RESEARCH ARTICLE OPEN ACCESS

Transmission by an Embedded System with Enhancements in Voice Processing Technologies

G.Sitha Annapurna[1] ,B.Mamatha[2], P.Annapurna[3],
[1]B.Tech, M.Tech Assistant Professor in ECE Dept., Sri vasavi Institute of Engineering and Technology, Nandamuru, Pedana (MD),Krishna D.t,A.P
[2]B.Tech, M.Tech Assistant Professor in ECE Dept., Sri vasavi Institute of Engineering and Technology, Nandamuru, Pedana (MD),Krishna D.t,A.P
[3]B.Tech,(M.Tech) Assistant Professor in ECE Dept., Sri vasavi Institute of Engineering and Technology, Nandamuru, Pedana (MD),Krishna D.t,A.P

Abstract
The paper reports that the robot can transmit the data such as video, audio, images. The robot can be controlled using the human voice. There are two embedded systems first one is robot controlling system(MASTER) which is used to control the robot, second one is voice controlled robot(SLAVE) which responds according to the instructions coming from the controlling system. These two embedded systems are communicated through wireless. We can use anyone one of wireless protocols such as IR, NFC, Bluetooth, Zigbee, WI-FI in order to establish a bridge between the MASTER and SLAVE. The voice controlled robot can understand the instructions with the help of the voice recognition system. Spinx-4 is a speech recognizer system written entirely in the java programming language.Sphinx-4 started out as a port of Sphinx-3 to the Java programming language, but evolved into a recognizer designed to be much more flexible than Sphinx-3, thus becoming an excellent platform for speech research.Sphinx-4 is an HMM-based speech recognizer. HMM stands for Hidden Markov Models, Sphinx-4 are a type of statistical model In HMM based speech recognizers.
Index Terms – Embedded system; voice processing; wireless transmission

I. Introduction

There have been considerable efforts to achieve the hand-free control of computers. There are usually two ways to control computers and robots on a hand free basis voice commands and gesture commands. These hand-free input methods have been considered as an alternative to the traditional keyboard inputs. The main objective of this paper is to implement a voice recognition system onto a robot for controlling its movement with a high degree of accuracy and robustness. The users will be able to control the robot by either speaking directly to the robot or speaking to a microphone connected to a computer, which will extract the command and pass it to robot through wireless network connection. It is also possible to transmit the data from voice control robot to controlling system. The data which is sent by robot is received by personal computer. We can see on the display of the personal computer whatever sent by the voice control robot.

II. Wireless protocol

2.1. Introduction

Bluetooth's primary function is to replace cables. It's a technology built around short-range radio links between mobile PCs, mobile phones and other portable devices. The technology promises to eliminate the confusion of cables, connectors and protocols confounding communications between today's high tech products.

2.2. Specifications:

The Following specifications are required

2.2.1 Open specification:

The Bluetooth Special Interest Group (SIG) has produced a specification for Bluetooth wireless communication that is publicly available and royalty free.

2.2.3 Short-range wireless:

Today, much of the communication takes place over cables. There are a wide variety of connectors with many combinations of shapes, sizes and number of pins. These cables can become quite burdensome to users. With Bluetooth technology, these devices can communicate without wires over air using radio waves to transmit and receive data. Bluetooth wireless technology is specifically designed for short-range (nominally 10 meters) communications. one result of this design is very low power consumption, making the technology well suited for use with small, handheld devices that typically are powered by batteries.

2.2.4Voice and data:

Voice is now commonly transmitted and stored in digital formats. Bluetooth wireless communication makes provisions for both voice and data, and thus it is an ideal technology for enabling all sorts of devices to communicate using either or both of these content types.

2.2.5 Anywhere in the world:

Many forms of wireless communications are regulated in many parts of the world; radio frequency spectrum usage often requires a license with strict transmission power obligations. Bluetooth wireless communications operate within a chosen frequency spectrum that is unlicensed throughout the world (2.4 GHz). Thus, devices that employ Bluetooth wireless communication can be used unmodified, no matter where a person might be.

