Microprocessors and Microsystems 34 (2010) 274–284

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

An embedded audio–visual tracking and speech purification system on a dual-core processor platform

Jwu-Sheng Hu, Ming-Tang Lee *, Chia-Hsing Yang

Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, Taiwan, ROC

ARTICLE INFO

Article history:
Available online 4 June 2010

Keywords:
Audio–visual tracking
Dual-core
Embedded processor
Speech enhancement
Microphone array

ABSTRACT

Design of an embedded audio–visual tracking and speech purification system is described in this paper. The system is able to perform human face tracking, voice activity detection, sound source direction estimation, and speech enhancement in real-time. Estimating the sound source directions helps to initialize the human face tracking module when the target changes the direction. The implementation architecture is based on an embedded dual-core processor, Texas Instruments DM6446 platform (Davinci), which contains an ARM core and a DSP core. For speech signal processing, an eight-channel digital microphone array is developed and the associated pre-processing and interfacing features are designed using the Altera Cyclone II FPGA. All the experiments are conducted in a real environment and the experimental results show that this system can execute all the audition and vision functions in real-time.

1. Introduction

Integration of auditory and visual perceptions has become a trend in robots [1] or intelligent human–machine interfaces [2]. For intelligent machines to interact naturally with human, the abilities to understand spoken language and respond to auditory events are necessary since the auditory system can provide useful information about the environment, such as sound source location and the interpretation of the content of the sound sources. However, in real environment, the speech signal is easily corrupted by interference signal such as other talkers or room reverberation. Hence, sound source tracking and speech enhancement are two basic functions of the auditory system. Secondly, sound source tracking is usually followed by visual tracking when the target is within the view scope of the camera. The integrated interface greatly enhances the robustness of human or target tracking. However, the integration also poses a great challenge in technology development.

This paper describes the auditory–visual system architecture, algorithms and implementation on an embedded platform that can be applied to robots or other human–machine interfaces.

The integration of audio and visual perceptions has already been involved in many applications. For example, the SIG robot [3] based on binaural auditory system estimates sound source direction in the horizontal plane using two microphones information and separates sound sources for speech recognition. Then the research team combines visual tracking system into the SIG robot to achieve more robust estimation [4,5]. The SIG system is implemented by distributed processing of five nodes with 1.8 GHz Pentium-IV. The ARMAR III [6] is designed as an labourer in the kitchen. The ARMAR III can recognize and classify sounds with six microphones. Also, it can identify the user face and detect the user gesture with the auditory–visual perception. Suwannathat et al. [7] proposed a mobile robot which can detect the speaker direction by the integration of microphone arrays and an omni-directional camera. The robot also matches human templates to determine whether a person is nearby or not. Wang et al. [8] proposed a PC-based microphone array system which can enhance speech signal and estimate the speaker's direction. The speech enhancement and direction estimation algorithms are implemented on a plugged-in DSP board. The work of [9] evaluated the real-time performance of the proposed algorithms on a desktop PC's (3.0 GHz) for 4 cameras and 12 microphones. Connell et al. tried to demonstrate a small-vocabulary audio–visual ASR on an 1.8 GHz PC [10]. The video processing for capturing and analyzing a human face alone takes 67% of the computing resources. There are also plenty of research efforts dedicated to explore the theoretical foundation of the related problems involved [11–15]. Basically, these works were conducted to support the real-time capability of the proposed algorithms and PCs were used as a standard platform for comparison.

Despite the great effort in studying the audio–visual interface, there hasn't been a report on embedded implementation of this technology. As described before, for applications such as robotics and vehicles, embedded implementation is necessary to meet the cost and size constraints. However, the implementation of a real-time audio–visual interface can be quite different depending on

* Corresponding author.
E-mail address: lhoney.ece95g@nctu.edu.tw (M.-T. Lee).

0141-9331/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.micpro.2010.05.004

the features and algorithms. Usually, it involves several sub-systems and is not easy to design on an embedded platform. For example, to acquire microphone array data requires multi-channel synchronized A/D interface and off-the-shelf acquisition hardware, which are mostly for PC architecture. Further, the real-time constraints of simultaneous audio and video processing on a resource-limited embedded platform also pose some interesting issues in design. In this work, we propose and implement the human face tracking, adaptive speech enhancement, voice activity detection and sound source direction estimation on an embedded dual-core processor platform (Texas Instruments DM6446) [16]. The dual-core processor platform includes an Advanced RISC Machine (ARM) subsystem and a DSP subsystem. All of the audio algorithms are executed on the DSP subsystem, and ARM is used to execute the human face tracking algorithm and to control input/output (I/O) devices. The audio interface includes an eight-channel microphone array. To acquire the microphone signals simultaneously, a special hardware is designed using digital microphones and field programmable gate array (FPGA). The FPGA is responsible for pre-processing the microphone signals and transmits the data to the platform via asynchronous external memory interface (AEMIF).

The paper is organized as the following: Section 2 describes the related algorithms implemented and their associated computation procedure. The multi-channel microphone interface is presented in Section 3. Section 4 shows the implementation architecture and software development. Experiments in real environments to access the real-time performance of the system are explained in Section 5.

2. Implementation methodology

In this section, we introduce the algorithms used in the proposed system. They include the voice activity detection (VAD), the speech enhancement function and the visual (human face) tracking algorithm.

2.1. Voice activity detection

The voice activity detection (VAD) algorithm [17] detects the presence of speech by adjusting itself according to current environmental noise based on the estimation of the long-term spectral envelope (LTSE). The LTSE tracks the spectral envelope using long-term rectangular speech window information. Assume that $\hat{s}(n)$ is the n-th sample of the VAD system input. Then, the J-order LTSE can be defined as:

$$\mathrm{LTSE}_J(k,l) = \max\{\widehat{S}(k,l+j)\}_{j=-J}^{j=+J} \tag{1}$$

where $\widehat{S}(k,l)$ denotes the amplitude spectrum of $\hat{s}(n)$ at frame l and frequency band k. In addition, the decision rule is formulated in terms of the long-term spectral divergence (LTSD). The J-order LTSD is defined as:

$$\mathrm{LTSD}_J(l) = 10\log_{10}\left(\frac{1}{B}\sum_{b=0}^{K-1}\frac{\mathrm{LTSE}_J^2(k,l)}{\widehat{N}^2(k,l-1)}\right) \tag{2}$$

where K means the number of frequency bands. $\widehat{N}^2(k,l-1)$ is the estimated noise spectrum for the band k. Then, the J-order LTSD is used to compare with an adaptive threshold γ to determine the existence of speech signals. The threshold γ is adapted depending on the value of noise energy E:

$$\gamma = \begin{cases} \gamma_0 & E \leqslant E_0 \\ \gamma_0 + \frac{\gamma_1-\gamma_0}{E_1-E_0}(E-E_0) & E_0 < E < E_1 \\ \gamma_1 & E \geqslant E_1 \end{cases} \tag{3}$$

with

$$E = \sum_{b=0}^{K-1}\hat{N}^2(k,l-1)$$

where γ_0 and γ_1 are thresholds when the system is working in the quietest and noisiest conditions respectively. E_0 and E_1 are the corresponding noise energies. The result of VAD is set to one (i.e., speech signal is detected) when the value of LTSD is larger than the value of γ; otherwise, it is set to zero (i.e., no speech signal is detected). Afterwards, the noise spectrum is updated as:

$$\widehat{N}(k,l) = \begin{cases} \psi\widehat{N}(k,l-1) + (1-\psi)\widehat{N}_{\mathrm{Q}}(k,l) & \text{if VAD} = 0 \\ \widehat{N}(k,l-1) & \text{if VAD} = 1 \end{cases} \tag{4}$$

where ψ is the adaption weight and $\widehat{N}_{\mathrm{Q}}(k,l)$ is the average of estimated noise spectrum magnitudes at frequency band k over $(2\mathrm{Q}+1)$ frame:

$$\widehat{N}_{\mathrm{Q}}(k,l) = \frac{1}{2\mathrm{Q}+1}\sum_{q=-\mathrm{Q}}^{\mathrm{Q}}\widehat{N}(k,l+q) \tag{5}$$

