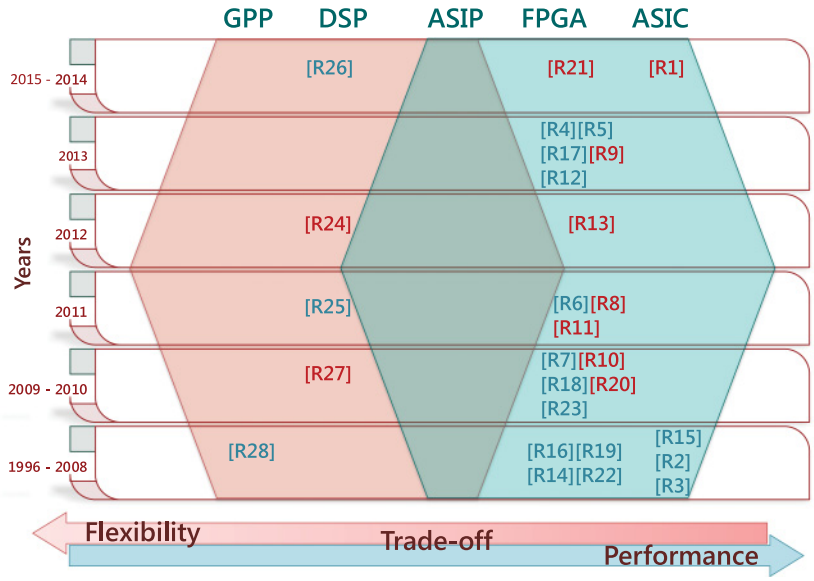


Fig. 9. Selected works scaled for appearance date, flexibility, and efficiency. Legend: R1: Jia-Ching Wang et al. [2014], R2: Nedevschi et al. [2005], R3: Pihl et al. [1996], R4: Bapat et al. [2013], R5: Buitrago et al. [2013], R6: Chen et al. [2011], R7: Cheng et al. [2009], R8: Wang et al. [2011], R9: Ramos-Lara et al. [2013], R10: Ramos-Lara et al. [2009], R11: EhKan et al. [2011], R12: He et al. [2013], R13: Li et al. [2012], R14: Yoshizawa et al. [2006], R15: Han et al. [2003], R16: Bourke and Rutenbar [2008], R17: Lakshmi and Rao [2013], R18: Lin and Rutenbar [2009], R19: Vargas et al. [2001], R20: Sarkar and Saha [2010], R21: Ehkan et al. [2015], R22: Wang et al. [2002], R23: Vu et al. [2010], R24: Lizondo et al. [2012], R25: Manikandan et al. [2011], R26: Suryawanshi and Ganorkar [2014], R27: Hegde [2009], R28: Kao and Rajasekaran [2000]. Red references: works on speaker recognition. Green references: works on speech recognition.

First, it is important to notice that the number of performance-based solutions is much bigger than flexibility-based ones (23 against 5). This first result confirms that speaker/speech recognition systems are compute-intensive applications, particularly when real-time and embedded domain constrains are added. However, with the advances in DSPs performance, works using these platforms appeared in 2009, thus benefiting from flexibility to look for more accuracy in the results. The use of DSPs can also be used due to the fast memory access, inherent to these platforms architecture.

Second, we notice that works on accelerating speaker recognition appeared in 2009, compared to speech recognition, which appeared in 1996. As for the importance of speaker recognition, it is noteworthy that speaker identity is the only biometric that may be easily tested remotely through the existing infrastructure, namely the telephone network. With the growing number of mobile telephones, speaker recognition will become more popular in the future. In terms of deployment, speaker recognition is in its early stages. Thus, accelerating the process and meeting real-time constrains will definitely become a must in the near future.

Last but not least, there is a dearth of applications that address useful tradeoffs. On one side, GPP and DSP allow energy efficiency as well as a reduced area. On the other side ASICs are well used for a specific task that improves the performance and not the flexibility. FPGA offers also a certain degree of performance as well as flexibility due to their reconfigurable aspect.

