A Voice-Driven Scene-Mode Recommendation Service for Portable Digital Imaging Devices

Yoo Rhee Oh, *Student Member*, *IEEE*, Jae Sam Yoon, *Student Member*, *IEEE*, Hong Kook Kim, *Senior Member*, *IEEE*, Myung Bo Kim, and Sang Ryong Kim

Abstract — *In this paper, we propose a voice-driven scene-mode recommendation service in order to more easily select scene-modes on portable digital imaging devices such as digital cameras and camcorders. In other words, the proposed service is designed to recommend or automatically change the scene-mode by recognizing a user's voice command regarding scene or scene-related words. To realize such a service, we implement a system which is mainly composed of voice activity detection, automatic speech recognition (ASR), utterance verification, and word-to-scene-mode mapping. However, several optimization methods should be applied since portable digital imaging devices operate on embedded systems with limited resources. In addition, a speech adaptation database for acoustic models is developed such that the ASR system can adjust to the characteristics of the microphones and operating environments. Finally, the performance of the voice-driven scene-mode recommendation system is measured in terms of processing time and scene-mode recognition accuracy (SMRA). It is shown from the experiments that the average processing time and the average SMRA are around 500 ms and 98.0% for 50 scene-related words, respectively, and 1200 ms and 96.8% for 200 scene-related words.*[1]

Index Terms — Portable digital imaging devices, scene-mode recommendation, voice interface, embedded speech recognition.

I. INTRODUCTION

A wide range of consumer electronics has been developed that have the capability to capture images or video such as digital imaging devices, communications devices, and home appliances. Most commonly, these devices typically include digital cameras, cellular phones, and camcorders. In order to assist a user in easily taking high-quality images, many digital image-capturing devices support various scene-modes that are accessible by a means of buttons, dials, or menus. This type of scene-mode is a helpful and handy function that automatically changes the exposure, aperture, focus, and other related settings for a given scene [1]; thus, even an amateur can take high-quality images or videos. For example, when 'night scene' is selected, several settings are automatically changed to enable the user to take a nice image at night. However, not all users are familiar with which scene mode is appropriate for the image to be captured. In order to overcome this problem, a scene-mode recommendation service could adopt a voice-driven interface [2]. Indeed, such a voice-driven scene-mode recommendation has several benefits. First, it reduces the number of actions required for a user to activate a scene-mode. For example, in order to select the desired scene-mode, at least four steps are typically required in a menu-based scene-mode service, including main menu selection, scene-mode menu selection, movement to a situation-specific scene-mode, and scene-mode selection. Second, the scene-mode recommendation service could be naturally distinguished by a built-in push-and-talk button designed to select the desired scene-mode. Third, a broad concept of the scene-mode recommendation service could be adopted by recognizing scene-related words and mapping them into an appropriate scene-mode. In this case, a user could just utter a word such as 'spaghetti' and the voice-driven scene-mode recommendation service would then change several settings for the scene-mode 'food' by mapping the recognized scene-related word 'spaghetti' onto the scene-mode 'food'.

In this paper, we propose a voice-driven scene-mode recommendation service on a portable digital imaging device in order to provide better scene-mode selection to device users. To this end, we first design the overall scenario for the proposed voice-driven scene-mode recommendation service. That is, the service is activated when a user presses a push-and-talk button and utters a scene or scene-related word. After recognizing the word, the word is mapped into an appropriate scene-mode, and this scene-mode is then shown on the device along with related settings such as exposure, aperture, and focus. Next, based on this scenario, we construct a system for the proposed voice-driven scene-mode recommendation service. The system is composed of seven main parts: voice input acquisition, speech recognition feature extraction, voice activity detection, automatic speech recognition, utterance verification, word-to-scene-mode mapping, and scene-mode activation.

We then need to optimize the performance of the recommendation system since it is likely equipped with limited resources such as a low performance central processing unit (CPU) and a small memory size. This optimization includes fixed-point arithmetic programming and its optimization, memory optimization of acoustic models for speech recognition, and the mathematical approximation of

[1] This work was supported by Samsung Electronics Co., Ltd. and the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2007-314-D00245).

Yoo Rhee Oh, Jae Sam Yoon, and Hong Kook Kim are with the Department of Information and Communications, Gwangju Institute of Science and Technology (GIST), 1 Oryong-dong, Buk-gu, Gwangju 500-712, Korea (e-mail: {yroh, jsyoon, hongkook}@gist.ac.kr).

Myung Bo Kim and Sang Ryong Kim are with Camcorder Business Team, Digital Media Business, Samsung Electronics, 416 Meatan 3-dong, Yeongtong-gu, Suwon-si, Gyeonggi-do 443-742, Korea (e-mail: {kmbo.kim, srkim}@samsung.com).

Contributed Paper
Manuscript received August 25, 2009

logarithmic functions. In addition, we develop a speech adaptation database in order to adjust acoustic models to the characteristic of microphones and operating environments. In particular, we apply a posteriori/maximum likelihood linear regression (MAP/MLLR) adaptation for the acoustic model adaptation [3].

The remainder of this paper is structured as follows. Following this introduction, Section II presents the scenario for the proposed voice-driven scene-mode recommendation service for portable digital imaging devices. Next, Section III describes how to construct the proposed voice-driven scene-mode recommendation system, focusing on automatic speech recognition (ASR). Section IV discusses implementation issues of the proposed system on a portable digital imaging device that has limited resources. Section V then demonstrates the performance of the proposed voice-driven scene-mode recommendation system, and we finally conclude this paper in Section VI.

