

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE INC.,
Petitioner

v.

ZENTIAN LIMITED,
Patent Owner

Inter Partes Review Case No. IPR2023-00034
U.S. Patent No. 7,979,277

**DECLARATION OF CHRISTOPHER SCHMANDT
IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 7,979,277**

TABLE OF CONTENTS

I.	INTRODUCTION AND QUALIFICATIONS	15
A.	EDUCATIONAL BACKGROUND AND PROFESSIONAL EXPERIENCE	15
II.	METHODOLOGY: MATERIALS CONSIDERED	18
III.	OVERVIEW OF LEGAL STANDARDS	21
A.	PERSON OF ORDINARY SKILL IN THE ART	22
B.	OBVIOUSNESS.....	22
C.	ANALOGOUS ART	28
D.	CLAIM CONSTRUCTION	29
	1. Construction Pursuant to 35 U.S.C. § 112, ¶ 6	29
IV.	LEVEL OF A PERSON OF ORDINARY SKILL	31
V.	OVERVIEW OF THE TECHNOLOGY	32
A.	SPEECH RECOGNITION	32
B.	FEATURE VECTORS	40
C.	ACOUSTIC MODELS	49
D.	HIDDEN MARKOV MODELS	51
E.	DISTANCE CALCULATIONS	59
F.	GAUSSIAN DISTRIBUTION AND PROBABILITY	63
G.	SPEECH RECOGNITION SYSTEM HARDWARE.....	66
H.	PIPELINING	75
I.	INTERRUPTS	79
J.	PRIOR ART SPEECH RECOGNITION SYSTEMS	81
VI.	OVERVIEW OF THE '277 PATENT.....	82
VII.	SUMMARY OF UNPATENTABILITY	83
VIII.	OVERVIEW OF THE PRIOR ART	85
A.	OVERVIEW OF JIANG	85
B.	OVERVIEW OF BAUMGARTNER	85
C.	OVERVIEW OF BROWN.....	86
D.	OVERVIEW OF KAZEROONIAN	86

E.	OVERVIEW OF VENSKO	87
F.	OVERVIEW OF SMYTH	87
IX.	OPINIONS REGARDING GROUND 1: CLAIMS 1, 5, 7, 12, AND 14-16 ARE OBVIOUS OVER <i>JIANG, BAUMGARTNER, AND BROWN</i>.....	88
A.	INDEPENDENT CLAIM 1	88
1.	<i>Claim 1(Pre): “A speech recognition circuit, comprising”</i>	<i>88</i>
2.	<i>Claim 1(a): “an audio front end for calculating a feature vector from an audio signal,”</i>	<i>89</i>
3.	<i>Claim 1(b): “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”</i>	<i>94</i>
a)	The '277 Patent's Discussion of “extracted and/or derived quantities”	94
b)	<i>Jiang's Teachings</i>	<i>96</i>
4.	<i>Claim 1(c): “a calculating circuit for calculating distances indicating the similarity between a feature vector and a plurality of predetermined acoustic states of an acoustic model; and”</i>	<i>98</i>
a)	The '277 Patent's Discussion of “distances”	100
b)	<i>Jiang's Teachings</i>	<i>102</i>
(1)	“a calculating circuit”	102
(2)	“a plurality of predetermined acoustic states of an acoustic model”	104
(3)	“a feature vector”	107
(4)	“calculating distances ...”	110
c)	<i>Baumgartner Teaches a “calculating circuit”</i>	<i>121</i>
d)	Motivation to Combine <i>Jiang</i> and <i>Baumgartner's</i> Teachings	126
5.	<i>Claim 1(d): “a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words;”</i>	<i>127</i>
a)	“a lexical tree”	127

Declaration of Christopher Schmandt
Patent No. 7,979,277

b)	“the lexical tree comprising a model of words”	130
c)	“a search stage for using said calculated distances to identify words within a lexical tree”	131
6.	<i>Claim 1(e): “wherein said audio front end and said search stage are implemented using a first processor, and said calculating circuit is implemented using a second processor, and”</i>	<i>135</i>
a)	Overview of Mapping.....	135
b)	<i>Baumgartner</i> Teaches an “audio front end,” a “calculating circuit,” and a “search stage”	139
c)	<i>Baumgartner</i> Teaches an Audio Front End “implemented using a first processor”.....	146
d)	<i>Baumgartner</i> Teaches a Calculating Circuit “implemented using a second processor”	149
e)	<i>Baumgartner</i> Teaches a Search Stage “implemented using a first processor”.....	151
f)	Motivation to Combine <i>Jiang</i> and <i>Baumgartner</i>	153
7.	<i>Claim 1(f): “wherein data is pipelined from the front end to the calculating circuit to the search stage.”</i>	<i>160</i>
a)	Brown’s Teachings	162
b)	Motivation to Combine <i>Brown</i> with <i>Jiang- Baumgartner</i>	164
B.	DEPENDENT CLAIM 5	168
1.	“A speech recognition circuit as claimed in claim 1, wherein the said calculating circuit is configured to autonomously calculate distances for every acoustic state defined by the acoustic model.”	168
C.	DEPENDENT CLAIM 7	170
1.	“The speech recognition circuit of claim 1, wherein the feature vector comprises a plurality of spectral components of an audio signal for a predetermined time frame.”	170
D.	DEPENDENT CLAIM 12	171

1.	<i>“The speech recognition circuit of claim 1, wherein the audio front end is configured to input a digital audio signal.”</i>	171
E.	INDEPENDENT CLAIM 14	171
1.	<i>Claim 14(Pre): “A speech recognition circuit, comprising:”</i>	171
2.	<i>Claim 14(a): “an audio front end for calculating a feature vector from an audio signal,”</i>	171
3.	<i>Claim 14(b) “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”</i>	171
4.	<i>Claim 14(c): “calculating means for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and”</i>	172
5.	<i>Claim 14(d): “a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words;”</i>	173
6.	<i>Claim 14(e): “wherein said audio front end, said calculating means, and said search stage are connected to each other to enable pipelined data flow.”</i>	173
F.	INDEPENDENT CLAIM 15	173
1.	<i>Claim 15(Pre): “A speech recognition method, comprising:”</i>	173
2.	<i>Claim 15(a): “calculating a feature vector from an audio signal using an audio front end,”</i>	174
3.	<i>Claim 15(b): “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”</i>	174
4.	<i>Claim 15(c): “calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model using a calculating circuit; and”</i>	174
5.	<i>Claim 15(d): “using a search stage to identify words within a lexical tree using said calculated distances, the lexical tree comprising a model of words;”</i>	174

6.	<i>Claim 15(e): “wherein data is pipelined from the front end, to the calculating circuit, and to the search stage.”</i>	<i>175</i>
G.	INDEPENDENT CLAIM 16	175
1.	<i>Claim 16(Pre): “A non-transitory storage medium storing processor implementable code for controlling at least one processor to implement a speech recognition method, the code comprising:”</i>	<i>175</i>
2.	<i>Claim 16(a): “code for controlling the processor to calculate a feature vector from an audio signal,”</i>	<i>176</i>
3.	<i>Claim 16(b): “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”</i>	<i>176</i>
4.	<i>Claim 16(c): “code for controlling the processor to calculate a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and”</i>	<i>177</i>
5.	<i>Claim 16(d): “code for controlling the processor to identify words within a lexical tree using said calculated distances, the lexical tree comprising a model of words,” ...</i>	<i>177</i>
6.	<i>Claim 16(e): “wherein data is pipelined by the processor pursuant to the code from the feature calculation, to the distance calculation, and to the word identification.”</i>	<i>177</i>
X.	OPINIONS REGARDING GROUND 2: CLAIM 4 IS OBVIOUS OVER JIANG, BAUMGARTNER, BROWN, AND KAZEROONIAN	177
A.	DEPENDENT CLAIM 4	177
1.	<i>Claim 4: “A speech recognition circuit as claimed in claim 1, wherein the first processor supports multi-threaded operation, and runs the search stage and front ends as separate threads.”</i>	<i>177</i>
XI.	OPINIONS REGARDING GROUND 3: CLAIMS 9-10 ARE OBVIOUS OVER JIANG, BAUMGARTNER, BROWN, AND VENSKO	182
A.	DEPENDENT CLAIM 9	182
1.	<i>Claim 9: “The speech recognition circuit of claim 1, wherein the speech accelerator has an interrupt signal to</i>	

	<i>inform the front end that the accelerator is ready to receive a next feature vector from the front end.</i>	182
B.	DEPENDENT CLAIM 10	188
1.	<i>Claim 10: “The speech recognition circuit of claim 1, wherein the accelerator signals to the search stage when the distances for a new frame are available in a result memory.”</i>	188
XII. OPINIONS REGARDING GROUND 4: CLAIMS 1, 5, 7, 12, AND 14-16 ARE OBVIOUS OVER JIANG, BAUMGARTNER, BROWN, AND SMYTH		
A.	INDEPENDENT CLAIM 1	189
1.	<i>Claim 1(Pre): “A speech recognition circuit, comprising:”</i>	189
2.	<i>Claim 1(a): “an audio front end for calculating a feature vector from an audio signal,”</i>	189
3.	<i>Claim 1(b) “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”</i>	189
4.	<i>Claim 1(c): “a calculating circuit for calculating distances indicating the similarity between a feature vector and a plurality of predetermined acoustic states of an acoustic model; and”</i>	189
a)	“a calculating circuit”	190
b)	“a plurality of predetermined acoustic states”	192
c)	“[a plurality of predetermined acoustic states] of an acoustic model”	196
d)	“for calculating distances ...”	200
e)	Motivation to Modify Jiang-Baumgartner-Brown with Smyth	203
5.	<i>Claim 1(d): “a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words;”</i>	206
6.	<i>Claim 1(e): “wherein said audio front end and said search stage are implemented using a first processor,</i>	