2.2. Speech enhancement

This work uses the adaptive beamformer technique to purify the contaminated speech. The overall speech enhancement system architecture can be illustrated as Fig. 1, where $x_m(n)$ represents the m-th microphone received signal and $s_m(n)$ is pre-recorded speech corresponding to the m-th channel. This speech enhancement system is composed by two stages: silent stage and speech stage. The stages are switched by the VAD results. If the result of VAD is equal to zero, which means that no speech exists, the system will be executed in the silent stage. In the silent stage, the input signals ($x_m(n)$) which considered as environmental noise are added with pre-recorded reference signals to train the weighting vector. When the system is switched to speech stage, the trained weighting vector is passed to the lower beamformer to purify the received signals. In addition, the purified signal is passed to the VAD process to extract the speech signal.

The parameter update of the beamformer is executed in the time domain with normalized least mean square (NLMS) based approach [18]. Based on the system architecture, the formulation of microphone array speech purification system can be expressed as the following linear model:

$$e(n) = \hat{\mathbf{x}}(n)^T\mathbf{w} - r(n) \tag{6}$$

where

$$\hat{x}(n) = [\hat{x}_1(n) \quad \cdots \quad \hat{x}_M(n)]^T$$

$$\hat{x}_i(n) = [\hat{x}_i(n) \quad \cdots \quad \hat{x}_i(n-P+1)]^T$$

$$w = [w_{11} \quad \cdots \quad w_{1P} \quad \cdots \quad w_{M1} \quad \cdots \quad w_{MP}]^T$$

where M denotes the number of microphones, P denotes the filter order of each microphone signal, and the superscript T denotes the transpose operation. $\hat{\mathbf{x}}(n)$ is the $MP\times 1$ training signal vector constructed from the linear combination of the pre-recorded speech signal and the online recorded background noise. $r(n)$ denotes the reference signal, which is the pre-recorded speech of microphone 1. $e(n)$ is the unknown estimation disturbance and w is the $MP\times 1$ unknown filter coefficient vector of the beamformer to be estimated. The solution can be approximated iteratively by the recursion:

$$w(n+1) = w(n) + \frac{\mu}{\delta + \|\hat{x}(n+1)\|^2}\hat{x}(n+1)\times[r(n+1) - \hat{x}(n+1)^T w(n)] \tag{7}$$

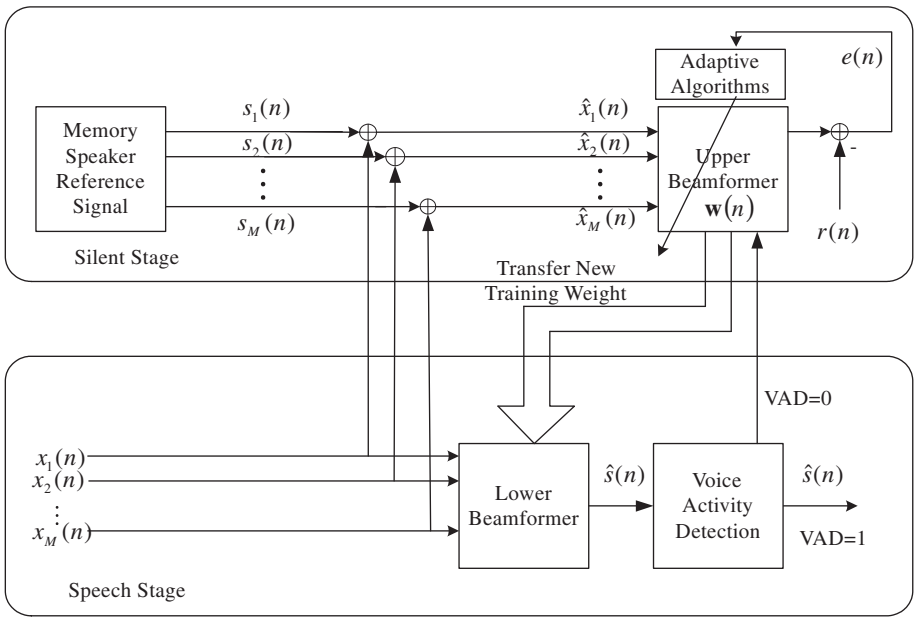

Fig. 1. Speech enhancement system architecture.

where δ is a small positive value in order to keep the denominator positive and $0 < \mu < 2$ should be satisfied for convergence. By calibration using pre-recorded speech signals, this method outperforms other un-calibrated algorithms in real applications [19].

2.3. Sound source tracking

The direction of arrival (DOA) estimation system architecture is illustrated in Fig. 2. It is also separated into two stages by the VAD results, namely, the silent stage and speech stage similar to the adaptive beamformer. In the silent stage, the B significant frequencies $k_{N1}, \ldots, k_{NB}$ will be chosen by comparing the received signals' amplitude spectrum in each frequency to represent the principal frequencies of the non-speech environment (the most significant frequencies of background noise).

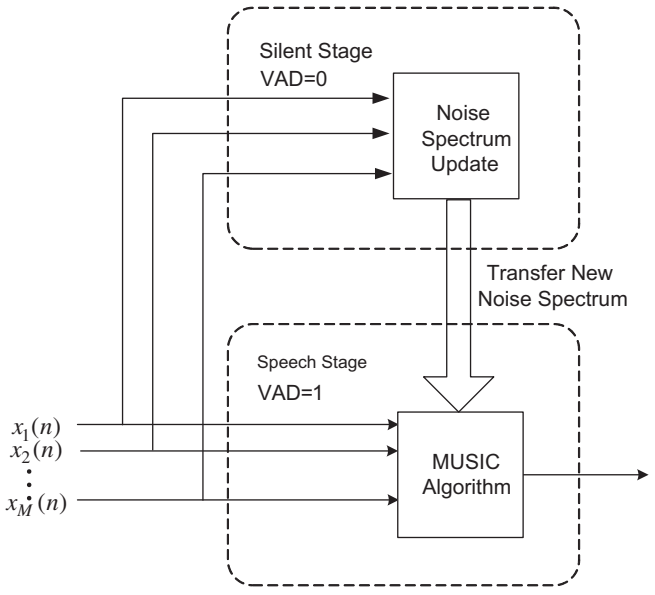

Fig. 2. Sound source detection and estimation system.