The used hardware platforms can be designed or used in a way that can significantly improve the overall system performance by reducing memory access demands. In Table VII, we gathered information related to techniques used in order to reduce the

Table VII. Techniques for Memory Access Improvements. Legend: R1: Jia-Ching Wang et al. [2014], R2: Nedevschi et al. [2005], R3: Pihl et al. [1996], R4: Bapat et al. [2013], R5: Buitrago et al. [2013], R6: Chen et al. [2011], R7: Cheng et al. [2009], R8: Wang et al. [2011], R9: Ramos-Lara et al. [2013], R10: Ramos-Lara et al. [2009], R11: EhKan et al. [2011], R12: He et al. [2013], R13: Li et al. [2012], R14: Yoshizawa et al. [2006], R15: Han et al. [2003], R16: Bourke and Rutenbar [2008], R17: Lakshmi and Rao [2013], R18: Lin and Rutenbar [2009], R19: Vargas et al. [2001], R20: Sarkar and Saha [2010], R21: Ehkan et al. [2015], R22: Wang et al. [2002], R23: Vu et al. [2010], R24: Lizondo et al. [2012], R25: Manikandan et al. [2011], R26: Suryawanshi and Ganorkar [2014], R27: Hegde [2009], R28: Kao and Rajasekaran [2000]

	Ref.	Fixed Point Arithmetic	Pipelining	Data-packed Mechanism	On Chip Memory	Double Buffering	Cochlear Filter	Hard Coded
ASIC	[R3]	Yes	Yes	—	—	—	—	—
	[R15]	—	Yes	—	—	—	—	—
	[R2]	—	—	—	Yes	—	—	—
	[R1]	—	—	—	—	—	—	—
FPGA/Speaker	[R10]	Yes	Yes	—	—	—	—	—
	[R20][R13][R9]	—	—	—	—	—	—	—
	[R8]	—	—	Yes	—	—	—	—
	[R11]	—	Yes	—	—	—	—	—
	[R21]	—	Yes	—	—	—	—	—
FPGA/Speech	[R19][R4][R12][R5]	—	—	—	—	—	—	—
	[R22]	—	Yes	—	—	—	—	—
	[R14]	Yes	Yes	—	—	—	—	—
	[R16]	—	Yes	—	—	—	—	—
	[R7]	—	Yes	—	—	—	—	—
	[R18]	—	Yes	—	—	Yes	—	—
	[R23]	Yes	Yes	—	—	—	—	—
	[R6]	Yes	Yes	—	—	—	—	—
	[R17]	Yes	—	—	—	—	—	—
DSP	[R28][R27][R26]	—	—	—	—	—	—	—
	[R25]	—	—	—	—	—	Yes	—
	[R24]	—	—	—	—	—	—	Yes

memory bandwidth or memory access demands by speaker/speech recognition applications. The most common one employed is the pipeline, which is used essentially for the feature extraction node. A fixed-point arithmetic format is used instead of a floating point because of its capability to reduce the memory bandwidth. It also reduces the system memory requirements because of the optimum required data width. In addition, the on-chip memory method, which consists of integrating several memory blocks (Flash/SRAM) on the same chip, reduces the number of memory accesses. However, this technique may not be well used because speaker/speech recognition systems require higher memory sizes. For the hard-coded technique, constant values are used, thus avoiding memory access but at the expense of flexibility.

For the aforementioned works in the table, we can notice that four of five fixed-point-based solutions mentioned several methods to reduce memory access requirements. The most common methods for these four works are fixed-point arithmetic and pipelining.

Furthermore, for reconfigurable-based solutions (FPGA), 4 of 7 works that treated speaker recognition and 8 of 12 works that treated speech recognition implemented techniques for reducing memory demands. We can also mention that the most-used ones are pipelining and fixed point, while the least-used ones are data packing and double buffering. If we recall the dataflow graph of Figure 3.2.2 shown in Section 3.2.2, then the pipelining technique is inherent to the graph structure, especially for features extraction. Since pipelining can be implemented only in

reconfigurable or fixed (ASIC-based) platforms, any implementation on such platforms not using pipelining shows poor performance in terms of memory access.