Declaration of Christopher Schmandt
Patent No. 7,979,277

	<i>and said calculating circuit is implemented using a second processor, and”</i>	206
7.	<i>Claim 1(f): “wherein data is pipelined from the front end to the calculating circuit to the search stage.”</i>	206
B.	DEPENDENT CLAIM 5	207
1.	<i>Claim 5: “A speech recognition circuit as claimed in claim 1, wherein the said calculating circuit is configured to autonomously calculate distances for every acoustic state defined by the acoustic model.”</i>	207
C.	DEPENDENT CLAIM 7	207
1.	<i>Claim 7: “The speech recognition circuit of claim 1, wherein the feature vector comprises a plurality of spectral components of an audio signal for a predetermined time frame.”</i>	207
D.	DEPENDENT CLAIM 12	207
1.	<i>Claim 12: “The speech recognition circuit of claim 1, wherein the audio front end is configured to input a digital audio signal.”</i>	207
E.	INDEPENDENT CLAIM 14	207
1.	<i>Claim 14(Pre): “a speech recognition circuit, comprising:”</i>	207
2.	<i>Claim 14(a): “an audio front end for calculating a feature vector from an audio signal,”</i>	207
3.	<i>Claim 14(b) “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”</i>	207
4.	<i>Claim 14(c): “calculating means for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and”</i>	208
5.	<i>Claim 14(d): “a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words;”</i>	208

Declaration of Christopher Schmandt
Patent No. 7,979,277

6.	<i>Claim 14(e): “wherein said audio front end, said calculating means, and said search stage are connected to each other to enable pipelined data flow.”</i>	208
F.	INDEPENDENT CLAIM 15	208
1.	<i>Claim 15(Pre): “A speech recognition method, comprising:”</i>	208
2.	<i>Claim 15(a): “calculating a feature vector from an audio signal using an audio front end,”</i>	208
3.	<i>Claim 15(b): “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”</i>	209
4.	<i>Claim 15(c): “calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model using a calculating circuit; and”</i>	209
5.	<i>Claim 15(d): “using a search stage to identify words within a lexical tree using said calculated distances, the lexical tree comprising a model of words;”</i>	209
6.	<i>Claim 15(e): “wherein data is pipelined from the front end, to the calculating circuit, and to the search stage.”</i>	209
G.	INDEPENDENT CLAIM 16	209
1.	<i>Claim 16(Pre): “A non-transitory storage medium storing processor implementable code for controlling at least one processor to implement a speech recognition method, the code comprising:”</i>	209
2.	<i>Claim 16(a): “code for controlling the processor to calculate a feature vector from an audio signal,”</i>	209
3.	<i>Claim 16(b): “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”</i>	210
4.	<i>Claim 16(c): “code for controlling the processor to calculate a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and”</i>	210

5. *Claim 16(d): “code for controlling the processor to identify words within a lexical tree using said calculated distances, the lexical tree comprising a model of words,” ...* 210
6. *Claim 16(e): “wherein data is pipelined by the processor pursuant to the code from the feature calculation, to the distance calculation, and to the word identification.”* 210

XIII. OPINIONS REGARDING GROUND 5: CLAIM 4 IS OBVIOUS OVER JIANG, BAUMGARTNER, BROWN, SMYTH, AND KAZEROONIAN 210

- A. DEPENDENT CLAIM 4 210
 1. *Claim 4: “A speech recognition circuit as claimed in claim 1, wherein the first processor supports multi-threaded operation, and runs the search stage and front ends as separate threads”* 210

XIV. OPINIONS REGARDING GROUND 6: CLAIMS 9-10 ARE OBVIOUS OVER JIANG, BAUMGARTNER, BROWN, SMYTH, AND VENSKO 211

- A. DEPENDENT CLAIM 9 211
 1. *Claim 9: “The speech recognition circuit of claim 1, wherein the speech accelerator has an interrupt signal to inform the front end that the accelerator is ready to receive a next feature vector from the front end.”* 211
- B. DEPENDENT CLAIM 10 211
 1. *Claim 10: “The speech recognition circuit of claim 1, wherein the accelerator signals to the search stage when the distances for a new frame are available in a result memory.”* 211

XV. CONCLUSION 212

CLAIM LISTING

Claim 1:

1(Pre) A speech recognition circuit, comprising:

1(a) an audio front end for calculating a feature vector from an audio signal,

1(b) wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;

1(c) a calculating circuit for calculating distances indicating the similarity between a feature vector and a plurality of predetermined acoustic states of an acoustic model; and;

1(d) a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words;

1(e) wherein said audio front end and said search stage are implemented using a first processor, and said calculating circuit is implemented using a second processor, and

1(f) wherein data is pipelined from the front end to the calculating circuit to the search stage.

Claim 4:

A speech recognition circuit as claimed in claim 1, wherein the first processor supports multi-threaded operation, and runs the search stage and front ends as separate threads.

Claim 5:

A speech recognition circuit as claimed in claim 1, wherein the said calculating circuit is configured to autonomously calculate distances for every acoustic state defined by the acoustic model.

Claim 7:

The speech recognition circuit of claim 1, wherein the feature vector comprises a plurality of spectral components of an audio signal for a predetermined time frame.

Claim 9:

The speech recognition circuit of claim 1, wherein the speech accelerator has an interrupt signal to inform the front end that the accelerator is ready to receive a next feature vector from the front end.

Claim 10:

The speech recognition circuit of claim 1, wherein the accelerator signals to the search stage when the distances for a new frame are available in a result memory.

Claim 12:

The speech recognition circuit of claim 1, wherein the audio front end is configured to input a digital audio signal.

Claim 14:

14(Pre) A speech recognition circuit, comprising:

14(a) an audio front end for calculating a feature vector from an audio signal,

14(b) wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;

14(c) calculating means for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model;
and

14(d) a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words;

14(e) wherein said audio front end, said calculating means, and said search stage are connected to each other to enable pipelined data flow.

Claim 15:

15(Pre) A speech recognition method, comprising:

15(a) calculating a feature vector from an audio signal using an audio front end,

15(b) wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;

15(c) calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model using a calculating circuit;
and

15(d) using a search stage to identify words within a lexical tree using said calculated distances, the lexical tree comprising a model of words;

15(e) wherein data is pipelined from the front end, to the calculating circuit, and to the search stage.

Claim 16:

16(Pre) A non-transitory storage medium storing processor implementable code for controlling at least one processor to implement a speech recognition method, the code comprising:

16(a) code for controlling the processor to calculate a feature vector from an audio signal,

16(b) wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;

16(c) code for controlling the processor to calculate a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and

16(d) code for controlling the processor to identify words within a lexical tree using said calculated distances, the lexical tree comprising a model of words,

16(e) wherein data is pipelined by the processor pursuant to the code from the feature calculation, to the distance calculation, and to the word identification.

I, Christopher Schmandt, declare as follows:

I. INTRODUCTION AND QUALIFICATIONS

1. I am over the age of 21 and am competent to make this declaration.

A. Educational Background and Professional Experience

2. I retired several years ago after a 40-year career at the Massachusetts Institute of Technology (“MIT”); for most of that time I was employed as a Principal Research Scientist at the Media Laboratory. In that role I also served as faculty for the MIT Media Arts and Sciences academic program. I was a founder of the Media Laboratory, a research lab which now spans two buildings.

3. I received my B.S. degree in Electrical Engineering and Computer Science from MIT in 1978, and my M.S. in Visual Studies (Computer Graphics) also from MIT. I was employed at MIT since 1980, initially at the Architecture Machine Group which was an early computer graphics and interactive systems research lab. In 1985, I helped found the Media Laboratory and continued to work there until retirement. I was director of a research group titled “Living Mobile.” My research spanned distributed communication and collaborative systems, with an emphasis on multi-media and user interfaces, with a strong focus on speech-based systems. I have over 70 published conference and journal papers and one book in the field of speech technology and user interaction.

4. For the first fifteen years of my career, my research emphasized speech recognition and speech user interfaces. I built the first conversational computer system utilizing speech recognition and synthesis (“Put That There”) starting in 1980. I continued to innovate speech user interfaces using recognition, text-to-speech synthesis, and recorded audio in a wide variety of projects. I built one of the first graphical user interfaces for audio editing, employing keyword recognition on voice memos in 1982 (Intelligent Ear). I built the first research-grade unified messaging system, which combined text and voice messages into a single inbox, with speech recognition over the phone for remote access, and a graphical user interface for desktop access in 1983 (Phone Slave). Along with my students we built the first system for real time spoken driving directions, including speech-accessible maps of Cambridge, Massachusetts in 1987 (Back Seat Driver). We built some of the earliest speech-based personal assistants for managing messages, calendar, contacts, etc. (Conversational Desktop 1985, Chatter 1993, MailCall 1996). We built quite a few systems employing speech recognition in handheld mobile devices (ComMotion 1999, Nomadic Radio 2000, Impromptu 2001, and Symphony 2004, for example). We applied speech recognition to large bodies of everyday conversations captured with a wearable device and utilized as a memory aid (Memory Prosthesis 2004). We used speech recognition on radio newscasts to build a personalized version of audio newscasts (Synthetic News Radio, 1999) and also

investigated adding speech recognition to a mouse-based window system a few years earlier.

5. I was later awarded the prestigious Association for Computing Machinery (ACM) Computer Human Interface (CHI) Academy membership specifically for those years of work pioneering speech user interfaces.

6. In the course of my research, I built a number of speech recognition client/server distributed systems, with the first being in 1985. Much of the initial motivation for a server architecture was that speech recognition required expensive digital signal processing hardware that we could not afford to put on each computer, so a central server with the required hardware was used. Later versions of the speech recognition server architecture allowed certain computers to perform specialized tasks serving a number of client computers providing voice user interfaces, either on screens or over telephone connections.

7. Because of my early work with distributed speech systems, I served for several years in the mid-1990s with a working group on the impact of multimedia systems on the Internet reporting to the Internet Engineering Task Force (IETF) and later the Internet Activities Board (IAB). This work impacted emerging standards such as Session Initiation Protocol (SIP).