II. SCENARIO OF THE PROPOSED VOICE-DRIVEN SCENE-MODE RECOMMENDATION SERVICE FOR PORTABLE DIGITAL IMAGING DEVICES

A voice-driven scene-mode recommendation service for portable digital imaging devices implies that a scene-mode and its corresponding settings are automatically changed by recognizing user's voice. Fig. 1 shows the overall scenario of the proposed voice-driven scene-mode recommendation service applied to a portable digital imaging device. In the figure, the proposed service is activated when a user presses a designated push-and-talk (PAT) button. After being initialized, the other buttons are locked and a pressed button-shaped icon is displayed to notify the user of the time when a scene-related word should be uttered. It is noted here that the PAT button can be released by the user at any time. If voice activity detection (VAD), which will be described in Section III-A, successfully detects the end of the voice input, then all buttons are unlocked even if the user is still pressing the PAT button. Thus, from now on, the PAT button is referred to as a soft PAT button. That is, by having a soft PAT button, the voice-driven scene-mode recommendation service is activated when the button pressed but deactivated when the button is released or the end of the voice input is detected. Next, the user's voice is recognized and then the scene mode of the image-capturing device is changed into the desired scene-mode along with related settings according to the recognition result. However, if the recognition fails, the scene-mode that is currently in use is maintained.

III. PROPOSED VOICE-DRIVEN SCENE-MODE RECOMMENDATION SYSTEM

In this section, we explain the proposed voice-driven scene-mode recognition system that is implemented based on the scenario described in Section II. Fig. 2 shows the overall procedure of the system. As shown in the figure, the proposed system consists of two main threads: a '*voice input thread*'

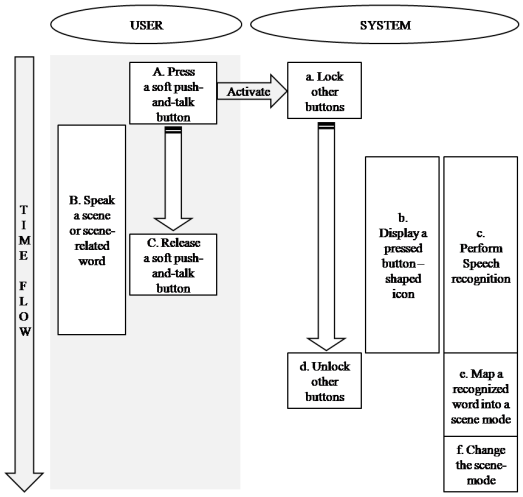

Fig. 1. Scenario of the voice-driven scene-mode recommendation service.

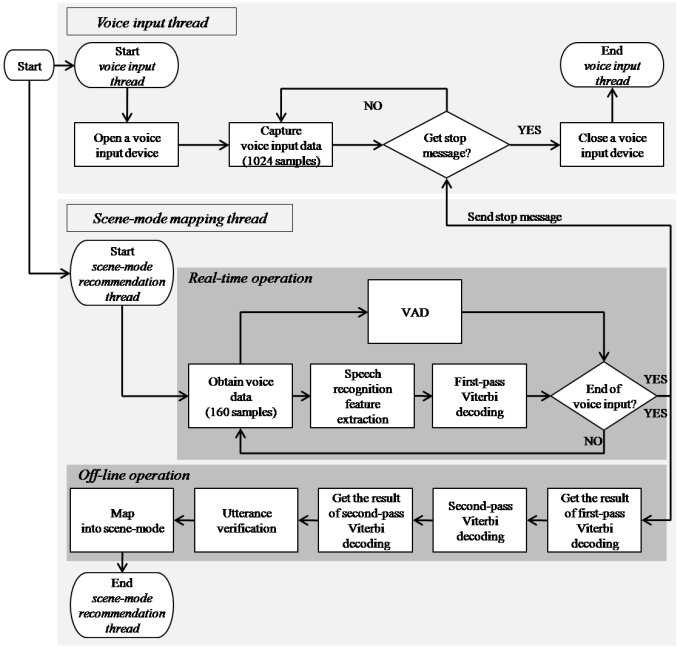

Fig. 2. Overall procedure of the voice-based scene-mode recommendation system implemented on a portable digital imaging device.

and a '*scene-mode mapping thread*'. The two threads are initiated when the voice-driven scene-mode recommendation system is activated by pressing the soft PAT button. The threads then operate as follows.

The '*voice input thread*' controls the flow of voice input from a microphone. When the '*voice input thread*' starts by pressing the soft PAT button, the voice input device is opened at a sampling rate of 16 kHz with a resolution of 16 bits. After that, the '*voice input thread*' obtains 1024 voice input samples which correspond to a length of 64 ms, and then moves the samples into a voice input buffer. Such voice data acquisition is carried out until the '*scene-mode mapping thread*' sends a stop message to the '*voice input thread*'.

On the other hand, the '*scene-mode mapping thread*' controls the flow of converting voice input data into a proper scene-mode based on the result of ASR. When the soft PAT

button is pressed and the '*scene-mode mapping thread*' starts, two different operations, real-time operation and off-line operation, are performed after initializing several variables.

The real-time operation of the '*scene-mode mapping thread*' is activated once every frame until the end of voice input is detected. First, a frame of the voice input samples is popped up from the voice input buffer, where we define a voice input frame as 10 ms, i.e., there are 160 samples. After that, speech recognition feature extraction and VAD are sequentially performed to recognize the user's voice and to deactivate the soft PAT button by detecting the end of voice input.

Next, a first-pass Viterbi decoding is performed for real-time ASR, and a decision based on the result of the Viterbi decoding is then made as to whether or not the soft PAT button is deactivated. If it is determined that the soft PAT button is still activated, the real-time operation is repeated. Otherwise, the '*scene-mode mapping thread*' posts a stop message to run the off-line operation.