Finally, for programmable solutions, such as using DSP as a hardware platform, two of five works tried to reduce memory bandwidth using either cochlear filter or hard-coded techniques. We note that these flexible solutions may offer only a small improvement rate in terms of memory access. Thus, the overall system performance is not that important.

All of these concluding remarks about the state of the art on hardware architectures for speaker recognition applications lead to the projection of new research trends that can be undertaken in the future. Indeed, reconfigurable processors can play an important role in the fast-evolving research of speech/speaker recognition due to its adaptability.

7. CONCLUSION

A speaker recognition system can be considered a promising biometric solution that can be widely used in embedded systems. The constraints of the embedded domain, together with the high computation amount due to speech processing, lead to the necessity of using the appropriate underlying hardware architecture in order to meet the requirements. These technologies vary in terms of flexibility and performance.

In this article, a survey of state-of-the-art research in hardware architectures for speaker recognition applications is given. Due to the similarities of some process phases between speaker and speech recognition, we also added the research done for speech recognition applications. As a conclusion, one can notice that 64% (18 of 28) of the surveyed works concern speech recognition. In addition, FPGA is the most popular platform for acceleration (19 of 28). This is probably due to their reconfiguration capability and the achievable speedup, in addition to pipelining implementation capabilities. Indeed, 10 works of 19 applied the pipelining technique. Furthermore, DSP is used less (5 of 28) due to their high cost and high energy consumption. Last, ASICs are the least used (only 4 of 28), as they are not flexible. Despite the achievable flexibility/performance tradeoff, ASIP-/rASIP-based solutions are missing. This is probably due to the fact that they belong to an upcoming trend, thus leading to future research that can be undertaken.

Indeed, based on the previous work presented in this article, it is shown that a successful implementation of speaker recognition system needs a platform solution that must answer performance as well as memory access demand issues. Consequently, a new proposed platform of an ASIP/rASIP/reconfigurable processor design could be the answer for embedded speaker recognition systems. The reason is that the needed accuracy level can be adjusted using a flexible, and thus programmable, technology. At the same time, a customized pipelined architecture and instruction set would help answer computation, memory, cost, and energy demands. With the major research and commercial advances in the field of ASIP and reconfigurable processors design automation, these solutions would become easy to design and use.

ACKNOWLEDGMENTS

The authors thank the reviewers for their valuable comments and suggestions to improve the quality of the article, although they may not agree with all of the interpretations of the survey. The authors also thank Prof. Dr.-Ing. Jeronimo Castrillon of Cfaed, TU Dresden, for his constructive comments and continuous help.

REFERENCES

2015a. Agnito Commercial product. Retrieved January 2015 from http://www.agnitio-corp.com/products/commercial/voice-authentication.

2015b. Agnito Government application. Retrieved January 2015 from http://www.agnitio-corp.com/products/government/id-verification.

2015a. Agnito Voiceprint identification. Retrieved January 2015 from http://www.agnitio-corp.com/products/government/voiceprint-identification.

2015. IdentityXs Authenticator: identification credentials. Retrieved January 2015 from http://www.identityx.com/markets/identification-credentials/.

2015. IdentityXs Authenticator: Logical Acess. Retrieved January 2015 from http://www.identityx.com/markets/logical-access/ .

2015. Nuance Automatic Speech Recognition. Retrieved January 2015 from http://www.nuance.com/for-business/automatic-speech-recognition/index.htm.

2015b. VoiceGrid: Automated Voice Biometric System. Retrieved January 2015 from http://speechpro.com/product/biometric/voicegridnation.

2015c. VoiceGrid RT: Sophisticated Distributed Solution for Real-time Speaker Identification. Retrieved January 2015 from http://speechpro.com/product/biometric/voicegridrt.

2015. Voxomos innovations in speech. Retrieved March 2015 from http://voxomos.com/Default.aspx.

Naufal Alee, Phaklen Ehkan, R. Badlishah Ahmad, and Naseer Sabri. 2013. Speaker recognition system: Vulnerable and challenges. *Int. J. Eng. Techno.* 5, 4 (2013), 3191–3195.