8. In my faculty position I taught graduate level courses in speech technology and user interaction design, and directly supervised student research and

theses at the Bachelors, Masters, and PhD level. I oversaw the Masters and PhD thesis programs for the entire Media Arts and Sciences academic program during my more senior years. I also served on the Media Laboratory intellectual property committee for many years.

II. METHODOLOGY: MATERIALS CONSIDERED

9. I have relied upon my education, knowledge and experience with speech technology and speech recognition systems, as well as the other materials as discussed in this declaration in forming my opinions.

10. For this work, I have been asked to review U.S. Patent No. 7,979,277 (“the ‘277 Patent”) (Ex. 1001) including the specification and claims, and the ‘277 Patent’s prosecution history (“‘277 File History”) (Ex. 1002). In developing my opinions relating to the ‘277 Patent, I have considered the materials cited or discussed herein, including those itemized in the Exhibit Table below.

<u>Exhibit</u>	<u>Description</u>
Exhibit 1001	U.S. Patent No. 7,979,277 (“‘277 Patent”)
Exhibit 1002	Prosecution History for the ‘277 Patent (“‘277 File History”)
Exhibit 1004	U.S. Patent No. 6,374,219 to Jiang (“Jiang”)
Exhibit 1005	U.S. Patent No. 5,819,222 to Smyth et al. (“Smyth”)
Exhibit 1006	U.S. Patent No. 6,959,376 to Boike et al. (“Boike”)
Exhibit 1007	U.S. Patent Application Publication No. 2002/0049582 to Baumgartner et al. (“Baumgartner”)
Exhibit 1008	Harry Newton, <i>Newton’s Telecom Dictionary</i> , 19 th Edition, CMP Books, March 2003 (“Newton”)
Exhibit 1009	Kevin L. Kloker, <i>The Motorola DSP56000 Digital Signal Processor</i> , IEEE Micro. Vol. 6, Issue 6, pp. 29-48, December 1986 (“Kloker”)

Declaration of Christopher Schmandt
Patent No. 7,979,277

Exhibit 1010	Christopher Schmandt, <i>Voice Communication with Computers</i> , Van Nostrand Reinhold, 1994 (“ <i>Schmandt</i> ”)
Exhibit 1011	U.S. Patent No. 6,499,012 to Peters et al. (“ <i>Peters</i> ”)
Exhibit 1012	Ph.D. Thesis of Mosur Ravishankar (“ <i>Ravishankar</i> ”)
Exhibit 1015	Lawrence Rabiner, <i>Fundamentals of Speech Recognition</i> , Pearson Education, 1993 (“ <i>Rabiner</i> ”)
Exhibit 1016	Richard L. Klevans, <i>Voice Recognition</i> , Artech House Inc. 1997 (“ <i>Klevans</i> ”)
Exhibit 1017	U.S. Patent No. 7,120,582 to Young et al. (“ <i>Young</i> ”)
Exhibit 1018	John Holmes, <i>Speech Synthesis and Recognition</i> , 2 nd Edition, Taylor & Francis, 2001 (“ <i>Holmes</i> ”)
Exhibit 1019	U.S. Patent No. 4,926,488 to Nadas et al. (“ <i>Nadas</i> ”)
Exhibit 1020	Frederick Jelinek, <i>Statistical Methods for Speech Recognition</i> , The MIT Press, 1997 (“ <i>Jelinek</i> ”)
Exhibit 1021	Shigeru Katagiri, <i>Handbook of Neural Networks for Speech Processing</i> , Artech House Inc., 2000 (“ <i>Katagiri</i> ”)
Exhibit 1022	Lawrence R. Rabiner, <i>A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition</i> , Proceedings of the IEEE, VOL. 77, NO 2, pp. 257-286, February 1989 (“ <i>Rabiner89</i> ”)
Exhibit 1023	U.S. Patent No. 6,182,037 to Maes (“ <i>Maes</i> ”)
Exhibit 1024	U.S. Patent Application Publication No. 2003/0220796 to Aoyama et al. (“ <i>Aoyama</i> ”)
Exhibit 1025	U.S. Patent No. 5,895,447 to Ittycheriah et al. (“ <i>Ittycheriah '447</i> ”)
Exhibit 1026	U.S. Patent No. 6,092,045 to Stubbley et al. (“ <i>Stubbley</i> ”)
Exhibit 1027	U.S. Patent No. 6,151,574 to Lee et al. (“ <i>Lee</i> ”)
Exhibit 1028	U.S. Patent No. 6,580,814 to Ittycheriah et al. (“ <i>Ittycheriah</i> ”)
Exhibit 1029	U.S. Patent No. 6,452,348 to Toyoda (“ <i>Toyoda</i> ”)
Exhibit 1030	U.S. Patent No. 6,195,634 to Dudemaine et al. (“ <i>Dudemaine</i> ”)
Exhibit 1031	U.S. Patent No. 5,893,058 to Kosaka (“ <i>Kosaka</i> ”)
Exhibit 1032	U.S. Patent Application Publication No. 2001/0010039 to Yang (“ <i>Yang</i> ”)
Exhibit 1033	U.S. Patent No. 5,754,878 to Asghar et al. (“ <i>Asghar</i> ”)
Exhibit 1034	U.S. Patent No. 6,041,400 to Ozcelik et al. (“ <i>Ozcelik</i> ”)
Exhibit 1035	Declaration of June Ann Munford (“ <i>Munford</i> ”)
Exhibit 1036	U.S. Patent No. 5,699,456 to Brown et al. (“ <i>Brown</i> ”)

Declaration of Christopher Schmandt
Patent No. 7,979,277

Exhibit 1037	International Publication No. WO 99/41684 to Kazeroonian et al. (“ <i>Kazeroonian</i> ”)
Exhibit 1038	U.S. Patent No. 4,567,606 to Vensko et al. (“ <i>Vensko</i> ”)
Exhibit 1039	U.S. Patent No. 6,182,203 to Simar, Jr. et al. (“ <i>Simar</i> ”)
Exhibit 1047	U.S. Patent No. 6,879,954 to Nguyen et al. (“ <i>Nguyen</i> ”)
Exhibit 1050	Binu K. Mathew et al., <i>A Gaussian Probability Accelerator for SPHINX 3</i> (“ <i>Mathew I</i> ”)
Exhibit 1051	Binu K. Mathew et al., <i>A Low-Power Accelerator for the SPHINX 3 Speech Recognition System</i> (“ <i>Mathew II</i> ”)
Exhibit 1052	U.S. Patent No. 4,718,094 to Bahl et al. (“ <i>Bahl</i> ”)
Exhibit 1053	Hermann Ney et al. <i>Dynamic Programming Search Strategies: From Digit Strings to Large Vocabulary Word Graphs</i> C.-H. Lee et al. (eds.), <i>Automatic Speech and Speaker Recognition</i> (“ <i>Ney</i> ”)
Exhibit 1054	U.S. Patent No. 7,058,678 to Luick et al. (“ <i>Luick</i> ”)
Exhibit 1055	<i>Dictionary of Electrical and Computer Engineering</i> , 6 th Edition, McGraw-Hill. 2003 (“ <i>Dictionary of Electrical and Computer Engineering</i> ”)
Exhibit 1056	<i>Microsoft Computer Dictionary</i> , 5 th Edition, Microsoft Press, 2002 (“ <i>Microsoft Computer Dictionary</i> ”)
Exhibit 1058	Abdulmesih Aktas et al., <i>Multi-DSP and VQ-ASIC Based Acoustic Front-End for Real-Time Speech Processing Tasks</i> , First European Conference on Speech Communication and Technology EUROSPEECH '89, Paris, France, September 1989 (<i>Atkas 1989</i>)
Exhibit 1059	Abdulmesih Aktas et al., <i>Speaker Independent Continuous Hmm-Based Recognition of Isolated Words on A Real-Time Multi-Dsp System</i> , Second European Conference on Speech Communication and Technology EUROSPEECH '91, Genova, Italy, September 1991 (<i>Atkas 1991</i>)
Exhibit 1060	U.S. Patent No. 6,029,221 to Wu et al. (“ <i>Wu</i> ”)

11. I have considered these materials from the viewpoint of a POSITA as of the priority date of the '277 Patent. For the purposes of this declaration, I have been asked to assume that the priority date of the '277 Patent is September 14, 2004.

I note that my opinions provided in this Declaration are made from the perspective

of a POSITA as of this priority date of the '277 Patent, unless expressly stated otherwise. To the extent that I use any verb tense in this Declaration that is present tense (e.g., “a POSITA would understand” instead of “a POSITA would have understood”), such verb tense should be understood to be my opinion as of the '277 Patent’s priority date (again, unless expressly stated otherwise). I merely use the present verb tense for ease of reading.

III. OVERVIEW OF LEGAL STANDARDS

12. In formulating my opinions, I have been instructed to apply certain legal standards. I am not a lawyer. I do not expect to offer any testimony regarding what the law is. Instead, the following sections summarize the law as I have been instructed to apply it in formulating and rendering my opinions found later in this declaration. I understand that, in an *inter partes* review (“IPR”) proceeding, patent claims may be deemed unpatentable if it is shown that they were anticipated or rendered obvious in view of the prior art. I understand that prior art in an IPR is limited to patents or printed publications that predate the priority date of the patent at issue. I understand that questions of claim clarity (definiteness) and enablement cannot be considered as a ground for considering the patentability of a claim in these proceedings.

A. Person of Ordinary Skill in the Art

13. I understand that the '277 Patent, the record of proceedings at the Patent Office (which I understand is called the "File History" or "Prosecution History"), and the teachings of the prior art are evaluated from the perspective of a person of ordinary skill in the art ("POSITA"). I understand that the factors considered in determining the ordinary level of skill in the art include: (i) the levels of education and experience of persons working in the field; (ii) the types of problems encountered in the field; and (iii) the sophistication of the technology. I may also consider, if available, the education level of the inventor, prior art solutions to the problems encountered in the art, and the rapidity with which innovations are made in the relevant art.