The main function of the off-line operation of the '*scene-mode mapping thread*' is a second-pass Viterbi decoding. In other words, a second-pass Viterbi decoding is performed for all voice input data using the restricted search space from the result of the first-pass Viterbi decoding. After all, we have a recognized word after verifying whether or not the Viterbi score of the recognized word is highly confident. Finally, the appropriate scene-mode is estimated using a word-to-scene-mode mapping table, which then is changed from the previous scene-mode.

The following subsections provide a more detailed description of the soft PAT button and VAD, speech recognition feature extraction, Viterbi decoding, utterance verification, and word-to-scene-mode mapping.

A. Soft push-and-talk button and voice activity detection

As mentioned in Section II, the soft PAT button is activated when the button is pressed by a user. On the other hand, deactivation of the soft PAT button is determined by combining the results of VAD and the status of button. In other words, we first examine the number of frames inputted after the soft PAT button is activated. If the number is larger than a predefined threshold, $thr_{time\text{-}over}$, we decide that it is a time-over and thus deactivate the button. Except for the time-over, VAD tries to detect the end of the voice input. If N frames have passed after VAD detects the end of the last voice input, we decide that the voice input has been ended and deactivate the button. Otherwise, we decide to fail to detect the end of voice input with another time over and thus deactivate the button. In this paper, we set $thr_{time\text{-}over}$ =300 and N=100.

In order to detect the end of voice input, we use a Teager energy-based VAD algorithm [4] that is known to have several benefits. First, the Teager energy has better performance to detect word boundaries for fricatives and plosives than a conventional energy [5]. Second, the computational complexity of the Teager energy-based VAD algorithm is lower than that of the conventional energy-based VAD [6][7]; for more detailed description on the Teager energy-based VAD algorithm, refer to [4].

B. Speech recognition feature extraction

As a speech recognition feature, we extract 12 mel-frequency cepstral coefficients (MFCCs) with a logarithmic energy, and concatenate their first and second derivatives, resulting in a 39-dimensional feature vector for each frame. It is known that ASR performance can be improved if logarithmic energy is normalized by the following way [8]. That is, we first compute the maximum logarithmic energy $LogE_{\max}$ over the entire frames of the input voice. In addition, in order to maintain the dynamic range of the normalized logarithmic energy within a certain range, we bound the normalized energy with a predefined threshold, thr_{LogE}, by using the equation of

$$LogE_{\min} = LogE_{\max} - thr_{LogE} \tag{1}$$

where $LogE_{\min}$ is the minimum logarithmic energy and thr_{LogE}=50 in this work. Then, the logarithmic energy of the n-th frame, $LogE(n)$, is normalized by using $LogE_{\max}$ and $LogE_{\min}$ such that

$$LogE'(n) = \begin{cases} LogE_{\min}, & if \;\; LogE(n) < LogE_{\min} \\ LogE_{\max} - LogE(n), & otherwise \end{cases} \tag{2}$$

where $LogE'(n)$ is the normalized logarithmic energy of the n-th frame. The normalized feature vector is finally constructed by replacing the normalized energy and its first and second derivatives with the unnormalized energy and its corresponding derivatives.

C. Viterbi decoding

As shown in Fig. 2, the proposed voice-driven scene-mode recommendation system adopts a two-pass Viterbi decoding algorithm [9][10] that provides both a fast and accurate recognition process. In other words, the first-pass Viterbi decoding is performed whenever voice input data enter into the system, and then it provides an N-best list. Next, the second-pass Viterbi decoding is performed within the search space restricted by the N-best list obtained from the first-pass Viterbi decoding. Since the second-pass Viterbi decoding uses a restricted search space, it takes around 500 ms for the task of 50-word recognition, which to the users is considered as a real-time response. Thus, such a two-pass Viterbi decoding algorithm is deemed suitable for portable consumer electronics that typically have limited CPU performance.

As mentioned in the previous subsection, we have two different feature vectors for each frame; an original (or unnormalized) feature vector and a normalized feature vector. For the first-pass Viterbi decoding, unnormalized feature vectors are used because energy normalization should be performed after collecting the energy parameters of all the speech frames. Thus, we need to wait to perform energy normalization until the end of speech is detected, which is a barrier for the real-time implementation of ASR. However, it

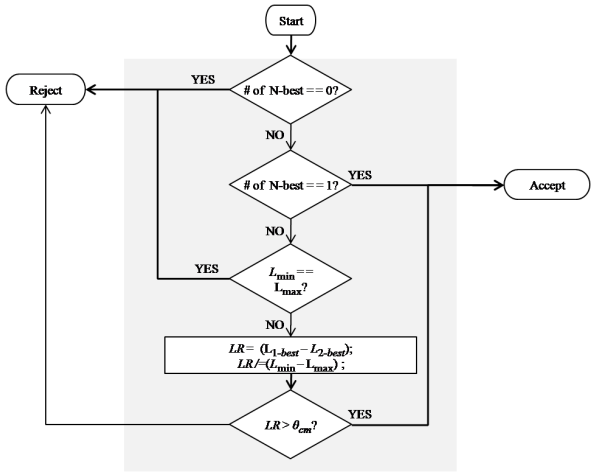

Fig. 3. Procedure of the utterance verification based on a likelihood ratio test.

is generally recognized that normalized feature vectors can provide better ASR performance than the unnormalized ones. Therefore, as a compromise between real-time operation and ASR performance, we create an N-best list from the first-pass Viterbi decoding. In this paper, we obtain a 20-best list from the first-pass Viterbi decoding; the second-pass Viterbi decoding that uses the normalized feature vectors subsequently yields the actual word recognition results from 20 candidate words.

TABLE I
EXAMPLES OF WORD-TO-SCENE-MODE MAPPING, WHICH IS A PART OF THE TABLE, $TABLE_{SCENE\text{-}MODE}$.