Mohammad Azhari. 2011. *A CUDA Based Parallel Implementation of Speaker Verification System*. Ph.D. Dissertation. Eastern Mediterranean University (EMU).

Ojas A. Bapat, Richard M. Fastow, and Jens Olson. 2013. Acoustic coprocessor for HMM based embedded speech recognition systems. *IEEE Trans. Cons. Electron.* 59, 3 (2013), 629–633.

Homayoon Beigi. 2011. *Fundamentals of Speaker Recognition*. Springer.

Mohamed Faouzi BenZeghiba. 2005. *Joint Speech and Speaker Recognition*. Ph.D. Dissertation. École Polytechnique Fédérale de Lausanne.

Samarth Bharadwaj, Mayank Vatsa, and Richa Singh. 2014. Biometric quality: A review of fingerprint, iris, and face. *EURASIP J. Image Vid. Process.* 2014, 1 (2014), 1–28.

Patrick J. Bourke and Rob A. Rutenbar. 2008. A low-power hardware search architecture for speech recognition. In *INTERSPEECH*. 2102–2105.

Herve A. Bourlard and Nelson Morgan. 1994. *Connectionist Speech Recognition: A Hybrid Approach*. Vol. 247. Springer.

Kevin W. Bowyer, Karen P. Hollingsworth, and Patrick J. Flynn. 2013. A survey of iris biometrics research: 2008–2010. In *Handbook of Iris Recognition*. Springer, 15–54.

Byron Buitrago, Johnny Aguirre, Andrés Benavides, and Marcela Rivera. 2013. A Hardware Accelerator Design Process for Speech Recognition Application. Retrieved from http://weblidi.info.unlp.edu.ar/WorldComp2013-Mirror/p2013/ESA2562.pdf.

William M. Campbell, Joseph P. Campbell, Douglas A. Reynolds, Elliot Singer, and Pedro A Torres-Carrasquillo. 2006. Support vector machines for speaker and language recognition. *Comput. Speech Lang.* 20, 2 (2006), 210–229.

Anupam Chattopadhyay. 2013. Ingredients of adaptability: A survey of reconfigurable processors. *VLSI Des.* 2013, Article 10 (Jan. 2013). DOI:http://dx.doi.org/10.1155/2013/683615

Anupam Chattopadhyay, Rainer Leupers, Heinrich Meyr, and Gerd Ascheid. 2008. *Language-Driven Exploration and Implementation of Partially Re-configurable ASIPs*. Springer Science & Business Media.

Tao Chen, Jiawei Zheng, Xingsi Zhang, Shengchang Cai, and Yun Chen. 2011. A hardware accelerator for speech recognition applications. In *Proceedings of the 2011 IEEE 9th International Conference on ASIC (ASICON'11)*. 760–763. DOI:http://dx.doi.org/10.1109/ASICON.2011.6157316

Octavian Cheng, Waleed Abdulla, and Zoran Salcic. 2009. Speech recognition system for embedded real-time applications. In *2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. 118–122. DOI:http://dx.doi.org/10.1109/ISSPIT.2009.5407487

Kresimir Delac and Mislav Grgic. 2004. A survey of biometric recognition methods. In *Proceedings of the 46th International Symposium on Electronics in Marine, 2004*. IEEE, 184–193.

Phaklen EhKan, Timothy Allen, and Steven F. Quigley. 2011. FPGA implementation for GMM-based speaker identification. *Int. J. Reconfig. Comput.* 2011, Article 3 (Jan. 2011). DOI:http://dx.doi.org/10.1155/2011/420369

P. Ehkan, F. F. Zakaria, MNM Warip, Z. Sauli, and M. Elshaikh. 2015. Hardware implementation of MFCC-based feature extraction for speaker recognition. In *Advanced Computer and Communication Engineering Technology*. Springer, 471–480.

Amin Fazel and Shantanu Chakrabartty. 2011. An overview of statistical pattern recognition techniques for speaker verification. *IEEE Circ. Syst. Mag.* 11, 2 (2011), 62–81.

Keinosuke Fukunaga. 1990. *Introduction to Statistical Pattern Recognition*. Academic Press.