2.3. History

Bluetooth was invented in 1994 by L. M. Ericsson in Sweden. Japp Hartseen is one of the co-inventors to Bluetooth; he says that the technology came about almost by accident. The original intention was to make a wireless connection between something like an earphone and a cordless headset and the mobile phone. When they realized that they could tap into a low radio frequency that required no licensing and was available to anyone in the world that wanted it, Hartseen and his colleagues at Ericsson began to experiment with different computer chips. They developed small radio chips that made wireless connection between devices containing them. To encourage adoption of Bluetooth and with hope of developing a global standard, Ericsson decided to give the specification for the technology away for free. When the technique was out there, The Bluetooth Special Interest Group (SIG) was formed. The members were Toshiba, IBM, Ericsson, Nokia and Intel. Today SIG have over 1200 company members.

As VLSI technology advances and computing power grow in the past twenty years, robots became more and more intelligent, robust and consumed less and less power. Moreover, they are required to handle more and more so called teamwork. It means that they must be developed to possess the capability of constructing a network and performing cooperative works. A key driving force in the development of cooperative mobile robotic system is their potential for reducing the need for human presence in dangerous applications [1]. It is possible to establish a point to point connection using Zigbee protocol. Instead of point to point connection, In order To communicate with more than one robot we use a Bluetooth Wi-Fi integrated chip. So that it is possible to control the robots by selecting the appropriate robots.[2-3]

III. Wireless data transmission

3.1 HARDWARE SYSTEM:

This system mainly adopts B/S structure, embedded WEB server as the center, and captures the video by USB camera, then carries on MPEG-4 compression. The compressed MPEG-4 frame RTP package is transmitted through RTP protocol. As a result, the user can see the video image directly that come from WEB server through the IE browser in the monitoring terminal. Figure.1 shows the hardware construction of the system for remote video capture and transmission based on ARM. We select the ARM9 development board, provided by Beijing Hengfeng Ruike Technology Co., Ltd, as the hardware platform of the system. This system bases on embedded chip S3C2410AL constructed on ARM9, its basic frequency can be up to 203MHz, and the board carries 64MB SDRAM and 64MB NANDFLASH. The resources of the main board mainly include: two USB host interface, a USB device interface, a 10M/100M Ethernet port, a serial port, a JTAG interface, touch screen and so on.

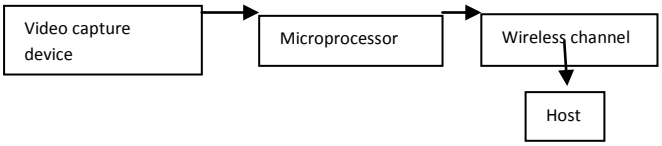

Figure.1 Hardware construction of the system for remote video capture and transmission based on ARM

3.2 SOFTWARE SYSTEM:

The platform of software system is embedded operating system whose kernel is Linux-2.6.30.4. Embedded Linux operating system has quite a lot of advantages. The source code of operating system opens completely, can be cut out and has low cost; the kernel is small and stable, which of function is powerful; the operating system supports various hardware platform and multitasking processe, which can provide better performance of real time. These characteristics are very suitable for embedded application, and Linux itself provides complete TCP/IP protocol directly, which can be easily applied to network.

3.2.1 A. Module of Embedded Web Server

In numerous Web server, there are many lightweights, such as boaweb, thttp, httpd, etc. however, if you want to get higher security and provide convenient embedded Web server for the last Web development period, those servers cannot meet the case. So we select the recently released AppWeb3.0, which has a high level security, to

construct embedded Web server. The architecture of the AppWeb3.0 is shown in Fig. 2.