Scene-related word	Scene mode (preference)		
	1st	2nd	3rd
skate	sports	fast shutter	snow
ski	snow	sports	fast shutter
spaghetti	food	auto	portrait
food	food	food	food

D. Utterance verification

To verify the results of the second-pass Viterbi decoding, a likelihood ratio test is applied. Fig. 3 shows the procedure of the utterance verification in the proposed system. As shown in the figure, we first examine the number of words in the N-best list. If the number is equal to zero, the utterance is rejected because there is no word recognized from the Viterbi decoding. On the other hand, the utterance is accepted if the number of words in the N-best list is only one.

Next, we obtain the minimum log likelihood, $L_{\min}$, and the maximum log likelihood, $L_{\max}$, by comparing the Viterbi scores of words in the N-best list. The likelihood ratio (LR) is then calculated by using the equation of

$$LR = \frac{L_{1-best} - L_{2-best}}{L_{\max} - L_{\min}} \tag{3}$$

where $L_{1\text{-}best}$ and $L_{2\text{-}best}$ are the likelihoods of the words ranked as the first place and the second place from the N-best list, respectively. Finally, we verify the recognition results by comparing LR with two predefined threshold θ_{cm}. In other words, the recognition result is accepted if $LR > \theta_{cm}$; otherwise, it is rejected. In this paper, we set θ_{cm} as 0.2. The accepted word is then mapped into a proper scene mode by following the procedure described in the next subsection.

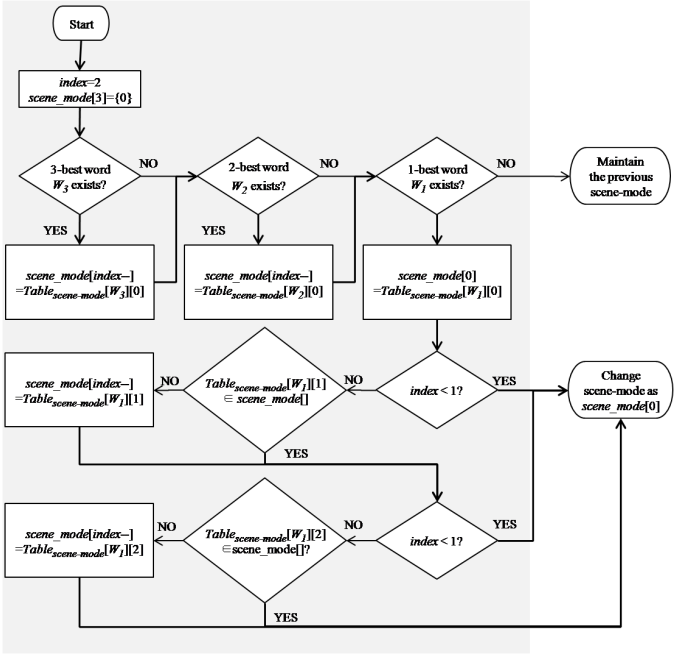

Fig. 4. Procedure of mapping an accepted word into its corresponding scene-mode.

E. Word-to-scene-mode mapping

Assuming that the user does not know the exact name of a scene-mode, we create a mapping table to transform the accepted word into its corresponding scene-mode. First, we select nine scene-modes: 'auto', 'sports', 'spotlight', 'portrait', 'beach', 'snow', 'fast shutter', 'food', and 'waterfall'. Second, words related to each scene-mode word are collected from websites for photo galleries or product review sites regarding camcorders and digital cameras, and a list of 200 scene-mode or scene-mode related words is generated. For example, the scene-related words for the scene-mode 'sports' are 'badminton', 'marathon', 'tennis', 'skate', etc. Third, each scene or scene-related word is mapped into its three most preferable scene-modes. Thus, the size of the scene-mode mapping table, referred to as '$Table_{scene\text{-}mode}$', is a 200 x 3 matrix, a part of which is shown in Table I. As shown in the first row of the table, for the word 'skate', we set 'sports', 'fast shutter', and 'snow' as the first, second, and third most preferable scene-modes, respectively. Of course, a scene word can be recognized and the mode change can be performed into the scene-mode, as shown in the last row of the table.

Fig. 4 shows the procedure of mapping an accepted word into the corresponding scene-mode from three preferable scene-modes. For an accepted word W, we look up three scene-modes from $Table_{scene\text{-}mode.}$ That is, we have $scene\_mode[i]=Table_{scene\text{-}mode}[W][i]$ for i=0,1,2. Next, the scene-mode of the device actually changes to $scene\_mode[0]$

with the scene-modes *scene_mode*[1] and *scene_mode*[2] being listed in the display panel as alternatives. Note that if the portable digital imaging device supports a touch pad, one of the listed scene-modes could be selected by a user at this time.

IV. IMPLEMENTATION ISSUES OF THE SERVICE ON A PORTABLE DIGITAL IMAGING DEVICE

In this section, we first discuss implementation issues to achieve the real-time operation of the proposed system on a portable digital imaging device equipped with limited resources; an ARM processor with a CPU clock of 133 MHz and a memory size of 17 MB. In particular, we explain several optimization techniques that are applied to reduce the computational complexity and required memory size. In addition, we also discuss how to improve ASR performance by adapting acoustic models to the characteristics of the microphones and operating environments. Moreover, a graphical user interface (GUI) is explained, which is designed to assist the users of this service.

A. Complexity optimization

In order to implement the proposed voice-driven scene-mode recommendation system in real-time on an ARM processor, we first carry out fixed-point arithmetic programming because fixed-point operation is much faster than its corresponding floating-point operation even if the ARM processor can support both fixed-point and floating-point arithmetic. In other words, most 64-bit or 32-bit floating variables (*double*, *float*) are converted into 32-bit or 16-bit integer variables (*int*, *short*) in the source code for speech recognition feature extraction, VAD, and the two-pass Viterbi decoding.