Assume that $x_j(n)$, the received signal of the j-th microphone, is utilized to detect the speech. Let $X(k_j, l)$ be the amplitude spectrum of $x_j(n)$ for the frequency band k_j at frame l, for $j = 1, \ldots, B$. The B significant frequencies can be selected as follows:

$$\hat{X}(k_j) = \frac{1}{L}\sum_{l=0}^{L} X(k_j, l) \tag{8}$$

$$\{k_{N1}, \ldots, k_{NB}\} = \left\langle \left\{ \hat{X}(k_1), \ldots, \hat{X}(k_K) \right\} \right\rangle_b \quad K > B$$

where L is the number of frames to be averaged. K denotes the number of frequency bands. $\langle \bullet \rangle_b$ denotes selecting the biggest B values from the elements. When the result of VAD is one, the system will be switched to the speech stage and then the B significant frequencies of the silent stage are transferred to the DOA algorithm for estimating the speakers' directions. The well-known blind DOA estimation algorithm MUSIC is adopted to determine the arrival angle. Since speech signals are wideband, a wideband incoherent MUSIC algorithm [20] with arithmetic mean was implemented in this work. In speech stage, only C significant frequencies $k_{S1}, \ldots, k_{SC}$ are selected for wideband MUSIC algorithm to reduce the computation complexity and the frequencies can be described as

$$\{k_{S1}, \ldots, k_{SC}\} = \left\langle \left\{ \hat{X}(k_1), \ldots, \hat{X}(k_K) \right\} \right\rangle_c \quad K > C > B \tag{9}$$

However, the principal frequencies of speech stage may overlap those of silent stage and shall be removed. This leads to the following new set of frequencies:

$$\{k_{v1}, \ldots, k_{vR}\} = \{k_{S1}, \ldots, k_{SC}\} - \{k_{N1}, \ldots, k_{NB}\} \tag{10}$$

The speakers' directions are determined by finding the peaks of MUSIC spectrum:

$$J(\theta_i) = \frac{1}{\frac{1}{R}\sum_{r=1}^{R} A_i^H(\hat{k}_{vr}) P_N(\hat{k}_{vr}) A_i(\hat{k}_{vr})} \tag{11}$$

where $A_i(k_{vr})$ is the array manifold vector and $P_N(k_{vr})$ is the noise projection matrix. For the detail of the MUSIC implementation, please refer to [18].

Fig. 3. Illustration of the same spatial information with different color distribution for one bin.

2.4. Human face tracking

In our proposed system, we use the spatial-color mean-shift object tracking algorithm [21] to track human face. The main concept of mean-shift tracking is to find the candidate which is the most similar with target image by mean-shift iterations. The principle of mean-shift is to compare the color distribution of candidate region with the color distribution of the model, and to compute the similarity measure, Bhattacharyya coefficient, to observe the variation of gradient of candidate to find the mean-shift vector. Further, mean-shift finds the most similar region or the most possible area of the candidate.

Let the model image has M pixels, the candidate image has N pixels, and the associated color space can be classified into B bins. For example, in RGB color space, if each color is divided into eight intervals, the total number of bins is 512. This paper denotes the model image as $I_x = \{\mathbf{x}_i, \mathbf{c}_{\mathbf{x}i}, b_{\mathbf{x}i}\}_{i=1,\ldots,M}$, where $\mathbf{x}_i$ is the location of pixel i with color feature vector $\mathbf{c}_{\mathbf{x}i}$ which belongs to the $b_{\mathbf{x}i}$-th bin. Similarly, the candidate image can be denoted as $I_y = \{\mathbf{y}_j, \mathbf{c}_{\mathbf{y}j}, b_{\mathbf{y}j}\}_{j=1,\ldots,N}$. The dimension of the color feature vector is d, which is the number of color channels of a pixel for example, in RGB color space, $d = 3$ and $\mathbf{c}_{\mathbf{x}i} = (R_b, G_b, B_b)$.

As shown in Fig. 3, if black[1] and gray belong to the same bin, these two blocks have the same spatiogram, but they have different color patterns. To keep the robustness of color description of the spatiogram, we extend the spatiogram and define a new joint spatial-color model as

$$h_I(b) = \langle n_b, \boldsymbol{\mu}_{P,b}, \boldsymbol{\Sigma}_{P,b}, \boldsymbol{\mu}_{C,b}, \boldsymbol{\Sigma}_{C,b} \rangle, \quad b = 1, \ldots, B \tag{12}$$

where n_b, $\boldsymbol{\mu}_{P,b}$, and $\boldsymbol{\Sigma}_{P,b}$ are the same as the spatiogram proposed by Birchfield and Rangarajan [22]. Namely, n_b is the number of pixels, $\boldsymbol{\mu}_{P,b}$ is the mean vector of pixel locations, and $\boldsymbol{\Sigma}_{P,b}$ is the covariance matrix of pixel locations belonging to the b-th bin. Besides, we also add two additional elements. $\boldsymbol{\mu}_{C,b}$ is the mean vector of the color feature vectors and $\boldsymbol{\Sigma}_{C,b}$ is the associated covariance matrix.

The probability density function (p.d.f.) of the object in the image model can be estimated using kernel density function,

$$p_x(\mathbf{x}, \mathbf{c}_{\mathbf{x}}, b_{\mathbf{x}}) = \frac{1}{M}\sum_{i=1}^{M} K_P(\mathbf{x} - \boldsymbol{\mu}_{P,b(i)}, \boldsymbol{\Sigma}_{P,b(i)}) K_C(\mathbf{c}_{\mathbf{x}} - \boldsymbol{\mu}_{C,b(i)}, \boldsymbol{\Sigma}_{C,b(i)}) \cdot \delta(b_{\mathbf{x}} - b(i)) \tag{13}$$

where $b(i)$ is the color bin to which pixel i belongs to. K_P and K_C are multivariate Gaussian kernel functions and can be regarded as the spatially weighted and color-feature weighted function respectively. In (13), the spatial weighting among the color bins is selected to be a delta function. It is also possible to use a smooth kernel such as Gaussian [21]. Using the concept of the expectation of the estimated kernel density, we can define a new similarity measure function between the model $I_x = \{\mathbf{x}_i, \mathbf{c}_{\mathbf{x}i}, b_{\mathbf{x}i}\}_{i=1,\ldots,M}$ and candidate $I_y = \{\mathbf{y}_j, \mathbf{c}_{\mathbf{y}j}, b_{\mathbf{y}j}\}_{j=1,\ldots,N}$ as

$$\begin{aligned} J(I_x, I_y) = J(\mathbf{y}) &= \frac{1}{N}\sum_{j=1}^{N} p_x(\mathbf{y}_j, \mathbf{c}_{\mathbf{y}j}, b_{\mathbf{y}_j}) \\ &= \frac{1}{NM}\sum_{i=1}^{M}\sum_{j=1}^{N}\Big[K_p(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(i)}, \boldsymbol{\Sigma}_{P,b(i)}) K_C(\mathbf{c}_{\mathbf{y}j} - \boldsymbol{\mu}_{C,b(i)}, \boldsymbol{\Sigma}_{C,b(i)}) \\ &\quad \times \delta(b_{\mathbf{y}_j} - b(i))\Big] \end{aligned} \tag{14}$$

The spatial-color model $p_{\mathbf{x}}(\mathbf{x}, \mathbf{c}_{\mathbf{x}}, b_{\mathbf{x}})$ might be sensitive to small spatial changes under the measure function. This problem was discussed by O'Conaire et al. [23] and Birchfield and Rangarajan [24]. However, this model also gives advantages of orientation estimation. As shown in Fig. 4, if there is no deformation between candidate and target, and the distance of motion is not excessively large between two adjacent frames, we can consider the motion of object of two frames as a pure translation. Under these assumptions, the center of target ($\mathbf{x}$) with respect to the mean of location of the b-th bin ($\boldsymbol{\mu}_{P,b(i)}$) in the model is in proportion to the center of candidate ($\mathbf{y}$) with respect to the mean of location of the b-th bin ($\boldsymbol{\mu}_{P,b(j)}$) in the candidate image. As a result, we can obtain,

$$\begin{aligned} &\boldsymbol{\mu}_{P,b(i)} - \mathbf{x} = \boldsymbol{\mu}_{P,b(j)} - \mathbf{y} \\ &\Rightarrow \boldsymbol{\mu}_{P,b(i)} = \boldsymbol{\mu}_{P,b(j)} - \mathbf{y} + \mathbf{x} \end{aligned} \tag{15}$$