14. I understand that a person of ordinary skill in the art is not a specific real individual, but rather a hypothetical individual having the qualities reflected by the factors above. This hypothetical person has knowledge of all prior art in the relevant field as if it were arranged on a workshop wall and takes from each reference what it would teach to a person having the skills of a POSITA.

B. Obviousness

15. I understand that a claim may be invalid under § 103(a) if the subject matter described by the claim as a whole would have been "obvious" to a hypothetical POSITA in view of a single prior art reference or in view of a

combination of references at the time the claimed invention was made. Therefore, I understand that obviousness is determined from the perspective of a hypothetical POSITA. I further understand that a hypothetical POSITA is assumed to know and to have all relevant prior art in the field of endeavor covered by the patent in suit and all analogous prior art. I understand that obviousness in an IPR review proceeding is evaluated using a preponderance of the evidence standard, which means that the claims must be more likely obvious than nonobvious.

16. I also understand that an analysis of whether a claimed invention would have been obvious should be considered in light of the scope and content of the prior art, the differences (if any) between the prior art and the claimed invention, and the level of ordinary skill in the pertinent art involved. I understand as well that a prior art reference should be viewed as a whole. I understand that in considering whether an invention for a claimed combination would have been obvious, I may assess whether there are apparent reasons to combine known elements in the prior art in the manner claimed in view of interrelated teachings of multiple prior art references, the effects of demands known to the design community or present in the marketplace, and/or the background knowledge possessed by a POSITA. I also understand that other principles may be relied on in evaluating whether a claimed invention would have been obvious, and that these principles include the following:

Declaration of Christopher Schmandt
Patent No. 7,979,277

- A combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results;
- When a device or technology is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or in a different one, so that if a POSITA can implement a predictable variation, the variation is likely obvious;
- If a technique has been used to improve one device, and a POSITA would have recognized that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill;
- An explicit or implicit teaching, suggestion, or motivation to combine two prior art references to form the claimed combination may demonstrate obviousness, but proof of obviousness does not depend on or require showing a teaching, suggestion, or motivation to combine;
- Market demand, rather than scientific literature, can drive design trends and may show obviousness;

Declaration of Christopher Schmandt
Patent No. 7,979,277

- In determining whether the subject matter of a patent claim would have been obvious, neither the particular motivation nor the avowed purpose of the named inventor controls;
- One of the ways in which a patent's subject can be proved obvious is by noting that there existed at the time of invention a known problem for which there was an obvious solution encompassed by the patent's claims;
- Any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed;
- "Common sense" teaches that familiar items may have obvious uses beyond their primary purposes, and in many cases a POSITA will be able to fit the teachings of multiple patents together like pieces of a puzzle;
- A POSITA is also a person of ordinary creativity, and is not an automaton;
- A patent claim can be proved obvious by showing that the claimed combination of elements was "obvious to try," particularly when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions such

that a POSITA would have had good reason to pursue the known options within his or her technical grasp; and

- One should not use hindsight in evaluating whether a claimed invention would have been obvious.

17. I also understand that an obviousness determination includes the consideration of various factors such as: (1) the scope and content of the prior art, (2) the differences between the prior art and the asserted claims, (3) the level of ordinary skill in the pertinent art, and (4) the existence of secondary considerations such as commercial success, long-felt but unresolved needs, failure of others, and so forth.

18. I am informed that it is improper to combine references where the references teach away from their combination. I am informed that a reference may be said to teach away when a person of ordinary skill in the relevant art, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the patent applicant. In general, a reference will teach away if it suggests that the line of development flowing from the reference's disclosure is unlikely to be productive of the result sought by the patentee. I am informed that a reference teaches away, for example, if (1) the combination would produce a seemingly inoperative device, or (2) the references leave the impression that the product would not have the property

sought by the patentee. I also am informed, however, that a reference does not teach away if it merely expresses a general preference for an alternative invention but does not criticize, discredit, or otherwise discourage investigation into the invention claimed.

19. I am informed that even if a *prima facie* case of obviousness is established, the final determination of obviousness must also consider “secondary considerations” if presented. In most instances, the patentee raises these secondary considerations of non-obviousness. In that context, the patentee argues an invention would not have been obvious in view of these considerations, which include: (a) commercial success of a product due to the merits of the claimed invention; (b) a long-felt, but unsatisfied need for the invention; (c) failure of others to find the solution provided by the claimed invention; (d) deliberate copying of the invention by others; (e) unexpected results achieved by the invention; (f) praise of the invention by others skilled in the art; (g) lack of independent simultaneous invention within a comparatively short space of time; (h) teaching away from the invention in the prior art.

20. I am further informed that secondary considerations evidence is only relevant if the offering party establishes a connection, or nexus, between the evidence and the claimed invention. The nexus cannot be based on prior art features. The establishment of a nexus is a question of fact. While I understand that the Patent

Owner here has not offered any secondary considerations at this time, I will supplement my opinions in the event that the Patent Owner raises secondary considerations during the course of this proceeding.

21. Finally, I understand that obviousness in an IPR must be proven by a preponderance of the evidence.

C. Analogous Art

22. I have been informed that for a prior art reference to be proper for use in an obviousness analysis, the reference must be “analogous art” to the claimed invention. I have been informed that a reference is analogous art to the claimed invention if: (1) the reference is from the same field of endeavor as the claimed invention (even if it addresses a different problem); or (2) the reference is reasonably pertinent to the problem faced by the inventor (even if it is not in the same field of endeavor as the claimed invention). In order for a reference to be “reasonably pertinent” to the problem, it must logically have commended itself to an inventor’s attention in considering his problem. In determining whether a reference is reasonably pertinent, one should consider the problem faced by the inventor, as reflected either explicitly or implicitly, in the specification. I believe that all of the references that my opinions in this IPR are based upon are well within the range of references a person having ordinary skill in the art would consult to address the type of problems described in the Challenged Claims.

D. Claim Construction

23. I understand that the claim terms in this IPR will be construed according to their ordinary and customary meaning as understood in light of the claim language, the patent's description, and the prosecution history viewed from the perspective of a POSITA. I further understand that where a patent defines claim language, the definition in the patent controls, even if there are other definitions that might be understood by those working in the art.

1. Construction Pursuant to 35 U.S.C. § 112, ¶ 6

24. I have been informed that a claim of the '277 Patent includes a claim term written in means-plus-function format. I have been instructed to adopt the construction set forth below.

<i>Claim Term</i>	<i>Support</i>	<i>Structure and Function</i>
Claims 14: “ calculating means for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model ”	<i>'277 Patent, 37:66-67</i> (stating “calculation means may comprise the distance calculation engine”), 24:6-16 (referring to distance calculation engine as a coprocessor), Fig. 17 (illustrating distance calculation engine 204 as part of distance accelerator subsystem)	<u>Structure:</u> a processor <u>Function:</u> calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model

25. I have been informed by counsel that “calculating means” should be construed as a means-plus-function limitation. I have further been requested by counsel to provide an opinion as to what structure is described in the ’277 Patent corresponding to the claimed “calculating means,” including any equivalents. In my opinion, the calculating means is met by at least a processor.

26. I first note that the ’277 Patent does not recite the term “calculating means,” except in the claims. I note the ’277 Patent recites “[i]n the following claims, the calculation means may comprise the distance calculation engine described above.” *’277 Patent*, 37:66-67. Of course, calculating distances is discussed repeatedly throughout the ’277 Patent. The ’277 Patent describes an accelerator (*see* Fig. 17) that includes a “distance calculation engine.” The distance calculation engine is described as “convert[ing] feature vectors into Mahalanobis distances,” which is a well-known distance measurement. *’277 Patent*, 23:57-58. The MHD calculations are used “to determined distance values.” *’277 Patent*, 24:1-5. The ’277 Patent describes other distances being used other than MHD. *Id.* I discuss the distance calculation engine because it is the component described in the ’277 Patent as “calculating a distance” as claimed. *’277 Patent*, 25:32-33, 25:2-6, 25:18-20. That is, the calculating means is the distance calculation engine.

27. The distance calculation engine is described as being “designed as a speech accelerator, to operate as a loosely bound co-processor for a CPU running

speech recognition software.” ’277 Patent, 24:6-9. Therefore, in my opinion, the structure for performing the claimed functionality is at least a processor.

IV. LEVEL OF A PERSON OF ORDINARY SKILL

28. Based on my review and analysis of the ’277 Patent, the prior art cited herein, and the ordinary skill factors described in this section, a POSITA in the field of the ’277 Patent at the time of the earliest possible priority date (September 14, 2004) would have been knowledgeable regarding the field of speech recognition technology. In my experience working in this field, most workers of ordinary skill in the art as of the earliest possible priority date of September 14, 2004 would have had a master’s degree in computer engineering, computer science, electrical engineering, or a related field, with at least two years of experience in the field of speech recognition, or a bachelor’s degree in the same fields with at least four years of experience in the field of speech recognition. Additional education or experience might substitute for the above requirements. *See, e.g., ’277 Patent* (Ex. 1001), 1:4-2:2 (describing the “Field of the Invention” and providing a “Description of the Related Art”). A person with less relevant practical experience but with additional education can also qualify as a POSITA in the field of the ’277 Patent provided the additional education focused on speech recognition technology. When I refer to the understanding of a POSITA, I am referring to the understanding of such a person as of September 14, 2004.

29. As of September 14, 2004, I had more than ordinary skill in the art. I am, however, familiar with the skills and knowledge possessed by those I would have considered to be of ordinary skill in the art as of that date. For example, I knew people working in the field while I built systems employing speech recognition in handheld mobile devices (ComMotion 1999, Nomadic Radio 2000, Impromptu 2001, and Symphony 2004, for example).

30. My opinions provided in this declaration would not change in view of minor modifications to this level of ordinary skill.

V. OVERVIEW OF THE TECHNOLOGY

31. I was asked to briefly summarize the background of the technology from the standpoint of a POSITA prior to September 14, 2004.