Wang Fuqiu, Wei-Qiang Zhang, and Liu Jia. 2012. GPU accelerated GMM supervectors for speaker and language recognition. In *Proceedings of the 2012 IEEE 11th International Conference on Signal Processing (ICSP)*, Vol. 1. IEEE, 536–539.

Sadaoki Furui. 2005. 50 years of progress in speech and speaker recognition. *SPECOM 2005, Patras* (2005), 1–9.

Tamer S. Gaafar, Hitham M. Abo Bakr, and Mahmoud I. Abdalla. 2014. An improved method for speech/speaker recognition. In *Proceedings of the 2014 International Conference on Informatics, Electronics & Vision (ICIEV)*. IEEE, 1–5.

Wei Han, Kwok-Wai Hon, Cheong-Fat Chan, Tan Lee, Chiu-Sing Choy, Kong-Pang Pun, and Pak-Chung Ching. 2003. An HMM-based speech recognition IC. In *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003 (ISCAS'03)*, Vol. 2. IEEE, II–744.

Guangji He, Yuki Miyamoto, Kumpei Matsuda, Shintaro Izumi, Hiroshi Kawaguchi, and Masahiko Yoshimoto. 2013. A 40-NM 54-MW 3×-real-time VLSI processor for 60-kWord continuous speech recognition. In *Proceedings of the 2013 IEEE Workshop on Signal Processing Systems (SiPS)*. IEEE, 147–152.

Sneha Hegde. 2009. *Speaker Recognition Using TMS320C6713DSK*. Ph.D. Dissertation. University of Mumbai.

J. Henkel. 2003. Closing the SoC design gap. *Computer* 36, 9 (Sep. 2003), 119–121. DOI:http://dx.doi.org/10.1109/MC.2003.1231200

Anil K. Jain, Arun Ross, and Salil Prabhakar. 2004. An introduction to biometric recognition. *IEEE Trans. Circ. Syst. Vid. Technol.* 14, 1 (2004), 4–20.

J.-C. Jia-Ching Wang, L.-X. Lian, Y.-Y. Lin, and J.-H. Zhao. 2014. VLSI design for SVM-based speaker verification system. *IEEE Trans. Syst.* 99 (2014), 1. DOI:http://dx.doi.org/10.1109/TVLSI.2014.2335112

Lech Jzwiak, Nadia Nedjah, and Miguel Figueroa. 2010. Modern development methods and tools for embedded reconfigurable systems: A survey. *Integration* 43, 1 (2010), 1–33.

Yu-Hung Kao and Periagaram K. Rajasekaran. 2000. A low cost dynamic vocabulary speech recognizer on a GPP-DSP system. In *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'00)*, Vol. 6. 3215–3218 vol.6. DOI:http://dx.doi.org/10.1109/ICASSP.2000.860084

Tomi Kinnunen and Haizhou Li. 2010. An overview of text-independent speaker recognition: From features to supervectors. *Speech Commun.* 52, 1 (2010), 12–40.

Manish Kumar. 2015. *Performance Analysis of LPC and MFCC Techniques in Automatic Speech Recognition*. Ph.D. Dissertation. Thapar University Patiala.

V. Rajya Lakshmi and P. Venkat Rao. 2013. Hardware software codesign of automatic speech recognition system for embedded real-time applications. *Int. J. Instrum. Electr. Electron. Eng.* 1 (Sep. 2013), pp. 7–10.

C. H. Lee, L. R. Rabiner, R. Pieraccini, and J. G. Wilpon. 1990. Acoustic modeling for large vocabulary speech recognition. *Comput. Speech Lang.* 4, 2 (1990), 127–165.

Chin-Hui Lee, Frank K Soong, and Kuldip Paliwal. 2012. *Automatic Speech and Speaker Recognition: Advanced Topics*. Vol. 355. Springer Science & Business Media.

Edward A. Lee and David G. Messerschmitt. 1987. Synchronous data flow. *Proc. IEEE* 75, 9 (1987), 1235–1245.