Substituting (15) into (14), we can obtain the new similarity measure function as the following:

$$\begin{aligned} J(\mathbf{y}) = \frac{1}{NM}\sum_{i=1}^{M}\sum_{j=1}^{N}\Big[&K_P(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)} + \mathbf{y} - \mathbf{x}, \boldsymbol{\Sigma}_{P,b(i)}) \\ &\cdot K_C(\mathbf{c}_{\mathbf{y}j} - \boldsymbol{\mu}_{C,b(i)}, \boldsymbol{\Sigma}_{C,b(i)})\delta(b_{\mathbf{y}j} - b(i))\Big] \end{aligned} \tag{16}$$

The best candidate for matching can be found by computing the maximum value of the similarity measure. Let the gradient of the similarity function with respect to the vector $\mathbf{y}$ equals to $\mathbf{0}$, i.e., $\nabla J(\mathbf{y}) = \mathbf{0}$, then we can obtain the new position $\mathbf{y}_{\text{new}}$ of the target to be tracked,

$$\begin{aligned} &\nabla J(\mathbf{y}) = \mathbf{0} \\ &\Rightarrow \frac{1}{NM}\sum_{i=1}^{M}\sum_{j=1}^{N}(\boldsymbol{\Sigma}_{P,b(i)})^{-1}(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)} - \mathbf{y} + \mathbf{x}) K_P K_C \delta(b_{\mathbf{y}j} - b(i)) = \mathbf{0} \end{aligned} \tag{17}$$

By arranging this equation, the new position $\mathbf{y}_{\text{new}} = \mathbf{y}$ can be described as

$$\begin{aligned} \mathbf{y}_{\text{new}} = &\left\{\sum_{i=1}^{M}\sum_{j=1}^{N}(\boldsymbol{\Sigma}_{P,b(i)})^{-1} K_P K_C \delta(b_{\mathbf{y}_j} - b(i))\right\}^{-1} \\ &\times \left\{\sum_{i=1}^{M}\sum_{j=1}^{N}(\boldsymbol{\Sigma}_{P,b(i)})^{-1}(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)}) K_P K_C \delta(b_{\mathbf{y}j} - b(i))\right\} + \mathbf{x} \end{aligned} \tag{18}$$

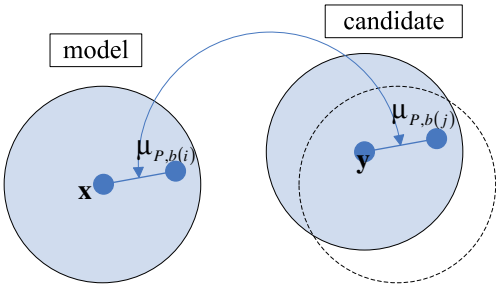

Fig. 4. Illustration of pure translation.

[1] For interpretation of color in Figs. 3–5 and 8–17 the reader is referred to the web version of this article.

where

$$K_P = K_P(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)} + \mathbf{y}_{old} - \mathbf{x}, \boldsymbol{\Sigma}_{P,b(i)}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)} + \mathbf{y}_{\text{old}} - \mathbf{x})^T(\boldsymbol{\Sigma}_{P,b(i)})^{-1}(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)} + \mathbf{y}_{\text{old}} - \mathbf{x})\right)}{2\pi|\boldsymbol{\Sigma}_{P,b(i)}|^{1/2}}$$

and

$$K_C = K_C(\mathbf{c}_{\mathbf{y}j} - \boldsymbol{\mu}_{C,b(i)}, \boldsymbol{\Sigma}_{C,b(i)}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{c}_{\mathbf{y}j} - \boldsymbol{\mu}_{C,b(i)})^T(\boldsymbol{\Sigma}_{C,b(i)})^{-1}(\mathbf{c}_{\mathbf{y}j} - \boldsymbol{\mu}_{C,b(i)})\right)}{(2\pi)^{3/2}|\boldsymbol{\Sigma}_{C,b(i)}|^{1/2}}$$

In the sequel, we define $\mathbf{y}_{\text{old}}$ as the current position.

3. Multi-channel microphone interface

In order to obtain multi-channel audio data for speech enhancement and sound source tracking algorithms, a multi-channel microphone interface using FPGA is implemented. Digital microphones were used due to their minimal interference with the digital circuit. A digital microphone is a device that both an amplifier and a sigma-delta modulator [25] are embedded in the microphone. It outputs a 1-bit data stream insensitive to noise and also has the small size advantage. Therefore, digital decimation filters have to be implemented to obtain the quantized microphone signals in a desired bandwidth.

To achieve the decimation process and data transmission between digital microphones and Texas Instruments DM6446 platform, we use an Altera Cyclone II FPGA. In the FPGA design, the decimation filter process consists of an infinite impulse response (IIR) filter and downsample process. For each channel, data from digital microphones is a 1-bit stream sampled at 12 MHz clock. The 1-bit stream data is transferred into 16-bit parallel data with a reduction of sampling rate 16 kHz. The IIR filter is implemented by a moving-average low-pass filter at a cut-frequency of 8 kHz, and the pole of this filter is shifted to maintain stability. The same architecture is also used for the high-pass filter at cutoff frequency about 80 Hz to remove DC components.

The audio data is transmitted from FPGA to Davinci through external memory interface. On the FPGA side, we implement the transceiver by a ping-pong structure. Audio data flow is described below and the functional block diagram of FPGA is showed in Fig. 5:

(1) 1-bit data stream is sampled from each microphone, and then the decimation filter transfers it into 16-bit data stream.
(2) When the 16 kHz clock is rising, we write the eight-channel data into corresponding ping (or pong) buffer; at the meantime, the pong (or ping) buffer can be read by Davinci.
(3) As the buffer is full, the *ping-pong controller* exchanges the R/W enable for the ping and pong buffer, and raise the *full* signal until Davinci starts to receive data.
(4) When Davinci starts to access data, it would first check the *full* signal which corresponds to a specified address. If the *full* signal is active, then Davinci starts to send read address (*rdaddr*) and receive data while pulling down the *full* signal.

4. Dual-core platform programming

The embedded dual-core platform, DM6446 EVM (Davinci), consists of the ARM subsystem and DSP subsystem. The ARM subsystem mainly controls the system operation and I/O access, and the DSP subsystem computes various algorithms. Communication between ARM and DSP is performed using DSP/BIOS Link™ (an inter-processor communications scheme offered by Texas Instruments).

4.1. Receiving audio data from FPGA

In Davinci, the audio data of eight-channel microphones is received through the asynchronous external memory interface (AEMIF) [26]. The Davinci platform memory map is shown in Fig. 6. In Fig. 6, we can see that the AEMIF supports four addressable spaces (from CS2 to CS5). Each space provides up to 32 MB, and there are two choices for the data bus width, 8-bit and 16-bit. To match the sound data from FPGA, the data bus width is set to 16-bit. We choose the space CS3 (0×04000000–0×06000000) due to its capability for accepting plug-in daughter cards, such as memory.

In practice, we accelerate the data transmission time by tuning the time in *Setup*, *Strobe*, and *Hold* states in the signal transmission procedure. It efficiently reduces the CPU loading.

4.2. DSP subsystem programming

The DSP subsystem mainly computes the audio algorithms, including VAD, DOA estimation, and speech enhancement. The complete software flowchart in DSP subsystem is described in Fig. 7. When the ping (or pong) buffer is full in FPGA, DSP will obtain eight-channel sound data from FPGA by accessing AEMIF. After the data transmission of a buffer is complete, a 10th-order beamformer to enhance the speech signal is activated to reduce the background noise, and the eight-channel data is integrated into one channel (enhanced signal). The enhanced sound data is passed into VAD system to check whether there is speech or not. The sound data and VAD detection results will be sent to the ARM subsystem by DSP/BIOS Link.