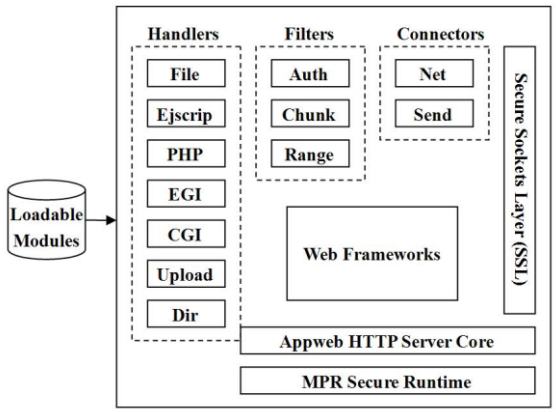

Figure 2. Appweb 3.0 architecture.

AppWeb3.0 is open source Web server, which completely follows GPL(GNU General Public License) software license agreement. The software adopts modular design to realize function of the groupware. This version improves the performance of the system, introduces the JavaScript framework at the server side, and supports Windows CE, Windows 7, Vista, Mac OS X, and Free BSD. When running, it only needs 800K bytes of memory. And more, it adds JavaScript and Ejscript framework for self developing, supports dynamic pages make, CGI, SSL that can be loaded module, ID authentication of the type of summary, single and multi-threaded applications, and also supports MVC(model-view-controller)pattern, providing the ORM(Object Relational Mapping). So it can be used in the embedded system perfectly. When carrying on the AppWeb transplant on the development board, some environment variables and configuration options need to be modified.

- We set CC, AR, LD, RANLIB, STRIP (environmental variables) to the corresponding location of crosscompilation tool, and set CC_FOR_BUILD to the location of host. For example: as at the shellcommand in Linux system, we input: export CC=/usr/local/arm/3.4.1/bin/arm-linux gcc, other similar.
- After setting, we carry on configure, for which many parameters are provided in the AppWeb website. The type of the goal board is set as "arm-s3c2410- linux" through "—host" parameter. In order to carry on the cross-compiling on the PC installed Linux, "—build" parameter set as "i686-pc-linux".
- After completing configuration, we carry out the commands of "make" and "make install", then download the homepage required by AppWeb, the CGI procedure and execution document of AppWeb to the development board. Finally, we configure IP and start AppWeb to visit the server through the IE browser.

3.2.2. Module for Video Capture based on V4L2

Compared withV4L, V4L2(Video4Linux2) provides with better extension and flexibility, and supports more hardware device. However, V4L2 has made the thorough transformation based V4L, so both are incompatible to each other. V4L2 provides a set of unified API for the video application procedure, which can operate all kinds of different video capture devices by calling the standard system function, greatly simplified the development and maintenance of video system. V4L2 provides the following interfaces:

- Video capture interface (/dev/video)
- Video overlay interface (/dev/video)
- Video input interface (/dev/video)
- Video output interface (/dev/video)
- Codec interface (/dev/video)
- Effect devices interface (/dev/video)
- Raw vertical blancking interval (VBI) interface (/dev/vbi)
- Sliced VBI data interface (/dev/vbi)
- Teletext interface (/dev/vtx)
- Radio interface (/dev/radio)
- Radio data system (RDS) interface (/dev/radio) [5]

In this article, we will cover usage of the video capture interface, provided by the /dev/video character device. The flow chart of Video capture is shown in Figure. 3.