A. Speech Recognition

32. Speech recognition using computer systems was known and widely used well before the priority date of the '277 Patent. Some of the earliest speech recognition systems were developed in the 1950s and 1960s. *Rabiner* (Ex. 1015), 37-38. By the 1980s, speech recognition systems using Hidden Markov Models were well known and “widely applied in virtually every speech-recognition research laboratory in the world.” *Rabiner*, 39. In general, speech recognition refers to “the technologies used to allow computers to recognize the words in human speech.” *Schmandt* (Ex. 1010), 45. Indeed, speech was known to be used in “a variety of

computing environments based on appreciating its role in human communication. Speech can be used as a method of *interacting* with a computer to place requests or receive warnings and notices. Voice can also be used as the underlying *data* itself[.]” *Schmandt*, 11. Further, at least as early as 1994 (publication date of *Schmandt*), “[d]esktop workstations can already support both these speech functions.” *Id.*

33. One of the initial processes in a speech recognition system is “parameter extraction,” which entails “methods of extracting information from speech waveforms. Parameter or feature extraction consists of preprocessing an electrical signal to transform it into a usable digital form, applying algorithms to extract only speaker-related information from the signal[.]” *Klevans* (Ex. 1016), 12. Such extraction involves “digital signal processing (DSP) methods that are common to all computer speech systems.” *Klevans*, 12-13. *Klevans* teaches that “the sound wave created by an individual’s speech is transduced into an analog electrical signal via a microphone. The electrical signal is sampled and quantized, resulting in a digital representation of the analog signal.” *Klevans*, 13. Following such analog to digital processing, “conditioning” may be performed on the signal, which includes bandpass filtering to remove additive noise and spectral flattening to compensate for “the effect of the vocal tract on the excitation signal created by the vibrating vocal folds.” *Id.*

34. In general, it was known that speech recognizer systems included three basic components: (1) “speech **representation** that is computationally efficient for pattern matching” such as “Hidden Markov Models” that “are a statistical representation based on the various ways words or phonemes may be pronounced.” *Schmandt*, 45-46. I discuss Hidden Markov Models in more detail at Section V.D below. (2) A “set of **templates** or **models**, which are descriptions of each word to be recognized, in the representation of speech used by the recognizer. The templates describe the words in the recognizer’s vocabulary, i.e., those words that the recognizer can identify. They are reference models against which an input can be compared to determine what was spoken.” *Schmandt*, 46. (3) A “**pattern matching** algorithm to determine which template is most similar to a specimen of speech input. This element of the speech recognizer must determine word boundaries, locate the most similar template, and decide whether the difference between the input and the selected template is minor enough to accept the word.” *Id.* In general, a speech recognizer captures incoming speech and converts it “to the chosen internal representation (see Figure 7.1). The pattern matcher then selects the template that most closely matches the input or rejects the utterance if no template is a close enough match.” *Id.* The functional elements of an exemplary speech recognizer are depicted below in Figure 7.1.

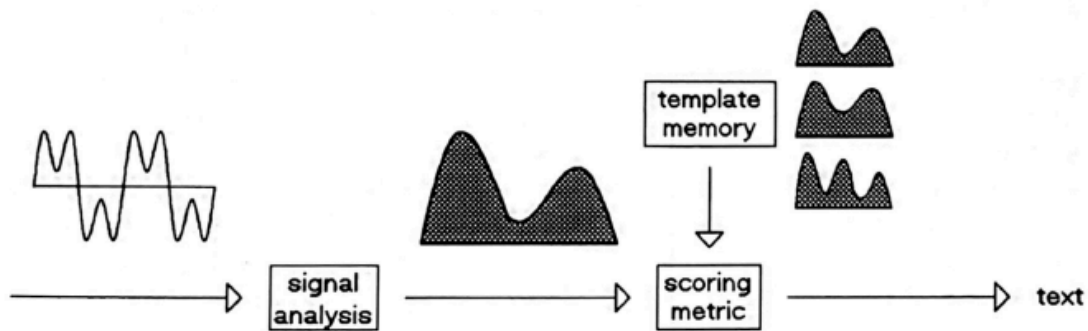


Figure 7.1. The functional elements of a speech recognizer. The user's speech is digitized and converted to the recognizer's internal representation. The captured speech is then compared with the words stored in the recognizer's template memory. The pattern matching algorithm then determines which template, if any, is the closest match.

Schmandt, 46 (“Figure 7.1”).

35. When discussing speech recognition, one important topic is phonemes, which are speech sounds that form the words in a language.

The **phonemic** layer is concerned with the string of phonemes of which the words are composed. Phonemes are the speech sounds that form the words of any language. Phonemes include all the sounds associated with vowels and consonants. A grunt, growl, hiss, or gargling sound is not a phoneme in English, so it cannot be part of a word; such sounds are not referred to as speech. At the phoneme layer, while talking we are either continuously producing speech sounds or are silent. We are not silent at word boundaries; the phonemes all run together.

Schmandt, 19. *Schmandt* further discloses that “[p]honemes are important small units, several of which make up most syllables.” *Schmandt*, 23. Further, *Schmandt*

Declaration of Christopher Schmandt
Patent No. 7,979,277

discloses that a “**phoneme** is a unit of speech, the set of which defines all the sounds from which words can be constructed in a particular language. There is at least one pair of words in a language for which replacing one phoneme with another will change what is spoken into a different word. Of course, not every combination of phonemes results in a word; many combinations are nonsense.” *Schmandt*, 23-24.

36. It was also known to utilize spectrograms for “characterizing the speech signal and representing the information associated with the sounds” “via a spectral representation.” *Rabiner*, 50. *Rabiner* teaches that “[p]erhaps the most popular representation of this type is the sound spectrogram in which a three-dimensional representation of the speech intensity, in different frequency bands, over time is portrayed.” *Id.* *Rabiner* teaches that the “spectral intensity at each point in time is indicated by the intensity (darkness) of the plot at a particular analysis frequency.” *Id.*

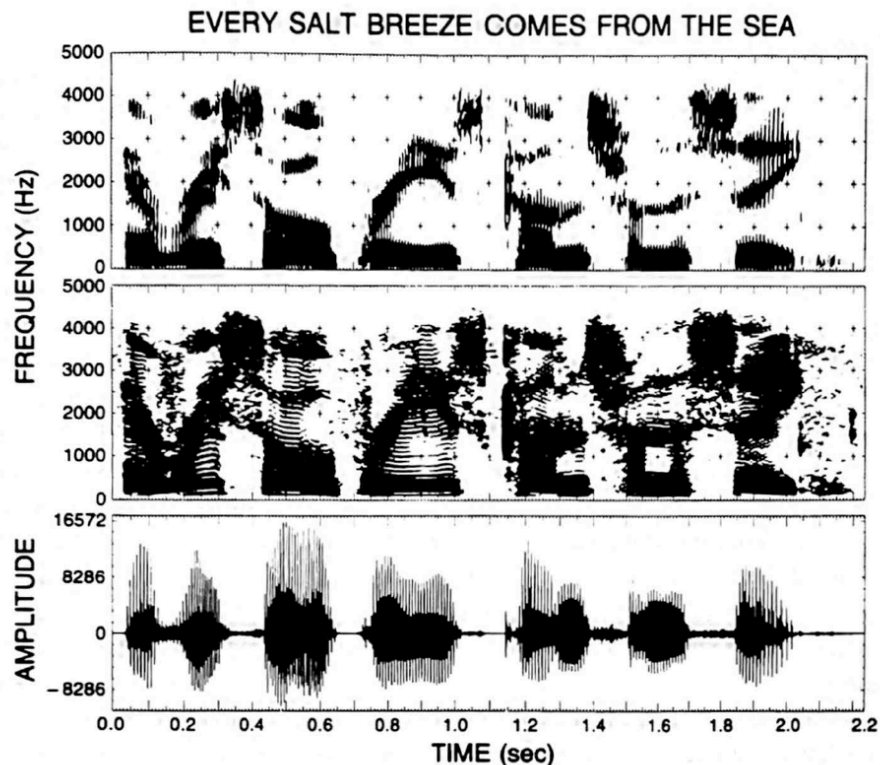


Figure 2.8 Wideband and narrowband spectrograms and speech amplitude for the utterance "Every salt breeze comes from the sea."

Rabiner, 50 ("Figure 2.8").

37. One speech recognition technique known well before the priority date of the '277 Patent was the statistical pattern-recognition approach to speech recognition. *Rabiner* (published in 1993) teaches the statistical pattern-recognition technique includes four main steps: feature measurement, pattern training, pattern classification, and decision logic. *Rabiner*, 82. *Rabiner* teaches the first step is "[f]eature measurement, in which a sequence of measurements is made on the input signal to define the 'test pattern.' For speech signals the feature measurements are usually the output of some type of spectral analysis technique, such as a filter bank

analyzer, a linear predictive coding [LPC] analysis, or a discrete Fourier transform (DFT) analysis.” *Id.* The second step, pattern training, involves creating reference patterns or models that characterize “the statistics of the features of the reference pattern.” *Id.* During the pattern classification step, an “unknown test pattern is compared with each (sound) class reference pattern and a measure of similarity (distance) between the test pattern and each reference pattern is computed.” *Id.* *Rabiner* further teaches that to “compare speech patterns (which consist of a sequence of spectral vectors), we require” “a local distance measure, in which local distance is defined as the spectral ‘distance’ between two well-defined spectral vectors[.]” *Id.* The fourth step is decision logic, “in which the reference pattern similarity scores are used to decide which reference pattern (or possibly which sequence of reference patterns) best matches the unknown test pattern.” *Id.*

38. It was also known to use lexical trees to organize vocabulary before the priority date of the '277 Patent. For example, *Young* teaches using a “lexical tree 500 that is initialized before processing begins. The lexical tree 500 represents the active vocabulary 230 based on the phonetic relationships between words in the vocabulary.” *Young* (Ex. 1017), 11:37-41. *Young* further teaches that the “lexical tree includes a root node 505 that represents new words entering the lexical tree. From the root node 505, the tree expands to a group 510 of nodes that correspond to phonemes with which words start.” *Young*, 11:41-44. *Young* teaches that “[e]ach

node in the group 510 represents a phoneme that appears at the beginning of one or more words” using an example where “a node 610 corresponds to all words in the vocabulary that start with the phoneme ‘H’. Together, the nodes in the group 510 include representations of the starting phoneme of every word in the vocabulary.”