Two different program stages are separated by the VAD result. If the VAD system determines that there is no speech (only background noise) in the environment, the beamformer parameters

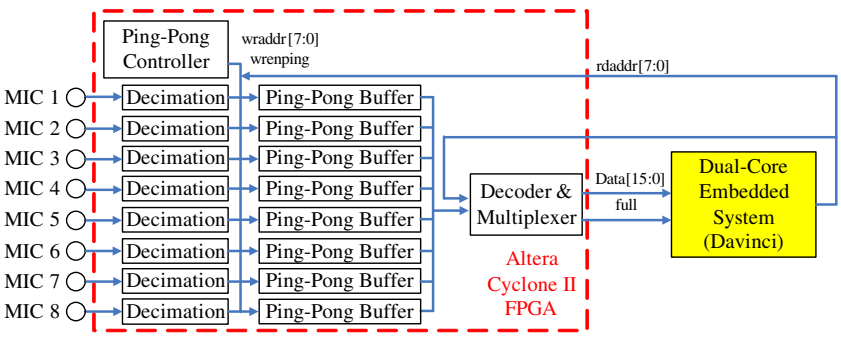

Fig. 5. FPGA functional block diagram.

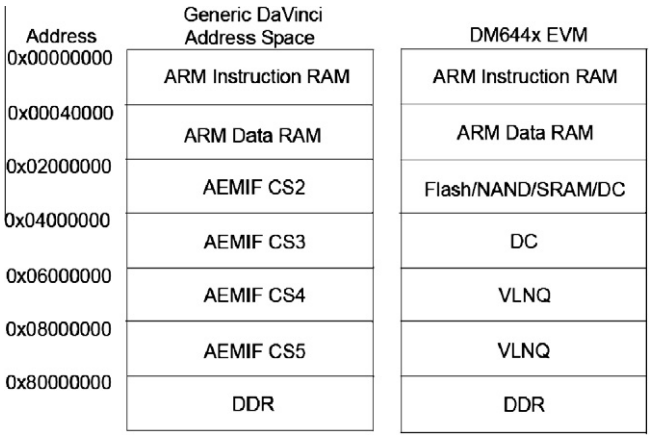

Fig. 6. Memory map of Davinci.

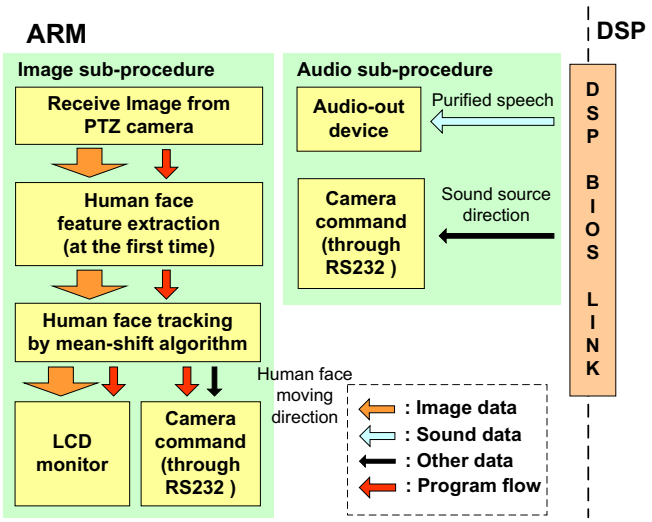

Fig. 8. Software flowchart of ARM subsystem.

update system will be activated, and the sound data will be used to train and update the parameters of the beamformer. Therefore, the beamformer has the ability to adapt to the background noise in different environments. The beamformer is updated by a designated reference speech signal from the data base and the input background noise with normalize least-mean-square (NLMS) algorithm, and the reference speech is chosen by the last estimation of sound source direction, which is provided by the sound source tracking system.

If the VAD block determines that there is speech, the sound source tracking system will be activated to estimate the direction of speaker. The estimated results provide not only for the beamformer to update its parameters, but also for the camera control at the ARM side sent by DSP/BIOS Link. After completing the whole program flow, the DSP subsystem will wait until the next buffer is filled.

4.3. ARM subsystem programming

The software flowchart for the ARM subsystem is showed in Fig. 8 which is divided into image and audio parts. For the audio part, the enhanced speech data received from DSP is output to audio device. Besides, the sound source direction information is used to control a PTZ camera through RS232 to turn the camera toward the speaker. To execute these two functions concurrently, two threads are built to handle the procedures.

Due to the computing burden of audio algorithms in the DSP subsystem, the human face tracking algorithm is implemented on the ARM subsystem. When the face is entering the frame at the first time, the human face features are extracted to build the target model. The target model won't be updated unless the face is out of the camera range. The mean-shift algorithm finds the displacement direction of human face, and then sends the control signal to PTZ camera to keep the face in the center of the tracking frame. The tracking results are displayed on the LCD monitor. For concurrency, a thread is created to control the image input and output, and two threads for human face tracking and camera control, respectively.

The execution time of the algorithms is showed in Table 1. Note that human face tracking is executed in the ARM side, and other audio algorithms are measured in the DSP side. The DSP side,

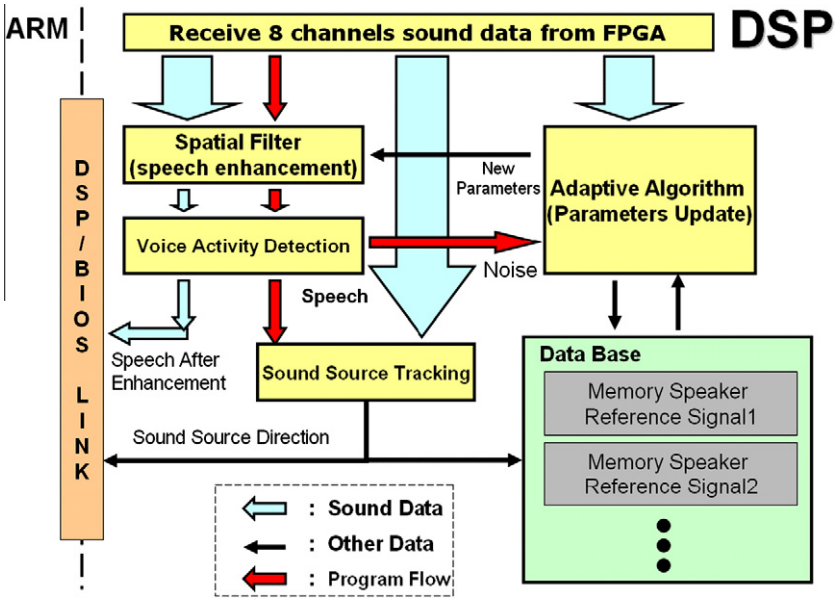

Fig. 7. Software flowchart of DSP subsystem.

Table 1
Execution time of the algorithms.

Algorithm	Execution time (ms)
Voice activity detection	0.041
Sound source tracking	0.247
Speech enhancement (beamformer)	0.275
Speech enhancement (adaptive algorithm)	3.293
Human face tracking	46.768

Table 2
Audio data transmission format.

Sampling frequency	16 kHz
Data size	16-bits
Type of transform	Short-time Fourier transform
STFT frame length	256
Frame period	16 ms
Overlap length	128
Windowing	Hamming window

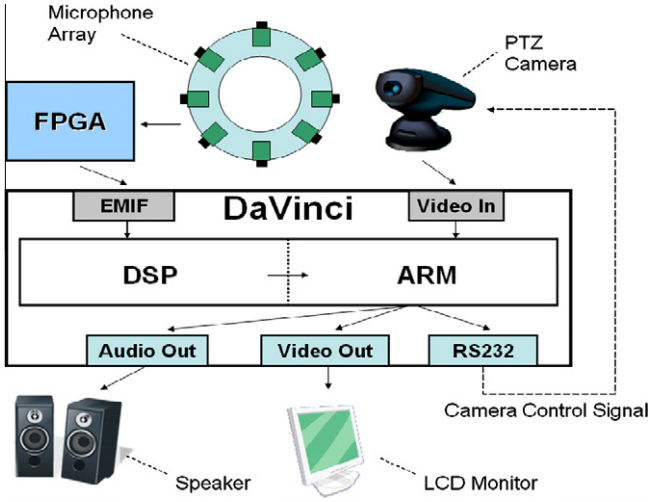

Fig. 9. System architecture.