Next, we perform complexity reduction at an algorithmic level. In the two-pass Viterbi decoding, intensive computations are required to compute the log-likelihood of multivariate Gaussian distributions. Let us suppose that $\log P(x)$ is the log-likelihood of a d-dimensional observation vector x with respect to a d-dimensional multivariate Gaussian mixture distribution with K mixtures. Then, $\log P(x)$ is computed as

$$\log P(x) = \log \sum_{k=1}^{K} \left(\frac{w_k}{\sqrt{(2\pi)^d |\Sigma_k|}} \exp\left(-\tfrac{1}{2}(x-\mu_k)\Sigma_k^{-1}(x-\mu_k)^T\right) \right) \quad (4)$$

where T denotes a vector transpose, and w_k, μ_k, and Σ_k are the weighting coefficient, the mean vector, and the covariance matrix of the k-th Gaussian mixture, respectively. Assuming that the covariance matrices are diagonal, Eq. (4) can be rewritten as

$$\log P(x) = \log \sum_{k=1}^{K} \left(A_k \exp B_k(x) \right) \quad (5)$$

TABLE II
COMPARISON OF THE PROCESSING TIME PER FRAME OF VITERBI DECODING BEFORE AND AFTER THE FIXED-POINT ARITHMETIC PROGRAMMING AND THE ALGORITHMIC LEVEL OPTIMIZATION

Acoustic models	Unnormalized acoustic models	Normalized acoustic models
Floating-point arithmetic programming	17.377 ms	6.229 ms
Fixed-point arithmetic programming	6.993 ms	1.280 ms
Fixed-point operation + algorithmic level optimization	6.050 ms	1.093 ms

where $A_k = w_k \Big/ \sqrt{(2\pi)^d \prod_{i=1}^{d} \sigma_{k,ii}^2}$, $B_k(x) = -\frac{1}{2}\sum_{i=1}^{d} \frac{(x_i - \mu_{k,i})^2}{\sigma_{k,ii}^2}$, $x = (x_1, \cdots, x_d)$, and $\sigma_{k,ii}^2$ is an element of the i-th row and the i-th column in the covariance matrix, Σ_k. Thus, $\log P(x)$ can be represented with the recursive form of

$$\begin{aligned} \log P(x) &= \log P_K(x) \\ &= \log P_{K-1}(x) + \\ &\quad \log\left(1 + \exp\left(\log A_K + B_K(x) - \log P_{K-1}(x)\right)\right) \end{aligned} \quad (6)$$

where $\log P_K(x)$ is a summation of logarithmic probability from the first Gaussian to the K-th Gaussian. Note that from Eq. (6), $\log A_k$ (for $k = 1, \cdots, K$) can be pre-computed since $\log A_k$ is independent on the observation vector x.

In addition, we compare $\log A_K + B_K(x)$ and $\log P_{K-1}(x)$. Here, if the absolute value of the difference between $\log A_K + B_K(x)$ and $\log P_{K-1}(x)$ is greater than a predefined threshold, $\log P_{thr}$, then we can approximate Eq. (6) as

$$\log P_K(x) \approx \begin{cases} \log A_K + B_K(x), & \textit{if } \log A_K + B_K(x) > \log P_{K-1}(x), \\ \log P_{K-1}(x), & \textit{otherwise.} \end{cases} \quad (7)$$

Moreover, the division operations for computing $B_k(x)$ are replaced with a multiplication because $1/\sigma_{k,ii}^2$ can be pre-computed and stored in memory [11].

Table II compares the processing times of Viterbi decoding with 50 scene-related words when the floating-point arithmetic programming and the fixed-point one before and after the algorithmic level optimization was performed for both unnormalized and normalized acoustic models. Here, d=39, and K=3 and 4 for unnormalized acoustic models and normalized acoustic models, respectively, and $\log P_{thr}$ = -7. As shown in the table, the processing times per frame (10 ms) for the unnormalized acoustic model were measured as 17.377 ms,

6.993 ms and 6.060 ms. These results yielded a 59.76% and 65.18% processing time reduction for the fixed-point arithmetic programming and algorithmic level optimization, respectively, compared to the floating-point arithmetic implementation. Similarly, when we compared the processing time of Viterbi decoding implemented with floating-point arithmetic, we achieved relative processing time reductions of 79.45% and 82.45% by the fixed-point arithmetic programming and the algorithmic level optimization.

TABLE III
COMPARISON OF MEMORY SIZE FOR THE ACOUSTIC MODELS BEFORE AND AFTER THE MEMORY OPTIMIZATION

Comparison / Acoustic model	Without optimization	With optimization
Energy-unnormalized acoustic model	19.12 MB	1.71 MB
Energy-normalized acoustic model	28.41 MB	2.27 MB
Total	47.53 MB	3.98 MB

Next, we further reduced the computational complexity by investigating the characteristics of the processor and instructions. According to the user's guide on the ARM11 processor [12], the structures of the source codes were modified. That is, we minimized branch statements such as '*if*' statements, '*switch'* statements. In addition, arithmetic operations with two 32-bit variables were restrained or replaced with those having 32-bit and 16-bit variables or two 16-bit variables [13].

B. Acoustic model optimization

Memory limitation is a primary concern for the successful implementation of the proposed system on a portable digital imaging device. Actually, the portable imaging device we used only has a total memory size of 17 MB, of which it uses about 10 MB for the operating system and system management. Thus, only about 7 MB is available for the proposed system. However, the memory sizes of the original energy normalized and unnormalized acoustic models are about 28 MB and 19 MB, respectively—in this case, our system cannot be realized on the device.