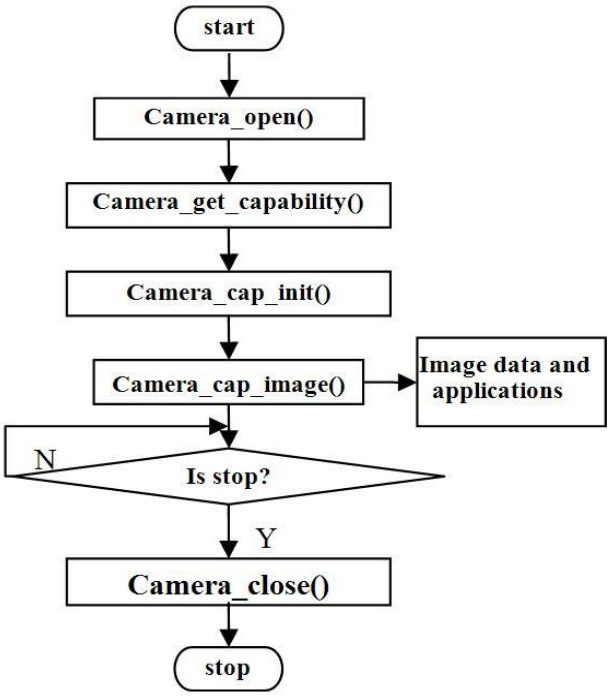

Figure 3. Flow chart of Video capture.

Most of functions of V4L2 is completed through the ioctl system call, the format of the grammar is: ioctl (int fd, int request, void *argp). Here, "fd" is file descriptor, which obtains through open( ) function; "request" is a command flag, telling system to do something; "argp" is user data pointer, transmitting the parameters or receiving the data. The main functions in the video capture procedure represent as follows:

- Camera_open( )：call open( ) function to open a device;
- Camera_get_capability ( ): Through calling ioctl( ) function, we obtain the related information of the device document using the VIDIOC_QUERYCAP command flag, and save to the struct variable of "struct v4l2_capability".
- Camera_cap_init ( ): call ioctl( ) function to set the video format and frame format by the command flags of the VIDIOC_S_STD and VIDIOC_S_FMT in this function. Then, apply for the frame buffer for the video data, the number of which is generally not more than 5. After that, we map the frame buffer to the user space and add it in queue by the function of mmap( ), preparing for saving video data.
- Camera_cap_image( )： call ioctl( ) function to caputure vedio data by the command flag of VIDIOC_STREAM, and pop the frame buffer whichhas video data from the queue through the command flag of VIDIOC_DQBUF, then re-add the frame buffer that has popped data to the queue tail with the command flag of VIDIOC_QBUF, so as to collect circularly.
- Camera_close( )： use the command flag of VIDIOC_STREAMOFF to stop video capture, then call close(fd) fuction to close video device.

3.2.3 Module of Compression and Coding for Video

In order to transmit the video data via internet , we have to compress and code the primitive image before transmission. We use MPEG-4 video codec here, because the MPEG-4 compression and coding technology has been already mature and especically suits for video transmission in the condition of low bandwidth. When capturing video, the object in motion is clear and the background is wheelchair-bed in this system. Since this system has these characteristics, in order to improve the compression ratio of video code to achieve a better performance, we can optimize the algorithm based on the block match movement estimation in MPEG-4. The ideals and methods are as follows：

• Identify the movement regions and non-movement areas. Modifying the MPEG-4 encoder slightly in the aspect of the movement estimation, we can detect the movement regions and non-movement areas. The method is that the macro blocks whose motion vector is zero are considered as a nonmovement regions and the rest as moving regions. Certainly, when sampling two frames as motion detection, the two frames may not be continual, but we 'd better to take the two primitive frames. If you take the restructuring frames, it will has a great distortion with the real scene, and a bad effect.

• Enhance speed of the motion detection using the method of sub-regional and recurrent motion detection. First, the rectangular video frame is divided into four rectangular areas in the same size, then carry on motion detection for each region. According to the necessity of the detection, each region may also use the same method to carry on the division and the motion detection again, and the depth of division is predetermined. If there is a region whose movement vectors of all the macro blocks have been determined to be zero, we consider it as the non-movement region, and no longer do the division. In the process of doing division of rectangular region, we must ensure that height and width are multiples of 16, otherwise, for filling .

• Sub-regional treatment. The MPEG-4 encoder that codes for each frame is based upon the macro block, so we can deal with macro block of movement region and non movement region separately. Regarding the macro block of non-movement region, we only code once for it during a period of time, and save it in the decoding site after decoding. When decoding, we call the former result of decoding directly. Certainly, in order to obtain a better effect, it is necessary to recode for the nonmovement region for a while.