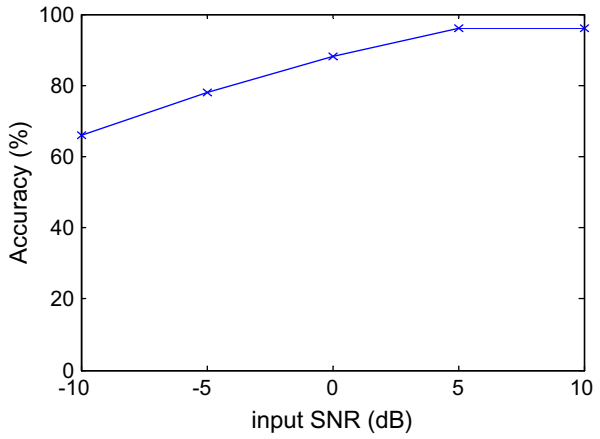

Fig. 11. VAD testing results.

C64xx DSP, performs at 600 MHz clock, and it can execute 8 instructions in a cycle time. The ARM side performs at 300 MHz clock otherwise.

5. Experimental results

The system was verified through experiments over a long period of time in a conference room with the size of 10 m × 6 m × 3.6 m. The experiments of VAD tests, speech enhancement performance evaluations, and sound source tracking tests were carried out using various speechsegments of different time periods. The human face tracking system was also evaluated by different objects.

The overall system architecture is illustrated in Fig. 9 with the picture of the system in Fig. 10. A circular array with eight digital microphones and 7 cm radius was constructed to acquire the sound data at 16 kHz sampling rate. Table 2 shows the corresponding data format for audio transmission. A demo video is available on the website for your reference [27].

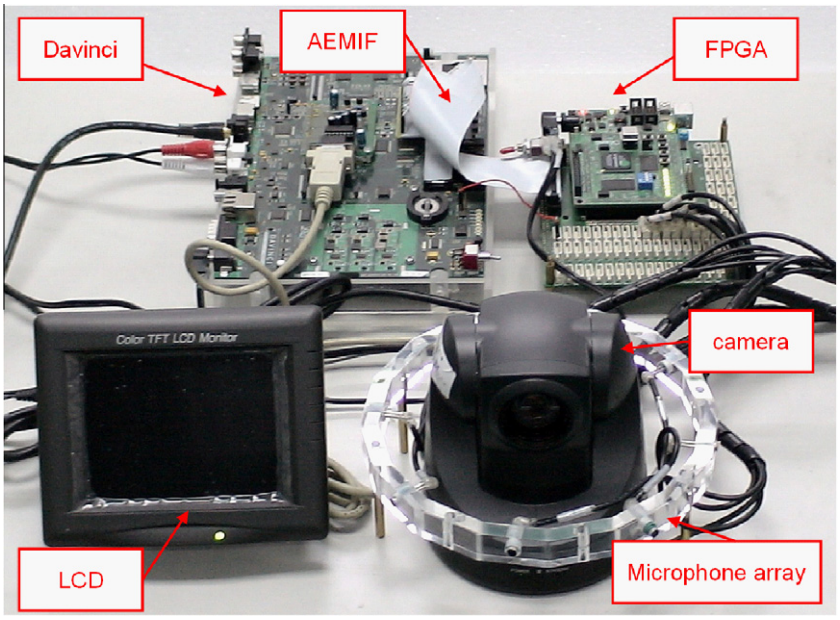

Fig. 10. The picture of the embedded system.

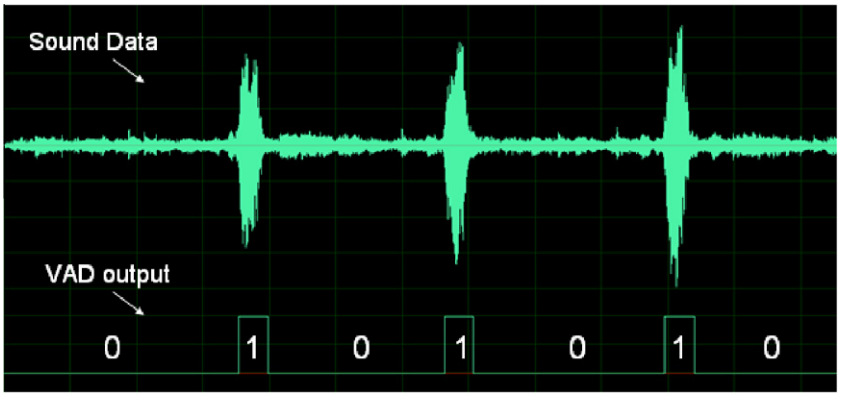

Fig. 12. A snapshot of VAD results.

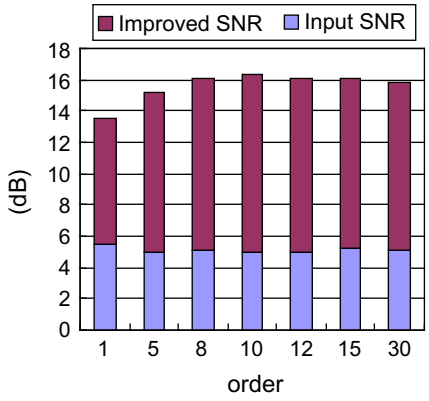

Fig. 13. The input SNR and improved SNR of different filter tap lengths.

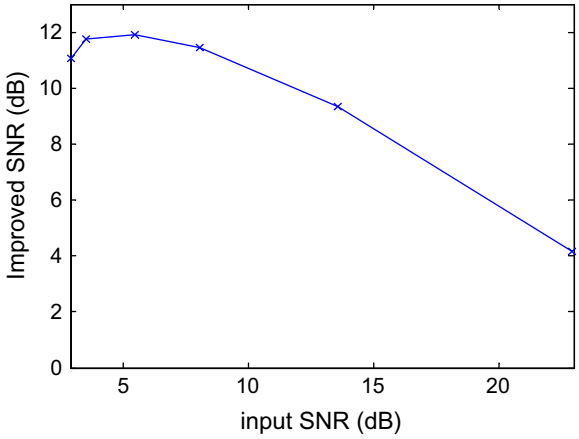

Fig. 14. Speech enhancement results for different SNR.

5.1. Voice activity detection test

The VAD algorithm was tested in a noisy environment, where two loudspeakers placed 40 cm apart playing music with a male singer as background noise. And the testing person uttered at a 40 cm distance from the microphone array with an angle of 90° to the line connecting the loudspeakers. Each testing data length was 2 s, and each testing data were tested 50 times at different SNR. The parameters of VAD algorithm are set as $\gamma_0 = 160$, $\gamma_1 = 5$, $E_0 = 100$, $E_1 = 220$. They are empirical values for the algorithm adaption. The enhanced sound data from speech enhancement system was sent directly to the VAD system for detecting there is speech or not.