Young, 11:47-53. Figs. 5 and 6 of *Young* are depicted below.

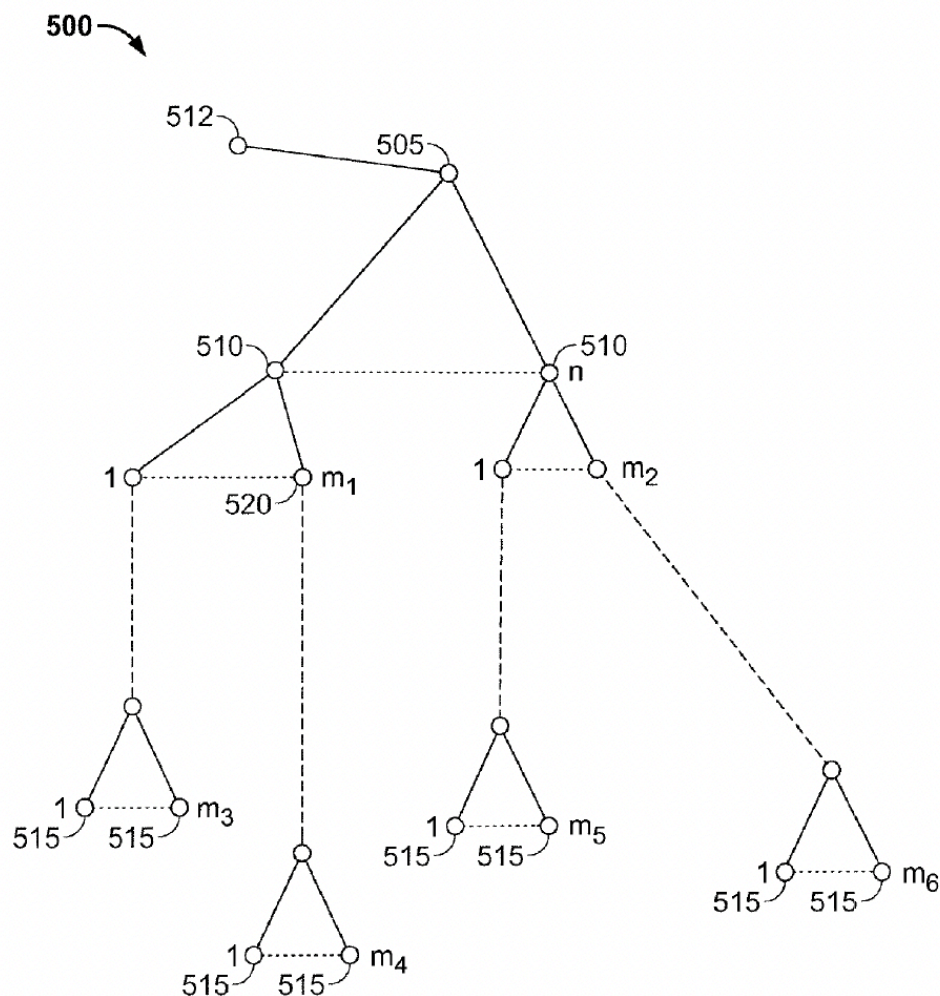


FIG. 5
(Prior Art)

Young, Fig. 5.

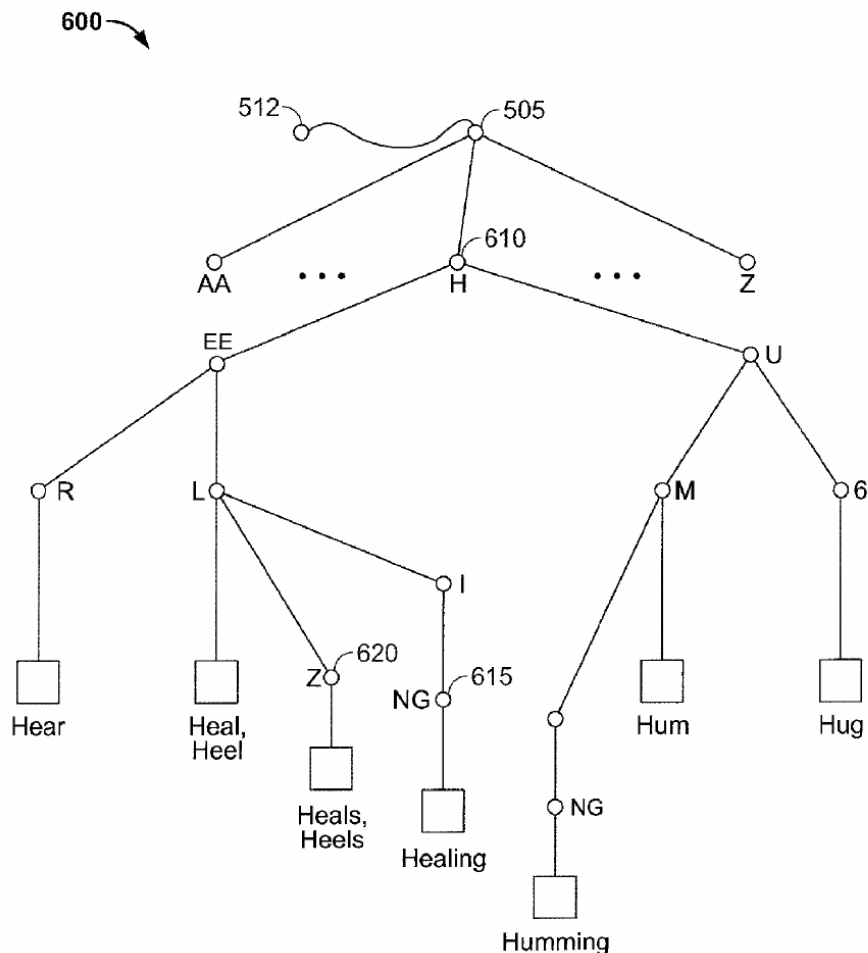


FIG. 6
(Prior Art)

Young, Fig. 6.

B. Feature Vectors

39. The '277 Patent describes feature vectors as comprising “a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame[.]” '277 Patent, 5:36-39; see also *id.*, 5:60-63, 6:8-10, 6:21-23, 6:32-34. Using feature vectors in speech recognition systems was well known before the priority date of the '277 Patent. For example, *Holmes* teaches that in “calculating a

distance between two words it is usual to derive a short-term distance that is local to corresponding parts of the words, and to integrate this distance over the entire word duration. Parameters representing the acoustic signal must be derived over some span of time, during which the properties are assumed not to change much.” *Holmes* (Ex. 1018), 13. *Holmes* further teaches that in “one such span of time the measurements can be stored as a set of numbers, or **feature vector**, which may be regarded as representing a point in multi-dimensional space. The properties of a whole word can then be described as a succession of feature vectors (often referred to as **frames**), each representing a time slice[.]” *Id.*

40. In general, feature vectors are representations of speech where each dimension of the vector corresponds to “some prescribed acoustic feature.” *Nadas* (Ex. 1019), 1:15-23 (reproduced below). For example, *Nadas* teaches an n-dimensional space where dimensions correspond to acoustic features.

In speech processing, speech is generally represented by an n-dimensional space in which each dimension corresponds to some prescribed acoustic feature. For example, each component may represent a amplitude of energy in a respective frequency band. For a given time interval of speech, each component will have a respective amplitude. **Taken together, the n amplitudes for the given time interval represent an n-component vector in the n-dimensional space.**

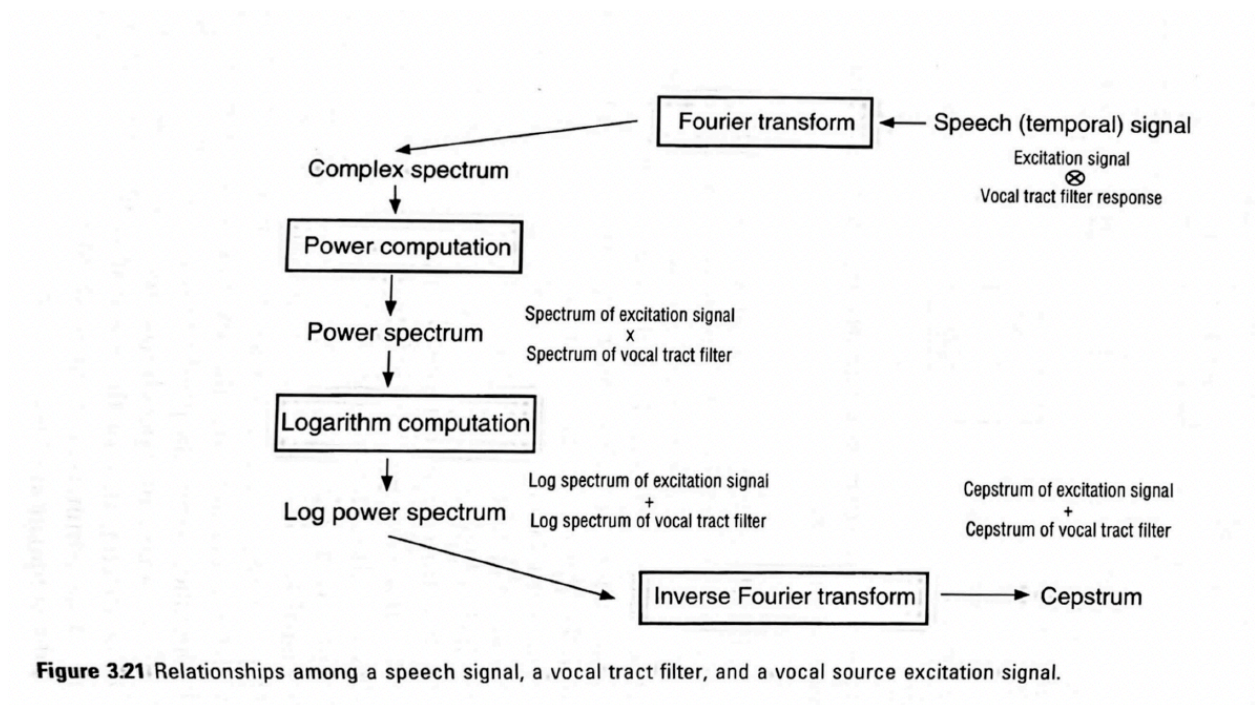
Id. (emphasis added). Additionally, *Nadas* teaches that “[w]hen an unknown speech input is uttered, for each time interval, a value is measured or computed for each of the n components, where each component is referred to as a feature. The values of all of the features are consolidated to form an n -component feature vector for a time interval.” *Nadas*, 1:32-37. *Nadas* teaches that “the components of the feature vectors (and prototype vectors)” may “correspond to well-known (1) Cepstral coefficients, (2) linear predictive coding coefficients, or (3) frequency band-related characteristics.” *Nadas*, 10:15-19. *Nadas* further teaches that a “speech type of event, such as a word or a phoneme, is characterized by a sequence of feature vectors in the time period over which the speech event was produced.” *Nadas*, 1:50-52. Further, *Jelinek* teaches that the “current continuous speech recognizers use as acoustic processor outputs vectors of cepstral coefficients [3].” *Jelinek* (Ex. 1020), 61.