3.2.4 Module of Network Transmission based on JRTPLIB

JRTPLIB is an object-oriented RTP library, follows the design of RFC 1889 completely, and supports many kinds of operating systems, such as Windows, Linux, FreeBSD, Solaris, Unix and VxWorks, etc. It adopts the socket mechanism and solves the network transmission of video streams well.

3.2.4.1 Data Sending

Figure.4 shows the flow chart of the sending data.

Firstly, we get the IP address and port number form the receiving site, then create a session instance with the class of RTP Session and call the functions of Create( ) and Set Time stamp Unit( ) to initialize the session instance. After initializing successfully, we set the goal address to send by calling the methods of RTP Session class, which are the functions of Add Destination ( ), Delete Destination ( ) and the Clear Destinations ( ). Then we set the default parameter of the RTP Session

instance by calling the functions of SetDefaultPayloadType( ), SetDefaultMark( ) and the SetDefaultTimeStampIncrement( ). At last, the stream data was sent by calling the method of SendPacket( ) in the class of RTPSession.

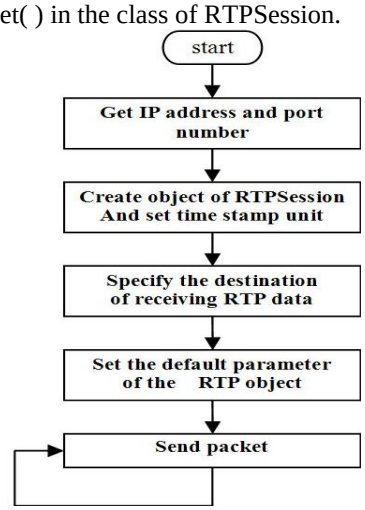

Figure 4. Flow chart of the sending data

Firstly, we get the IP address and port number form the receiving site, then create a session instance with the class of RTPSession and call the functions of Create( ) and SetTimestampUnit( ) to initialize the session instance. After initializing successfully, we set the goal address to send by calling the methods of RTPSession class,which are the functions of AddDestination( ), DeleteDestination( ) and the ClearDestinations( ). Then we set the default parameter of the RTPSession instance by calling the functions of SetDefaultPayloadType( ), SetDefaultMark( ) and the SetDefaultTimeStampIncrement( ). At last, the stream data was sent by calling the method of SendPacket( ) in the class of RTPSession.

3.2.4.2 Data Receiving

Figure.5 shows the flow chart of data receiving site.

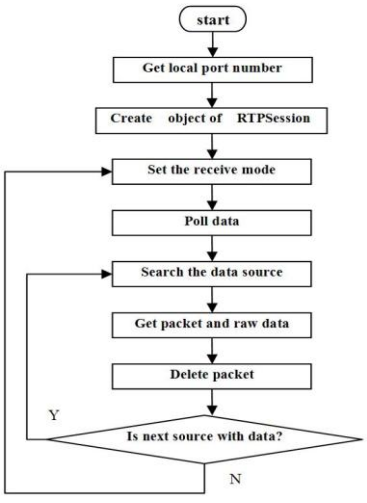

Figure 5. Flow chart of receiving data

Firstly, we obtain the port number which is set by users and create an object of RTPSession class; then we set the receiving mode by calling the methods of SetReceiveMode( ) and the AddToAcceptList( ). Secondly, we call the method of PollData( ) to receive the data packet of RTP or RTCP that has been sent.Then we can either call the methods of GotoFirstSource( ) and GotoNextSource( ) to search all sources, or call the methods of GotoFirstSourceWithDat( ) and GotoNextSourceWithData( ) to search those source that carry the data, when searching the RTP source. The reason is that it is allowed many participants (source) to exist in the same RTP session. After detecting the valid data source, we can call the method of GetNextPacket( ) to get a RTP data packet, then release it[4]

IV. Voice recognition

There are two essential characteristics for any effective ASR system: accuracy and speed. In addition, to meeting these demands, ASR systems face a number of additional challenges including the large variance that exists among individual human speech patterns (e.g. pitch, rate, inflection). A successful ASR system requires extraordinary flexibility to accommodate these variances. The process of ASR typically follows these steps:

1. Acoustic speech is captured by a microphone and undergoes analog-to-digital conversion
2. Short time frames of speech are converted into signal features that can be used to discern speech components
3. Series of frames are mapped to phonemes, the smallest building blocks of speech, and series of phonemes are mapped to words.
4. Contextual information is simultaneously processed to increase the accuracy of the ASR system. For example, contextual information would be able to assist in the selection of a series of homonyms (e.g. *merry*, *marry*, and *Mary*). The series of words are determined based on the a probabilistic measure and provided in a format suitable for the next stage in the application (e.g. text format).

Sphinx-4 is a state-of-the-art speech recognition system written entirely in the Java programming language. It was created via a joint collaboration between the Sphinx group at Carnegie Mellon University, Sun Microsystems Laboratories, Mitsubishi Electric Research Labs (MERL), and Hewlett Packard (HP), with contributions from the University of California at Santa Cruz (UCSC) and the Massachusetts Institute of Technology (MIT). Sphinx-4 started out as a port of Sphinx-3 to the Java programming language, but evolved into a recognizer designed to be much more

flexible than Sphinx-3, thus becoming an excellent platform for speech research. Live mode and batch mode speech recognizers, capable of recognizing discrete and continuous speech.

Sphinx-4 is a very flexible system capable of performing many different types of recognition tasks. As such, it is difficult to characterize the performance and accuracy of Sphinx-4 with just a few simple numbers such as speed and accuracy. Instead, we regularly run regression tests on Sphinx-4 to determine how it performs under a variety of tasks. These tasks and their latest results are as follows (each task is progressively more difficult than the previous task):

- Isolated Digits (TI46): Runs Sphinx-4 with pre-recorded test data to gather performance metrics for recognizing just one word at a time. The vocabulary is merely the spoken digits from 0 through 9, with a single utterance containing just one digit.
  (TI46 refers to the "NIST CD-ROM Version of the Texas Instruments-developed 46-Word Speaker-Dependent Isolated Word Speech Database".)
- Connected Digits (TIDIGITS): Extends the Isolated Digits test to recognize more than one word at a time (i.e., continuous speech). The vocabulary is merely the spoken digits from 0 through 9, with a single utterance containing a sequence of digits.
  (TIDIGITS refers to the "NIST CD-ROM Version of the Texas Instruments-developed Studio Quality Speaker-Independent Connected-Digit Corpus".)
- Small Vocabulary (AN4): Extends the vocabulary to approximately 100 words, with input data ranging from speaking words as well as spelling words out letter by letter.
- Medium Vocabulary (RM1): Extends the vocabulary to approximately 1,000 words.
- Medium Vocabulary (WSJ5K): Extends the vocabulary to approximately 5,000 words.
- Medium Vocabulary (WSJ20K): Extends the vocabulary to approximately 20,000 words.
- Large Vocabulary (HUB4): Extends the vocabulary to approximately 64,000 words

Sphinx-4 is an HMM-based speech recognizer. HMM stands for Hidden Markov Models, which is a type of statistical model. In HMM-based speech recognizers, each unit of sound (usually called a phoneme) is represented by a statistical model that represents the distribution of all the evidence (data) for that phoneme. This is called the acoustic model for that phoneme. When creating an acoustic model, the speech signals are first transformed into a sequence of vectors that represent certain characteristics of the signal, and the parameters of the acoustic model are then estimated using these vectors (usually called features). This process is called training the acoustic models.

During speech recognition, features are derived from the incoming speech (we will use "speech" to mean the same thing as "audio") in the same way as in the training process. The component of the recognizer that generates these features is called the front end. These live features are scored against the acoustic model. The score obtained indicates how likely that a particular set of features (extracted from live audio) belongs to the phoneme of the corresponding acoustic model.