In order to reduce the memory size of the proposed system, we optimized the number of the triphone acoustic models and the dimension of the multivariate Gaussian distribution. That is, original energy normalized and unnormalized acoustic models consisted of 10,138 and 14,751 state-tied triphone models, respectively, and each state-tied triphone model was trained with a multivariate Gaussian distribution with four mixtures. Among them, the triphone models used for scene related words were first extracted. Second, the dimensional of a multivariate Gaussian distribution was reduced from four to three for the energy unnormalized acoustic models. By reducing the dimension of a multivariate Gaussian distribution for the energy unnormalized acoustic models, the computational complexity could also be reduced.

Table III shows the memory size of the energy normalized and the energy unnormalized acoustic models with and without optimization. Before memory optimization, the total memory size of the acoustic models was 47.53 MB, and thus we could not operate the voice-driven scene-mode recommendation system. On the other hand, our memory optimization allowed the system operation since it reduced the total memory size of the acoustic models to 3.98 MB that was significantly less than 7 MB.

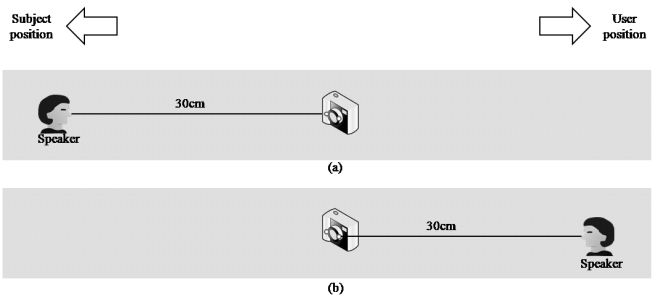

Fig. 5. Recording environment for developing a speech adaptation database, where adaptation speakers locate (a) either in the front of the microphones or (b) opposite to the microphones.

C. Acoustic model adaptation

On a portable digital imaging device, the position and the characteristics of microphones can degrade the performance of speech recognition. In a view of the microphone position, environmental noises are straightforwardly captured since the microphones face the subject, not the user. Moreover, operational noises such as focusing or zoom motor noise can also be easily recorded because portable digital imaging devices tend to be small in size and thus the components are tightly integrated. In this case, however, inexpensive microphones or omni-directional microphones can be mounted on imaging devices for a moderate price.

For these reasons, environmental noise adaption techniques such as maximum a posteriori (MAP) and maximum likelihood linear regression (MLLR) adaptation methods [3] were used to adjust the baseline acoustic models to these different acoustic environments. Actually, acoustic models in this work were initially trained with a speech database that was developed by using a close-talking microphone. Thus, the adaptation for different recording environments should be performed by collecting the speech adaptation database.

There were two different recordings carried out as shown in Fig. 5. First, nine adult speakers (7 males and 2 females) uttered 200 scene-related words six times in the front of the portable digital imaging device [Fig. 5(a)] so that the acoustic models could be adjusted to the characteristics of the microphones. And then, the nine speakers uttered the words three times behind the device [Fig. 5(b)] for the real position of the microphones. It was noted here that a microphone was mounted on the portable digital imaging device under office noise conditions.

D. Graphical user interface

The graphical user interface (GUI) is an important part of consumer electronics as it provides users with important information and guidelines. Therefore, we implemented a GUI

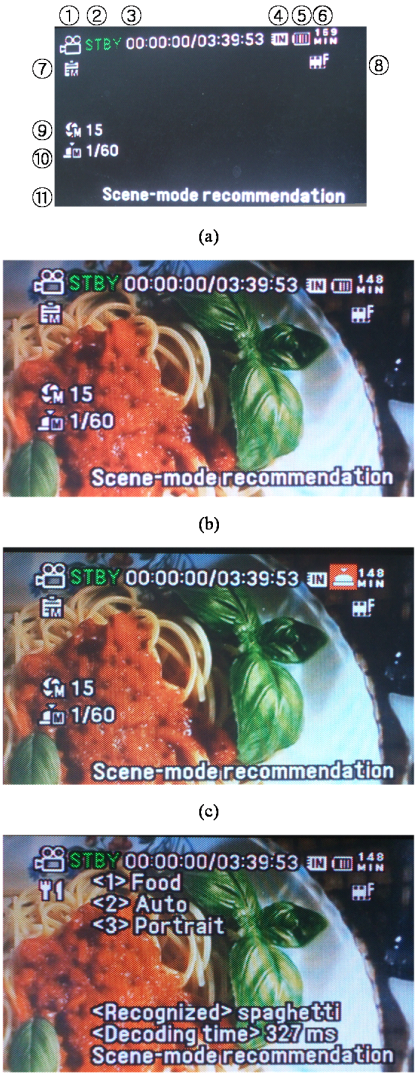

Fig. 6. Snapshot of graphical user interface (GUI) of the proposed voice-driven scene-mode recommendation system implemented on a portable digital imaging device; (a) the main GUI configuration, (b) the initial GUI when the device boots up, (c) the GUI when the soft PAT button is activated, and (d) the GUI when a user utters the word 'spaghetti'.

on the portable digital imaging device for the voice-driven scene-mode recommendation system. Fig. 6(a) shows a snapshot of the main GUI, where □ to □ in the top line display imaging mode, operating status, time counter (recording time/remaining recordable time), storage media, remaining battery level, and soft PAT button status, respectively. In addition, □ to □ then show the scene-mode, image resolution, manual exposure, manual shutter, and the system name.