Fig. 11 shows the VAD results under different input SNR. As it can be seen, the proposed VAD algorithm still has about 80% accuracy under low input SNR. This is because the VAD system is arranged after the speech enhancement system. With this

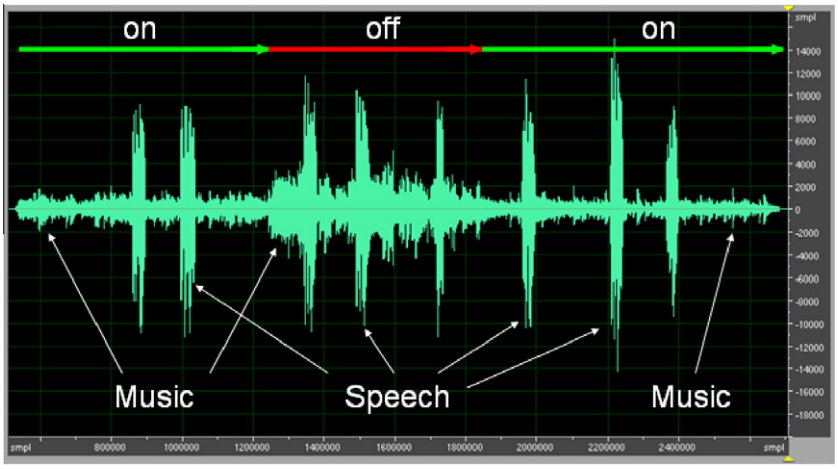

Fig. 15. The waveform of speech enhancement in real-time.

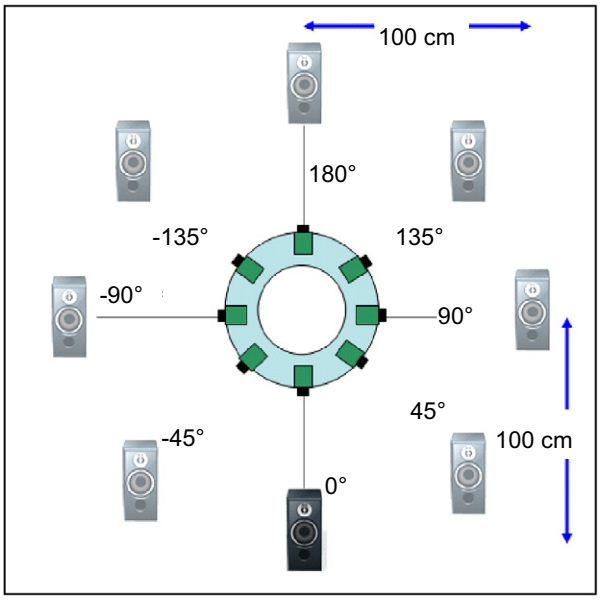

Fig. 16. Experiment environment for sound source tracking.

Table 3
Results of sound source direction estimations.

Actual direction (°)	Average (°)	Variance (°)
−135	−136.16	2.85
−90	−88.24	5.46
−45	−43.72	2.44
0	0.32	0.78
45	43.62	2.39
90	91.92	4.15
135	137.7	3.33
180	179.34	0.86

arrangement, the VAD system becomes more reliable and provides accurate information for speech enhancement weight update and sound source tracking system.

Fig. 12 shows a snapshot of the first microphone record and the real-time VAD results. The VAD output "1"stands for speech presence, and "0" stands for speech absence.

5.2. Speech enhancement experiment

To evaluate the performance of speech enhancement system, we first found the best tap length of the spatial filter, and then we compared the improvement SNR under different input SNR cases. Assume that the M_1-th to the N_1-th input data are only background noise, and the M_2-th to the N_2-th input data are mixed with noise and speech. Then the SNR is computed by

$$10\log\left(\frac{\sum_{i=M_2}^{N_2}x^2(i)}{N_2-M_2+1}\right)-10\log\left(\frac{\sum_{i=M_1}^{N_1}x^2(i)}{N_1-M_1+1}\right) \tag{19}$$

To find the optimal order of the filter, different tap lengths with 5 dB input SNR were tested. Fig. 13 shows that the filter with 10 tap lengths has the best SNR improvement (about 11.286 dB).

The SNR improvement of the speech signal is shown in Fig. 14. It is clear that except for the case when input SNR is 22.946 dB, which approaches the SNR limit of clean speech, the SNR improvement is about 11 dB in average.

A signal waveform of the speech enhancement result is shown in Fig. 15. The background noise is music with a male singer, and other experiment conditions are the same as VAD testing. The periods with symbol 'on' means the enhancement function is executed and vice versa.

5.3. Sound source tracking test

The experiment environment for sound source tracking is described in Fig. 16. The pre-record speech is played with different input angles compared to the reference microphone (MIC 1). The loud speaker was placed apart from the microphone array at a distance of 100 cm. The algorithm is tested 50 times at each direction, and the estimation results are described in Table 3.

The experiment results prove that the error of the averaged angle is less than 3° and the sound source tracking system is suitable

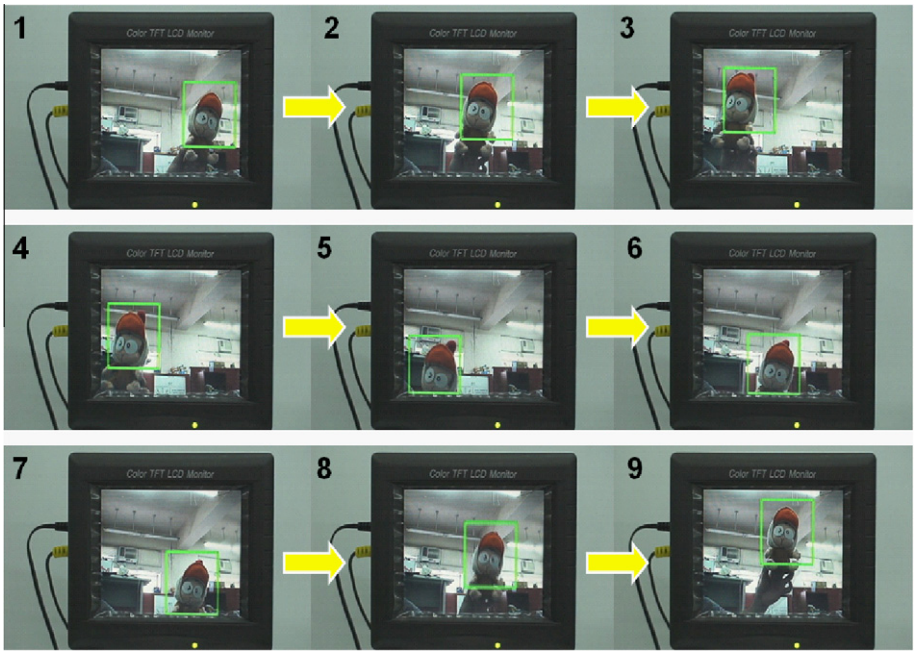

Fig. 17. Human face tracking results with a doll.

for integrating with other applications (e.g. face tracking system). Due to the unmatched characteristics of the microphones, the variance in Table 3 seems to be symmetric. The estimation error may also be caused by quantization errors in fixed-point computation, imperfect point sound source characteristic of the speaker, and different sound speeds affected by environment temperatures, etc. But the estimation error is small enough for identifying the speaker's angle relative to the system.

5.4. Human face tracking test

The PTZ camera was controlled by the sound source direction estimations stated in Section 2.3 and human face tracking results stated in Section 2.4. The camera can turn ±170° horizontally and turn ±120° vertically. In our case, we only control the horizontal directions.

The images were processed by the spatial-color mean-shift algorithm [21] to achieve the human face tracking. Since the algorithm's main idea is to match the similarity of the color distributions between the target region and the model, we can verify this function by several objects, not necessarily utilizing human faces.

Fig 17 shows the video sequence of the tracking result. A frame around the doll was drawn in real-time to show the tracking accuracy.