41. In more detail regarding cepstral coefficients, *Katagiri* teaches that “a speech signal can be considered an output of a vocal tract filter ($H(z)[.]$ ” *Katagiri* (Ex. 1021), 56. *Katagiri* further teaches that for “speech recognition, we need to extract only the information based on the vocal tract filter, which conveys linguistic class identity, from the speech signal. The acoustic feature representation/modeling using cepstrum has also been used extensively in the speech recognition field.” *Katagiri*, 57. *Katagiri* teaches that “[c]epstrum is an inverse Fourier transformation

of a logarithmic power spectrum. A variable for cepstrum function is then defined as quefrency, which is effectively a time domain variable.” *Id.* Further, *Katagiri* discloses cepstrum is the inverse Fourier transform of the summation of a vocal tract filter spectrum and excitation spectrum.

A speech signal is originally a scalar value function (discrete sample sequence after sampling) of time that is defined as the convolution of the excitation signal and the impulse response of a vocal tract filter. Through the Fourier transform, the speech signal is converted to its corresponding power spectrum, which is a function of frequency and is defined as the production of the vocal tract filter spectrum and the excitation spectrum. A simple logarithmic operation converts production to summation. In the log power spectrum domain, the speech spectrum becomes the summation of the two log spectra, that of the vocal tract filter and that of the excitation signal. Cepstrum is accordingly defined as the inverse Fourier transform of the summation of these two spectra.

Id. *Katagiri* teaches the relationship between a speech signal, vocal tract filter, and a vocal source excitation signal in Fig. 3.21 below.



Katagiri, 58 (“Figure 3.21”). *Katagiri* further teaches “the smooth spectrum of the vocal tract filter is transformed to cepstrum components in the low quefrequency region, and the periodic spectrum of the (voiced) excitation signal is transformed to cepstrum components in the high quefrequency region.” *Katagiri*, 59. Further, *Katagiri* teaches the “information localization in the quefrequency domain is clearly useful for extracting the vocal tract filter information or, in other words, the speech-recognition-oriented linguistic class information.” *Id.*

42. As discussed above regarding the feature measurement step, a linear predictive coding (LPC) analysis may be performed which results in feature vectors. *Rabiner* teaches that a “very important LPC parameter set, which can be derived directly from the LPC coefficient set, is the LPC cepstral coefficients, $c(m)$.”

Rabiner, 146. “The cepstral coefficients, which are the coefficients of the Fourier transform representation of the log magnitude spectrum, have been shown to be a more robust, reliable feature set for speech recognition than the LPC coefficients, the PARCOR coefficients, or the log area ratio coefficients.” *Rabiner*, 146-147. Furthermore, *Rabiner* teaches that “the result of the LPC analysis is a vector of Q weighted cepstral coefficients and an appended vector of Q cepstral time derivatives[.]” *Rabiner*, 148. *Rabiner* further teaches that for “LPC vectors (and related feature sets such as LPC derived cepstral vectors), measures such as the likelihood and cepstral distance measures are generally used” as a similarity or distance measure.” *Rabiner*, 156. *Rabiner89* discloses in detail the process of calculating observation vectors O (i.e., feature vectors) from speech samples (i.e., a digital audio stream) by performing a linear predictive coding (LPC) “front end spectral analysis.” *Rabiner89* (Ex. 1022), 21. In particular, “[t]he overall system is a block processing model in which a frame of N_A samples is processed and a vector of features O_i is computed.” *Id.*

43. In another example with similar teachings to *Rabiner*, *Maes* teaches using a well-understood acoustic front-end to extract feature vectors. *Maes* (Ex. 1023), 5:6-10 (“[A]n acoustic front-end 110 which is, itself, well-understood in the art, is used to sample utterances in an overlapping fashion and to extract feature

vectors 120 therefrom, preferably as MEL cepstra, delta and delta-delta coefficients along with normalized log-energies.”).

44. In another example, *Aoyama* teaches a feature extraction unit that analyzes audio input and extracts information to generate a feature vector. In this example, *Aoyama* teaches performing a Mel Frequency¹ Cepstrum Coefficient (MFCC) analysis and extracting a feature vector utilizing a linear predictive coefficient (LPC) or Cepstrum coefficient.

The feature extraction unit 91 analyses the input audio data in each adequate frame, such as Mel Frequency Cepstrum Coefficient MFCC analysis, and outputs the resulting MFCC to the matching unit 92 and the unregistered word section processing unit 96 as the feature vector (feature parameter). Then, in the feature extraction unit 91, it is possible that such as the linear predictive coefficient, Cepstrum coefficient, line spectrum, power per the fixed frequency (output of filter bank) can be extracted as the feature vector.

¹ I note that the Mel scale is a frequency representation that approximates the frequency response of the human auditory system, i.e., how humans hear frequency.

The matching unit 92, referring to the acoustic model memory unit 93, the dictionary memory unit 94 and the grammar memory unit 95 in utilizing the feature vector from the feature extraction unit 91 as occasion demands, speech recognizes the voice (input speech) entered into the microphone 51 based on such as the Hidden Markov model (HMM) law.

Aoyama (Ex. 1024), [0138]-[0139] (emphasis added); *see also Aoyama*, [0150]-[0152], [0156]-[0160].

45. In yet another example of feature vectors, *Ittycheriah* '447 teaches generating feature vectors that are “thirty-nine dimension vectors ... as is well-understood in the art” but that “other dimensions could be used” as well.

The preferred, text independent form of speaker class recognition system in accordance with the invention preferably utilizes feature vectors developed by an acoustic front end 110. Such feature vectors are preferably computed on overlapping 30 msec. frames with shifts of 10 msec. Physiologically related MEL cepstra, delta and delta-delta feature vectors are preferred as feature vectors for efficiency and effectiveness of speaker identification or verification although other known types of feature vectors could be used. **Such feature vectors and others, such as LPC cepstra, are thirty-nine dimension vectors in the case of the preferred embodiment, as is well-understood in the art, but other dimensions could be used to achieve the same meritorious effects** in accordance with the principles of the invention.

Ittycheriah '447 (Ex. 1025), 4:54-67 (emphasis added). I note that the '277 Patent also describes feature vectors as representing “a point in an N-dimensional space. Different embodiments of the invention may use different values of N. Generally, we see values of N in the range of 20 to 39, although values outside this range are not excluded.” '277 Patent, 12:33-37.

46. *Stublely* teaches converting speech samples into feature vectors is advantageous because feature vectors are more compact and provide better speech recognition performance than the speech samples themselves.

The purpose of the signal processing component is to convert the utterance, which is a series of samples of speech, into a more compact series of feature vectors. Besides being more compact, the feature vectors provide better speech recognition performance since they tend to remove many speaker-dependent effects and improve the robustness of the system to noise.

Stublely (Ex. 1026), 11:30-36.

47. *Lee* teaches input speech is analyzed and represented by feature vectors that correspond to phonemes that are characterized by hidden Markov models (HMMs).

[I]nput speech is analyzed in signal frames, represented by feature vectors corresponding to phonemes making up individual words. The phonemes

are characterized by hidden Markov models (HMMs), and a Viterbi algorithm is used to identify a sequence of HMMs which best matches, in a maximum likelihood sense, a respective concatenation of phonemes corresponding to an unknown, spoken utterance.

Lee (Ex. 1027), 1:24-31 (emphasis added).

C. Acoustic Models

48. It was also known before the priority date of the '277 Patent to use acoustic models in speech recognition systems. For example, *Jelinek* teaches “acoustic modeling” includes “a statistical *acoustic model* of the speaker’s interaction with the acoustic processor.” *Jelinek*, 21. *Jelinek* teaches one type of acoustic model is a “*phonetic acoustic model*” that “is based on an intuitive linguistic concept.” *Jelinek*, 30. For example, *Jelinek* teaches creating a phonetic dictionary for vocabulary and creating a correspondence between words and sequences of “symbols from a predetermined phonetic alphabet[.]” *Id.*

49. In another example, *Schmandt* teaches using a set of templates or models as a representation of speech used by the speech recognizer.

A set of **templates** or **models**, which are descriptions of each word to be recognized, in the representation of speech used by the recognizer. The templates describe the words in the recognizer’s vocabulary, i.e., those words that the recognizer can identify. They are reference models

against which an input can be compared to determine what was spoken.

Schmandt, 46.