According to the implementation described in Section III, the system displays different GUIs as follows. Fig. 6(b) shows a snapshot of the initial GUI on the LCD after the portable digital imaging device boots up. In particular, since the manual scene-mode is set as the initial scene-mode, the manual exposure and shutter information are displayed. Next, a pressed button-shaped icon is displayed immediately after the soft PAT button is activated [Fig. 6(c)]. Finally, three preferable recommended scene-modes as well as the recognized word are displayed on the LCD for a moment after the scene-mode is determined. Fig. 6(d) illustrates a snapshot of the GUI when a user utters the word 'spaghetti'. As shown at the bottom of the figure, 'spaghetti' is displayed along with the processing time. Note here that the processing time is measured as 327 ms, which confirms that the implemented system can provide the recommendation service in real time. In addition, three recommended scene-modes are displayed: '<1> Food', '<2> Auto', and '<3> Portrait', in this example. Finally, the scene-mode is changed to the first recommended scene-mode, 'Food'. Simultaneously, an icon for the scene-mode 'Food' is displayed on the LCD, as shown at the top left of Fig. 6(d) [location □ of Fig. 6(a)].

TABLE IV
SPECIFICATION OF THE PORTABLE DIGITAL IMAGING DEVICE USED FOR THE IMPLEMENTATION OF THE PROPOSED SYSTEM

Item	Description
Embedded OS	Montavista
CPU model	ARM1176JZF-S Processor
CPU clock	133 MHz
RAM	17 MB
Cache	16 KB

V. PERFORMANCE EVALUATION OF THE IMPLEMENTED VOICE-DRIVEN SCENE-MODE RECOMMENDATION SYSTEM

In this section, we evaluated the performance of the voice-driven scene-mode recommendation system implemented on a portable digital imaging device. The performance of the system was measured in terms of speech recognition accuracy, scene-mode recognition accuracy (SMRA), and processing time. Table IV briefly describes the specifications of the portable digital imaging device used for the implementation. Note that Montavista was used as the embedded operating system (OS) with CPU clock and RAM size of 133 MHz and 17 MB, respectively. It had also an eight-stage pipeline, a branch-prediction with return stack, and a vector floating-point coprocessor.

A. Scene-related word recognition performance

The baseline ASR system was trained with a subset of the read sentence speech database (CleanSent01) by Speech Information Technology & Industry Promotion Center (SiTEC) [14]. The CleanSent01 database consisted of 20,806 utterances that were recorded in a sound-proof room using AKG C414 ULS and Sennheiser close-talking microphones. In addition, the utterances were sampled at rate of 16 kHz with a resolution of 16 bits/sample. For every speech analysis frame of 10 ms long, a 39-dimensional speech feature vector was extracted, as mentioned in Section III-B.

The acoustic models were based on 3-state left-to-right, context-dependent, 4-mixture, and cross-word triphone hidden Markov models (HMMs), and trained using the HTK 3.2

TABLE V
COMPARISON OF AVERAGE PROCESSING TIME FOR DIFFERENT NUMBER OF SCENE-RELATED WORDS IN NOISY ENVIRONMENTS

No. of scene-related words / Environment	50	100	150	200
Office noise	500 ms	500 ms	600 ms	800 ms
Music noise	500 ms	600 ms	800 ms	1200 ms

TABLE VI
COMPARISON OF SCENE-MODE RECOMMENDATION ACCURACY OF THE VOICE-DRIVEN SCENE-MODE RECOMMENDATION SERVICE FOR 50 AND 200 SCENE-RELATED WORDS

No. of scene-related words	50	200
Scene-mode recommendation accuracy	98.0%	96.8%

toolkit [15]. All the triphone models were expanded from 42 monophones, which also included a silence model and a pause model, and the states of the triphone models were tied by employing a decision tree [16]. As a result, the acoustic models were composed of 74,090 triphones. Actually, we trained two different sets of acoustic models for the baseline ASR system according to energy normalization, i.e., energy-unnormalized acoustic models and energy-normalized acoustic models having 14,751 and 10,138 HMM states, respectively.

B. Processing time

Several steps of optimization process to reduce computational complexity were applied to overcome the resource limitation of the ARM processor. We then measured the average processing time of the system depending on the number of scene-related words in two different background environments, such as office noise and music noise. Table V show the comparison of the average processing times for different numbers of scene-related words under office and music noise conditions. It was shown from the table that the voice-driven scene-mode recommendation system responded within around 500 ms for 50 scene-related words in both background environments, though the system required more processing time as the number of scene-related words was increased.

C. Accuracy of the voice-driven scene-mode recommendation System

In order to evaluate the accuracy of the success mode change of the system, we measured the scene mode recognition accuracy (SMRA). Here, SMRA corresponded to the accuracy that the desired scene-mode was included in the list of three recommended scene-modes. Table VI shows the average SMRAs of the proposed voice-driven scene-mode recommendation service for 50 and 200 scene-related words. In this case, the average SMRAs were 98.0% and 96.8% for the 50 and 200 scene-related words, respectively.

VI. CONCLUSION

In this paper, we proposed a voice-driven scene-mode recommendation service as a means of selecting a desired scene-mode on a portable digital imaging device. The service was designed to recommend or automatically change a scene-mode by recognizing a user's voice. Here, a normal soft push-and-talk button was used to initiate the service, but deactivation of the button was performed by a combination of releasing the button or determining the end of an utterance by VAD. Next, we implemented the voice-driven scene-mode recommendation system, which was composed of seven parts: voice input acquisition, speech recognition feature extraction, voice activity detection, automatic speech recognition, utterance verification, word-to-scene-mode mapping, and scene-mode activation. In order to realize the system in real time on a portable digital imaging device with limited resources, several optimization techniques to reduce the computational complexity and required memory size were applied. Moreover, acoustic model adaptation was performed to compensate for the characteristics of the microphones and operating environments. The system performance was then measured in terms of processing time and scene-mode recognition accuracy (SMRA). It was shown from the experiments that the average processing time and the average SMRA were around 500 ms and 98.0% for 50 scene-related words, respectively, and 1200 ms and 96.8% for 200 scene-related words, respectively.

REFERENCES

[1] W. Ku, M. S. Kankanhalli, and J.-H. Lim, "Using camera settings templates ("scene modes") for image scene classification of photographs taken on manual/expert settings," *Lecture Notes in Computer Science*, vol. 4810, pp. 10-17, Dec. 2007.