6. Conclusion

In this paper, we implemented a high integrated audio–visual tracking system based on VAD, speaker direction estimation, speech enhancement, mean-shift, and Texas Instruments DM6446 EVM. An FPGA based eight-channel digital microphone array data acquisition system is also implemented. The estimation results of the sound source direction tracking system is reliable with mean error than 3°, and the results prove that it's helpful when the image is out of the frame. The proposed voice activity detection system can detect the voice activity accurately, and is combined with speech enhancement system and sound source tracking system tightly, which helps to reduce the total computing burden. Besides, the speech enhancement improves the signal-to noise ratio (SNR) about 11 dB in average.

As described before, for applications such as robotics and vehicles, embedded implementation is necessary to meet the cost and size constraints. However, the implementation of a real-time audio–visual interface can be quite different depending on the features and algorithms. Usually, it involves several sub-systems and not much work was reported to integrate the whole audio–visual tracking functions on an embedded system. Compare with prior works, we successfully integrated functionalities for real-time audio–visual tracking on an embedded system with limited computation power and achieved a complete human–machine interface on robots or other applications.

References

[1] K. Nakadai, T. Lourens, H.G. Okuno, H. Kitano, Active audition for humanoid, in: Proceedings National Conference on Artificial Intelligence, 2000, pp. 832–839.
[2] J. Hu, C.C. Cheng, W.H. Liu, Robust speaker's location detection in a vehicle environment using GMM models, IEEE Transactions on System, Man and Cybernetics, Part B 36 (2) (2006) 403–412.
[3] S. Yamamoto, K. Nakadai, H. Tsujino, T. Yokoyama, H. Okuno, Improvement of robot audition by interfacing sound source separation and automatic speech recognition with missing feature theory, in: Proceedings IEEE International Conference on Robotics and Automation, 2004, pp. 1517–1523.
[4] H.G. Okuno, K. Nakadai, T. Lourens, H. Kitano, Sound and visual tracking for humanoid robot, Springer Applied Intelligence 20 (2004) 253–266.
[5] K. Hyun-Don, K. Komatani, T. Ogata, H.G. Okuno, Auditory and visual integration based localization and tracking of humans in daily-life environments, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 2021–2027.
[6] J.G. Trafton, N.L. Cassimatis, M.D. Bugajska, et al., Enabling effective human–robot interaction using perspective-taking in robots, IEEE Transactions on Systems, Man and Cybernetics, Part A 35 (4) (2005) 460–470.
[7] T. Suwannathat, J.i. Imai, M. Kaneko, Omni-directional audio–visual speaker detection for mobile robot, in: IEEE International Symposium on Robot and Human interactive Communication, 2007, pp. 141–144.
[8] A. Wang, K. Yao, R.E. Hudson, D. Korompis, F. Lorenzelli, S.D. Soli, S. Gao, A novel DSP system for microphone array applications, in: IEEE International Symposium on Circuits and Systems, vol. 2, 1996, pp. 201–204.
[9] K. Nickel, T. Gehrig, R. Stiefelhagen, J. McDonough, A joint particle filter for audio–visual speaker tracking, in: Proceedings of the 7th International Conference on Multimodal Interfaces, 2005, pp. 61–68.
[10] J.H. Connell, N. Haas, E. Marcheret, C. Neti, G. Potamianos, S. Velipasalar, A real-time prototype for small-vocabulary audio–visual ASR, in: International Conference on Multimedia and Expo, 2003, pp. 469–472.
[11] T. Gehrig, K. Nickel, H.K. Ekenel, U. Klee, J. McDonough, Kalman filters for audio–video source localization, in: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005, pp. 118–121.
[12] L. Yoon Seob, C. Jong-Suk, K. Munsang, Particle filter algorithm for single speaker tracking with audio–video data fusion, in: IEEE International Symposium on Robot and Human Interactive Communication, 2007, pp. 363–367.
[13] D. Gatica-Perez, G. Lathoud, J.M. Odobez, I. McCowan, Audiovisual probabilistic tracking of multiple speakers in meetings, IEEE Transactions on Audio, Speech and Language Processing 15 (2) (2007) 601–616.
[14] R. Brunelli, A. Brutti, P. Chippendale, O. Lanz, M. Omologo, P. Svaizer, F. Tobia, A generative approach to audio–visual person tracking, Multimodal Technologies for Perception of Humans, Lecture Notes in Computer Science, vol. 4122, Springer Berlin/Heidelberg, 2007.
[15] K. Bernardin, R. Stiefelhagen, A. Waibel, Probabilistic integration of sparse audio–visual cues for identity tracking, in: Proceeding of the 16th ACM international conference on Multimedia, 2008, pp. 151–158.
[16] TI, DaVinciEVM_TechRef, Spectrum Digital Inc., October 2006.
[17] J. Ramírez, J.C. Segura, C. Benítez, D.l. Torre, Ángel, A. Rubio, Efficient voice activity detection algorithms using long-term speech information, Speech Communication 42 (2004) 271–287.
[18] J.S. Hu, C.C. Cheng, W.H. Liu, Processing of speech signals using microphone array for intelligent robots, Proc. Instn Mech. Engrs, Part I. J. Systems and Control Engineering 219 (I2) (2005) 133–144.
[19] J.S. Hu, C.C. Cheng, Frequency domain microphone array calibration beamforming for automatic speech recognition, IEICE Trans. Fundamentals E88-A (9) (2005) 2401–2411.
[20] M. Wax, T. Kailath, Detection of signals by information theoretic criteria, in: IEEE Transactions on Acoust. Speech Signal Processing, vol. ASSP-33, 1985, pp. 387–392.
[21] C. Yang, R. Duraiswami, L. Davis, Efficient mean-shift tracking via a new similarity measure, IEEE Conference on Computer Vision and Pattern Recognition 1 (2005) 176–183.
[22] S. Birchfield, S. Rangarajan, Spatiograms versus histograms for region-based tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2005, pp. 1158–1163.
[23] C. O'Conaire, N.E. O'Connor, A.F. Smeaton, An improved spatiogram similarity measure for robust object localization, in: IEEE Conference on Acoust. Speech Signal Processing, 2007.
[24] S. Birchfield, S. Rangarajan, Spatial histograms for region-based tracking, ETRI Journal 29 (5) (2007) 697–699.
[25] R. Schreier, G. Temes, Understanding Delta-Sigma Data Converters, IEEE Press John Wiley & Sons Inc., 2005.
[26] TI, TMS320DM644xDMSoC, Asynchronous External Memory Interface (EMIF) User's Guide, Literature Number: SPRUE20A, June 2006.
[27] Website: <http://140.113.150.64/Davinci_demo.wmv>.

Jwu-Sheng Hu received the B.S. degree from the Department of Mechanical Engineering, National Taiwan University, Taiwan, in 1984, and the M.S. and Ph.D. degrees from the Department of Mechanical Engineering, University of California at Berkeley, in 1988 and 1990, respectively. He is currently a Professor in the Department of Electrical and Control Engineering, National Chiao-Tung University, Taiwan, ROC. His current research interests include microphone array signal processing, active noise control, intelligent mobile robots, embedded systems and applications.

Ming-Tang Lee was born in Taoyuan, Taiwan, in 1984. He received the B.S. degree and the M.S. degree in Electrical and Control Engineering from National Chiao Tung University, Taiwan in 2007 and 2008 respectively. He is currently a Ph.D. candidate in Department of Electrical and Control Engineering at National Chiao Tung University, Taiwan. His research interests include sound source localization, speech enhancement, microphone array signal processing, and adaptive signal processing.

Chia-Hsing Yang was born in Taipei, Taiwan, in 1981. He received the B.S. degree and the M.S. degree in Electrical and Control Engineering from National Chiao Tung University, Taiwan in 2003 and 2005 respectively. He is currently a Ph.D. candidate in Department of Electrical and Control Engineering at National Chiao Tung University, Taiwan. His research interests include sound source localization, speech enhancement, microphone array signal processing, and adaptive signal processing.