The process of speech recognition is to find the best possible sequence of words (or units) that will fit the given input speech. It is a search problem, and in the case of HMM-based recognizers, a graph search problem. The graph represents all possible sequences of phonemes in the entire language of the task under consideration. The graph is typically composed of the HMMs of sound units concatenated in a guided manner, as specified by the grammar of the task. As an example, lets look at a simple search graph that decodes the words "one" and "two". It is composed of the HMMs of the sounds units of the words "one" and "two":

Constructing the above graph requires knowledge from various sources. It requires a dictionary, which maps the word "one" to the phonemes W, AX and N, and the word "two" to T and OO. It requires the acoustic model to obtain the HMMs for the phonemes W, AX, N, T and OO. In Sphinx-4, the task of constructing this search graph is done by the linguist.

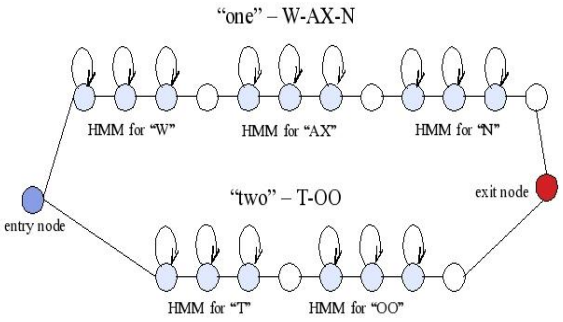

Figure 6 Search Graph

Usually, the search graph also has information about how likely certain words will occur is as shown in Figure 6. This information is supplied by the language model. Suppose that, in our example, the probability of someone saying "one" (e.g., 0.8) is much higher than saying "two" (0.2).Then, in the above graph, the probability of the transition between the entry node and the first node

of the HMM for W will be 0.8, while the probability of the transition between the entry node and the first node of the HMM for T will be 0.2. The path to "one" will consequently have a higher score.

Once this graph is constructed, the sequence of parameterized speech signals (i.e., the features) is matched against different paths through the graph to find the best fit. The best fit is usually the least cost or highest scoring path, depending on the implementation. In Sphinx-4, the task of searching through the graph for the best path is done by the search manager.

As you can see from the above graph, a lot of the nodes have self transitions. This can lead to a very large number of possible paths through the graph. As a result, finding the best possible path can take a very long time. The purpose of the pruner is to reduce the number of possible paths during the search, using heuristics like pruning away the lowest scoring paths.

As we described earlier, the input speech signal is transformed into a sequence of feature vectors. After the last feature vector is decoded, we look at all the paths that have reached the final exit node (the red node). The path with the highest score is the best fit, and a result taking all the words of that path is returned.

4.1 Sphinx-4 Architecture and Main Components:

When the recognizer starts up, it constructs the front end (which generates features from speech), the decoder, and the linguist (which generates the search graph) according to the configuration specified by the user. These components will in turn construct their own subcomponents. For example, the linguist will construct the acoustic model, the dictionary, and the language model. It will use the knowledge from these three components to construct a search graph that is appropriate for the task. The decoder will construct the search manager, which in turn constructs the scorer, the pruner, and the active list which is shown in Figure 7.

Most of these components represents interfaces. The search manager, linguist, acoustic model, dictionary, language model, active list, scorer, pruner, and search graph are all Java interfaces. There can be different implementations of these interfaces. For example, there are two different implementations of the search manager. Then, how does the system know which implementation to use? It is specified by the user via the configuration file, an XML-based file that is loaded by the configuration manager. In this configuration file, the user can also specify the properties of the implementations. One example of a property is the sample rate of the incoming speech data.