Peng Lee, Ming Dong, Weiqian Liang, and Runsheng Liu. 2007. Design of speech recognition co-processor for the embedded implementation. In *Proceedings of the IEEE Conference on Electron Devices and Solid-State Circuits, 2007 (EDSSC'07)*. IEEE, 1163–1166.

Jingjiao Li, Dong An, Lili Lang, and Dan Yang. 2012. Embedded speaker recognition system design and implementation based on FPGA. *Proc. Eng.* 29 (2012), 2633–2637. DOI:http://dx.doi.org/10.1016/j.proeng.2012.01.363 2012 International Workshop on Information and Electronics Engineering.

Peng Li, Hua Tang, and Weiqian Liang. 2009. Low power embedded speech recognition system based on a MCU and a coprocessor. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'09)*. IEEE, 625–628.

Edward C. Lin and Rob A. Rutenbar. 2009. A multi FPGA 10x real time high speed search engine for a 5000-word vocabulary speech recognizer. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*. ACM, 83–92.

Chuan Liu. 2009. cuHMM: A CUDA implementation of hidden Markov model training and classification. *The Chronicle of Higher Education* (2009).

Maximiliano Lizondo, Pablo D. Agüero, Alejandro J. Uriz, Juan C. Tulli, and Esteban L. Gonzalez. 2012. Embedded speaker verification in low cost microcontroller. *Congreso Argentino de Sistemas Embebidos* (2012), 128–133.

Lukáa Machlica, Jan Vaněk, and Zbyněk Zajíc. 2011. Fast estimation of gaussian mixture model parameters on gpu using cuda. In *Proceedings of the 2011 12th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*. IEEE, 167–172.

John Makhoul, Salim Roucos, and Herbert Gish. 1985. Vector quantization in speech coding. *Proc. IEEE* 73, 11 (1985), 1551–1588.

Amruta Anantrao Malode and Shashikant Sahare. 2012. Advanced speaker recognition. *Int. J. Adv. Eng. Technol.* 4, 1 (2012), 443–455.

J. Manikandan, B. Venkataramani, K. Girish, H. Karthic, and V. Siddharth. 2011. Hardware implementation of real-time speech recognition system using TMS320C6713 DSP. In *Proceedings of the 2011 24th International Conference on VLSI Design (VLSI Design)*. 250–255. DOI: http://dx.doi.org/10.1109/VLSID.2011.12

Limkar Maruti, Rao B. Rama, and Sagvekar Vidya. 2012. Speaker recognition using VQ and DTW. *IJCA Proc. ICACACT* 3 (Aug. 2012), 18–20.

Christian Mller. 2007. *Speaker Classification I: Fundamentals, Features, and Methods*. Lecture Notes in Computer Science, Vol. 4343. Springer.

JJVI Naik. 1990. Speaker verification: A tutorial. *IEEE Commun. Mag.* 28, 1 (1990), 42–48.

Sergiu Nedevschi, Rabin K. Patra, and Eric A. Brewer. 2005. Hardware speech recognition for user interfaces in low cost, lowpower devices. In *Proceedings of the 42nd Design Automation Conference 2005*. 684–689. DOI: http://dx.doi.org/10.1109/DAC.2005.193899

Richard D. Peacocke and Daryl H. Graf. 1990. An introduction to speech and speaker recognition. *Computer* 23, 8 (1990), 26–33.

Johnny Pihl, Torbjorn Svendsen, and Magne H Johnsen. 1996. A VLSI implementation of PDF computations in HMM based speech recognition. In *Proceedings of the 1996 IEEE TENCON, Digital Signal Processing Applications (TENCON'96)*, Vol. 1. IEEE, 241–246.

Marco Platzner and Norbert Wehn. 2010. *Dynamically Reconfigurable Systems: Architectures, Design Methods and Applications*. Springer Science & Business Media.

Lawrence Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286.

Lawrence Rabiner and Ronald Schafer. 2010. *Theory and Applications of Digital Speech Processing* (1st ed.). Prentice Hall, Upper Saddle River, NJ.

Lawrence R Rabiner and Biing-Hwang Juang. 1993. *Fundamentals of Speech Recognition*. Vol. 14. PTR Prentice Hall, Englewood Cliffs, NJ.