[2] R. Sharma, V. I. Pavlovic, and T. S. Huang, "Toward multimodal human-computer interface," *Proceedings of IEEE*, vol. 86, no. 5, pp. 853–869, May 1998.

[3] D. Zavagliakos, R. Schwartz, and J. McDonough, "Maximum a posteriori adaptation for large scale HMM recognizers," in *Proc. ICASSP*, pp. 725–728, May 1996.

[4] Y. R. Oh, J. S. Yoon, J. H. Park, and H. K. Kim, "A name recognition based call-and-call service for home robots," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 2, pp. 247–253, May 2008.

[5] G. S. Ying, C. D. Mitchell, and L. H. Jamieson, "Endpoint detection of isolated utterances based on modified Teager energy measurement," in *Proc. ICASSP*, pp. 732–735, Apr. 1993.

[6] L. R. Rabiner and M. R. Sambur, "An algorithm for determining the endpoints of isolated utterances," *Bell System Technical Journal*, vol. 54, no. 2, pp. 297–315, Feb. 1975.

[7] J. Lee, S. Baek, and K.-M. Sung, "A study on the endpoint detection algorithm based on a modified Teager energy," in *Proc. Acoustical Society of Korea Conference*, vol. 117, no. 2(s), pp. 407–410, Nov. 1998.

[8] N. Jakovljevic, M. Janev, D. Pekar, and D. Miskovic, "Energy normalization in automatic speech recognition," *Lecture Notes in Computer Science*, vol. 6246, pp. 341–347, Sept. 2008.

[9] A. Lee, T. Kawahara, and S. Doshita, "An efficient two-pass search algorithm using word trellis index," in *Proc. ICSLP*, pp. 1831–1834, Nov. 1998.

[10] M. Novak, R. Hampl, P. Krbec, V. Bergl, and J. Sediw, "Two-pass search strategy for large list recognition on embedded speech recognition platforms," in *Proc. ICASSP*, pp. 6–10, Apr. 2003.

[11] M. J. Flaherty and T. Sidney, "Real time implementation of HMM speech recognition for telecommunications applications," in *Proc. ICASSP*, vol. 6, pp. 145–148, Apr. 1994.

[12] ARM Limited, *ARM1176JZF-S Technical Reference Manual*, Mar. 2006.

[13] Z.-H. Tan and B. Lindberg, *Automatic Speech Recognition on Mobile Devices and Over Communication Networks*, Springer-Verlag, London, U.K., Apr. 2008.

[14] Y.-J. Lee, B.-W. Kim, and Y. Um, "Speech Information Technology & Industry Promotion Center in Korea: Activities and Directions," in *Proc. the 3rd International Conference on Language Resources and Evaluation (LREC)*, pp. 1851–1854, June 2002.
[15] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.2)*, Microsoft Corporation, Cambridge University Engineering Department, Dec. 2002.
[16] S. Young, J. Odell, and P. Woodland, "Tree-based state tying for high accuracy acoustic modeling," in *Proc. ARPA Human Language Technology Workshop*, pp. 307–312, Mar. 1994.

Yoo Rhee Oh (S'06) received a B.S. degree in Computer Science from Kyungpook National University, Korea in 2004, and an M.S. degree in Information and Communications Engineering from the Gwangju Institute of Science and Technology (GIST), Korea in 2006. She is now a Ph.D. student at GIST. Her current research interests include Korean speech recognition, non-native speech recognition, and the system integration of an embedded system.

Jae Sam Yoon (S'06) received a B.S. degree in Electronics Engineering from Kwangwoon University, Korea in 2004, and an M.S. degree in Information and Communications Engineering from Gwangju Institute of Science and Technology (GIST), Korea in 2006. He is now a Ph.D. student at GIST. His current research interests include robust speech recognition in noisy environments and speech enhancement.

Hong Kook Kim (M'99-SM'01) received a B.S. degree in Control and Instrumentation Engineering from Seoul National University, Korea in 1988. He then received both M.S. and Ph.D. degrees in Electrical Engineering from the Korea Advanced Institute of Science and Technology (KAIST), Korea in 1990 and 1994, respectively. He was a senior researcher at the Samsung Advanced Institute of Technology (SAIT), Kiheung, Korea, from 1990 to 1998. During 1998–2003, he was a senior technical staff member with the Voice Enabled Services Research Lab at AT&T Labs-Research, Florham Park, NJ. Since August 2003, he has been with the Department of Information and Communications, at the Gwangju Institute of Science and Technology (GIST) as a professor. His current research interests include speech recognition and coding, audio coding and 3D audio, and embedded algorithms and solutions for speech and audio processing for handheld devices.

Myung Bo Kim received a B.S. degree in Electrical Engineering from Yonsei University, Korea in 1991. He then joined the Camcorder R&D Team at Samsung Electronics Co. Ltd. in 1991. He received an M.S. degree in Electrical and Electronic Engineering from Yonsei University in 2004. He is now a principal engineer in charge of SW Management in camcorder systems at Samsung Electronics.

Sang Ryong Kim received M.S. and Ph.D. degrees in Electrical Engineering from the Korea Advanced Institute of Science and Technology (KAIST), Korea in 1982 and 1989, respectively. He was affiliated with Samsung Electronics, Co. Ltd., as a technical research staff from 1989 to 1993. During 1993–2007, he was a Vice President with the Human and Computer Interaction (HCI) Lab and the Interaction Lab at the Samsung Advanced Institute of Technology (SAIT), Korea. Since 2007, he has been an R&D Manager and an Executive Vice President at Samsung Electronics, Co. Ltd., Korea. His research interests include signal processing for digital imaging devices and advanced network technologies for consumer electronics.