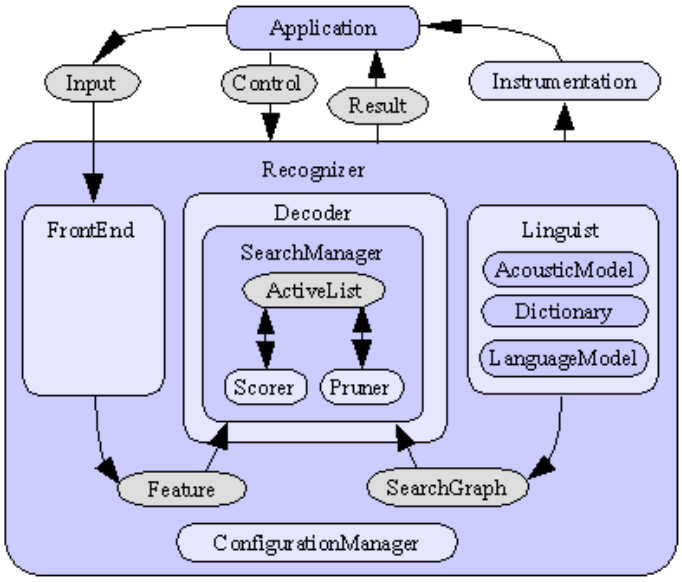

Figure 7 Sphinx-4 Architecture

The active list is a component that requires explanation. Remember we mentioned that there can be many possible paths through the search graph. Sphinx-4 currently implements a token-passing algorithm. Each time the search arrives at the next state in the graph, a token is created. A token points to the previous token, as well as the next state. The active list keeps track of all the current active paths through the search graph by storing the last token of each path. A token has the score of the path at that particular point in the search. To perform pruning, we simply prune the tokens in the active list.

When the application asks the recognizer to perform recognition, the search manager will ask the scorer to score each token in the active list against the next feature vector obtained from the front end. This gives a new score for each of the active paths. The pruner will then prune the tokens (i.e., active paths) using certain heuristics. Each surviving paths will then be expanded to the next states, where a new token will be created for each next state. The process repeats itself until no more feature vectors can be obtained from the front end for scoring. This usually means that there is no more input speech data. At that point, we look at all paths that have reached the final exit state, and return the highest scoring path as the result to the application [6]

V. Conclusion

In this paper, we present a voice recognition system for the robot control based on human voice. The accuracy reaches a level of 95% which is much higher than the ones provided by commercial systems in the market. The core acoustic modeling was achieved by using the HMM method and more words than the five commands are built in the linguist

database to improve the accuracy by reading the past recognized words in the buffer and increase or decrease of the possible occurrence of the output words. An embedded voice controlling system is designed in the paper which is based on S3C2410AL constructed on ARM9. The programs for video capturing, image compressing and data transmission are coded.

References

[1]. ad-hoc robot wireless communication by Zhigang Wang, MengChu Zhou and Nirwan Ansari
[2]. Wireless Transmission of RS232 Interface Signal Based on ZigBee by Guoxin Luo
[3]. The Study of Multi–Robot Communication of Autonomous Soccer Robots Based on C/S Mode by XiangZhongfan, WangQiang, WenHaiou
[4]. A novel embedded System of video capture and transmission applied in remote monitoring for wheelchair-bed service robots by He Qingyun, Liu Jizhong, Zhang Hua
[5]. voice based robot control by Peter X. Liu, A. D. C. Chan, R. Chen, K. Wang, Y. Zhu
[6]. A HMM Speech Recognition System Based on FPGA by Sujuan Ke, Yibin Hou, Zhangqin Huang, Hui Li

Authors

G.SITA ANNAPURNA received her master degree in Embedded Systems from JNTU Kakinada University in Gudlavalleru Engineering College, Gudlavalleru. She is an Asst Professor of ECE in Sri Vasvai Institute of Engineering and Technology, Nandamuru.

B.MAMATHA received her her master degree in Embedded Systems from JNTU Kakinada University in Pragati Engineering College, Surampalem.She is an Asst Professor of ECE in Sri Vasvai Institute of Engineering and Technology, Nandamuru.

P.ANNAPURNA received her bachelor degree in ECE from BIT Institute of Engineering & Technology. She is an Asst Professor of ECE in Sri Vasvai Institute of Engineering and Technology, Nandamuru.