Rafael Ramos-Lara, Mariano López-García, Enrique Cantó-Navarro, and Luís Puente-Rodriguez. 2009. SVM speaker verification system based on a low-cost FPGA. In *Proceedings of the International Conference on Field Programmable Logic and Applications, 2009 (FPL'09)*. IEEE, 582–586.

Rafael Ramos-Lara, Mariano López-García, Enrique Cantó-Navarro, and Luís Puente-Rodriguez. 2013. Real-time speaker verification system implemented on reconfigurable hardware. *J. Sign. Process. Syst.* 71, 2 (2013), 89–103.

Douglas A. Reynolds. 1995. Automatic speaker recognition using gaussian mixture speaker models. *Lincoln Lab. J.* (1995), 173–192.

Douglas A. Reynolds and L. P. Heck. 1991. Integration of speaker and speech recognition systems. In *Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing 1991 (ICASSP'91)*. IEEE, 869–872.

Zia Saquib, Nirmala Salam, Rekha P. Nair, Nipun Pandey, and Akanksha Joshi. 2010. A survey on automatic speaker recognition systems. In *FGIT-SIP/MulGraB*. 134–145.

Gourav Sarkar and Goutam Saha. 2010. Real time implementation of speaker identification system with frame picking algorithm. *Proc. Comput. Sci.* 2 (2010), 173–180.

Nilu Singh, R. A. Khan, and Raj Shree. 2012. Applications of speaker recognition. *Proc. Eng.* 38 (2012), 3122–3126. DOI: http://dx.doi.org/10.1016/j.proeng.2012.06.363

Umarani J. Suryawanshi and S. R. Ganorkar. 2014. Hardware implementation of speech recognition using MFCC and euclidean Distance. *Int. J. Adv. Res. Electr. Electron. Instrum. Eng* 3, 08 (2014), 11248–11254.

Roberto Togneri and Daniel Pullella. 2011. An overview of speaker identification: Accuracy and robustness issues. *IEEE Circ. Syst. Mag.* 11, 2 (2011), 23–61.

O. B. Tuzun, M. Demirekler, and K. B. Nakiboglu. 1994. Comparison of parametric and non-parametric representations of speech for recognition. In *Proceedings of the 7th Mediterranean Electrotechnical Conference 1994.* IEEE, 65–68.

Jan Vaněk, Jan Trmal, and Josef V. Psutka. 2011. Optimization of the gaussian mixture model evaluation on GPU. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association 2011 (INTERSPEECH'11)*.

Fabian Luis Vargas, Rubem Dutra Ribeiro Fagundes, and Daniel Burros Junior. 2001. A FPGA-based viterbi algorithm implementation for speech recognition systems. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)*, Vol. 2. IEEE, 1217–1220.

Ngoc-Vinh Vu, Jim Whittington, Hua Ye, and John Devlin. 2010. Implementation of the MFCC front-end for low-cost speech recognition systems. In *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS'10)*. IEEE, 2334–2337.

Jia-Ching Wang, Jhing-Fa Wang, and Yu-Sheng Weng. 2002. Chip design of MFCC extraction for speech recognition. *VLSI J.* 32, 1 (2002), 111–131.

Jhing-Fa Wang, Jr-Shiang Peng, Jia-Ching Wang, Po-Chuan Lin, and Ta-Wen Kuan. 2011. Hardware/software co-design for fast-trainable speaker identification system based on SMO. In *Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC'11)*. 1621–1625. DOI:http://dx.doi.org/10.1109/ICSMC.2011.6083903

Roman V. Yampolskiy and Venu Govindaraju. 2008. Behavioural biometrics: A survey and classification. *Int. J. Biometr.* 1, 1 (2008), 81–113.

Shingo Yoshizawa, Naoya Wada, Noboru Hayasaka, and Yoshikazu Miyanaga. 2006. Scalable architecture for word HMM-based speech recognition and VLSIimplementation in complete system. *IEEE Trans. Circ. Syst. I: Regul. Pap.* 53, 1 (2006), 70–77.

Received March 2015; revised April 2016; accepted July 2016