50. A further example of using acoustic models is taught by *Stubbley*. *Stubbley* teaches “comparing a series of observations representing unknown speech, to stored models representing known speech[.]” *Stubbley*, 1:5-10, 1:60-62. *Stubbley* teaches that “the models comprise representations of elements of speech[.]” *Stubbley*, 2:51-52. Further, *Stubbley* teaches in one example “the models comprise groups of representations of phonemes. This enables the models to be stored and used more efficiently.” *Stubbley*, 2:40-42.

51. In another example, *Ittycheriah* teaches storing a number of reference models in a memory accessible to a processor. *Ittycheriah* (Ex. 1028), 3:34-35 (“A reference model store 112 is connected to the model construction module 110 for storing the L reference models[.]”), *Ittycheriah* 4:30-32 (“Further, the reference model store 112 and the rank/distance store 120 may be electronic computer read/write memory or any other suitable memory.”).

52. In a further example, *Rabiner89* teaches characterizing signals in terms of signal models. *Rabiner89* teaches a “signal model can provide the basis for a theoretical description of a signal processing system which can be used to process the signal so as to provide a desired output.” *Rabiner89*, 1. Further, *Rabiner89*

teaches that such signal models include “statistical models in which one tries to characterize only the statistical properties of the signal,” with one example being a “hidden Markov model (HMM)” *Id.* Thus, one particular type of acoustic model is a Hidden Markov Model, which I discuss in detail below.

D. Hidden Markov Models

53. Before the priority date of the '277 Patent, Hidden Markov Models were used as a technique to “make pattern matching more powerful for dealing with variations in speech patterns.” *Schmandt*, 54. Indeed, *Jelinek* teaches that the “usual acoustic model employed in speech recognizers” is “the *hidden Markov model*[.]” *Jelinek*, 21. In general, a Hidden Markov Model (HMM) is “a two-stage probabilistic process that can be used as a powerful representation for speech. A Hidden Markov Model is a well-behaved mathematical construct[.]” *Schmandt*, 57. Further, it was known that HMMs consisted of states. “An HMM consists of a number of internal states; the model passes from an initial state to the final state as a step-by-step process generating an observable output at each step (state transition). For example, the states may correspond to phonemes contained in a word with the observable output corresponding to the presence or absence of a number of acoustic features.” *Schmandt*, 57-58. Additionally, “[a]t each step, the model can either move to a new state or stay in the current one. The model is ‘hidden’ in that we cannot observe the state directly but only its output. From the series of observable outputs, we can

attempt to guess when the model was in each state.” *Schmandt*, 58. “Alternatively, we can say whether some series of outputs was likely to have been generated by a particular HMM.” *Id.*

54. *Schmandt* teaches a “pedagogic example” of a “box-and-bowl setup” to explain Hidden Markov Models.

For example, consider the arrangement shown in Figure 7.10, in which a box and a bowl are each full of black balls and white balls. The box has many more black balls than white while white balls dominate the bowl. Starting with the box, we remove one ball from it at random and pass it to an observer in another room who cannot see what is being done. Then a normal six-sided die is tossed. If the result is less than four, then the source of the next ball is the bowl. If the die is greater than three, then the next ball will be selected from the box. The cycle is then repeated. Whenever we select from the box, a die throw of less than four shifts attention to the bowl. Once we start selecting from the bowl, we continue selecting from it unless the die shows a one, at which point we are finished. This model is likely to spend a majority of its cycles selecting from the bowl since a state transition from the box has a probability of $1/2$ of shifting to the bowl, but a bowl transition has a probability of only $1/6$ of terminating and $5/6$ of continuing with the bowl. We should expect this arrangement to initially present more black balls than

white to the observer (while it is in the box state) and then to produce more white balls. We should also expect more white balls overall. But keep in mind that this is a probabilistic process: Not all black balls come from the box and not all selections come from the bowl.

Id. *Schmandt* further teaches that the “box-and-bowl setup is a Hidden Markov Model; it is characterized by a number of *states* and associated probabilities for each *transition* from any state. The box and bowl are each a state; the rules about throwing the die define the transition properties.” *Schmandt*, 58-59. *Schmandt* also teaches that “[p]robability is expressed as a number between zero and one. If an event occurs with probability $1/2$, we would expect it to happen once in two trials. A probability of $1/6$ implies that we can expect the event once out of six trials. A smaller probability indicates that an event is less likely to occur.” *Schmandt*, 58.

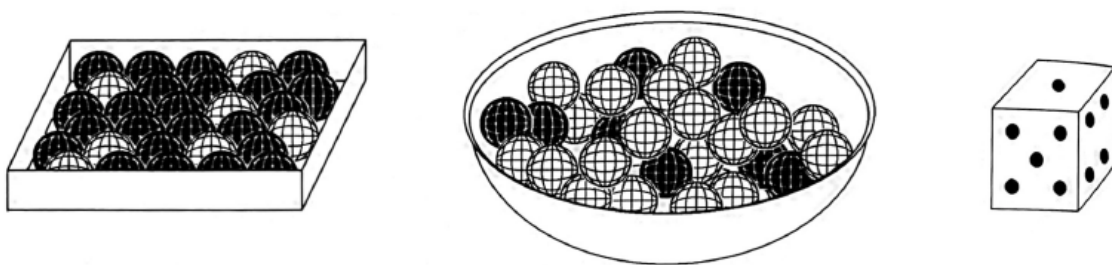


Figure 7.10. A box and a bowl full of colored balls illustrate a Hidden Markov Model.

Schmandt, 59 (“Figure 7.10”).

55. Further, *Schmandt* teaches a “more general HMM” as depicted in Figure 7.11 “A Markov Model” below. *Schmandt* teaches that “in each state $S(n)$

there is some probability $P(n,n)$ of remaining in that state and some probability $P(n, n+1)$ of transitioning to the next state.” *Schmandt*, 59. Additionally, *Schmandt* discloses that “[i]n some models, it may be possible to bypass a state or a series of states as indicated by the dashed arc labeled P_{13} in the figure. The sum of the probabilities of all the arcs leaving a state is one; the model accounts for every possible path through its states.” *Schmandt*, 59.

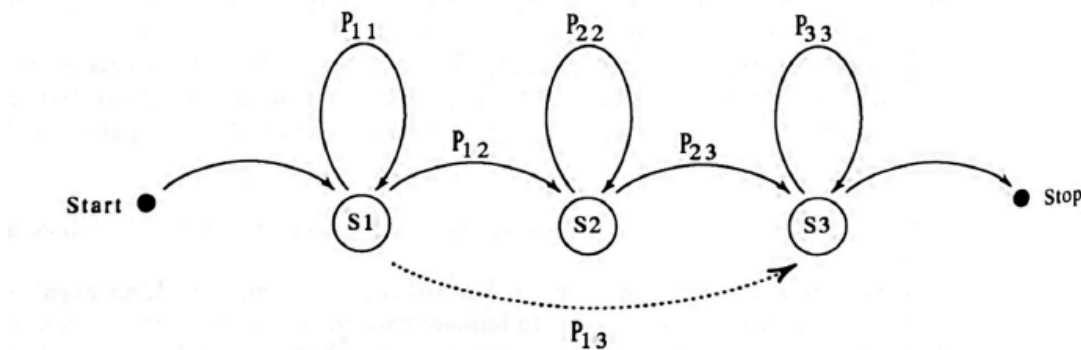


Figure 7.11. A Markov Model.

Schmandt, 59 (“Figure 7.11”).

56. *Schmandt* teaches that the example of the box-and-bowl setup and the more general HMM model apply to speech recognition because an HMM’s internal states can represent acoustic segments.

An HMM can be used to represent a word with internal states representing characteristic acoustic segments, possibly phonemes or allophones. The output of a state is a frame or vector of acoustic parameters or features; this output is probabilistic to

allow for variability in pronunciation and hence differences in the acoustic representation. The duration of an acoustic segment is a function of the number of steps in which the model is in the state corresponding to the segment. Staying in the same state, i.e. lengthening a phone, depends on the probability associated with the transition from that state to itself (P_{11} in Figure 7.11). Arcs such as P_{13} may be included to indicate that an intermediate state **S2** is optional during pronunciation, such as the second syllable in ‘chocolate.’

In terms of the basic recognizer components, the ‘templates’ consist of a set of HMMs with one HMM associated with every word. The acoustic representation of an input specimen of speech is *not* a collection of states but rather a set of acoustic parameters corresponding to the possible observed sequences from the template HMMs. For the HMM-based recognizer, classification consists of determining which HMM has the highest probability of producing the observed acoustic sequence. The Viterbi algorithm is a commonly used method of solving this classification problem[.]

Schmandt, 59-60 (emphasis added).

57. *Schmandt* further teaches that “Hidden Markov Models are a powerful representation of speech and consequently are used in many speech recognition systems currently under development.” *Schmandt*, 60. *Schmandt* also teaches that

HMMs can be used to “represent not only the phones of each word but also the probabilities of transitioning from one word to another.” *Id.*

58. *Rabiner* teaches that an observation vector in an HMM can be “LPC-derived weighted cepstral coefficients and weighted cepstral derivatives[.]” *Rabiner*, 231. For example, *Rabiner* teaches that the observation vector in the HMM can be a “38-component vector consisting of 12 cepstral coefficients, 12 delta cepstrum coefficients, 12 delta-delta cepstral coefficients, delta log energy, and delta-delta log energy.” *Rabiner*, 280; *see also Rabiner*, 301-302 (teaching spectral vectors with 38 components).

59. Another example of using an HMM for speech recognition well before the priority date of the '277 Patent is taught by *Nadas*. *Nadas* (issued in 1990) teaches an acoustic coprocessor 200 for speech recognition and “contemplates a probabilistic embodiment in which each prototype—rather than identifying a vector—corresponds to a probabilistic finite state machine PFSM (or Markov model).” *Nadas* (Ex. 1019), 9:68-10:4.

60. In another example, *Aoyama* teaches using an HMM as the acoustic model for speech recognition and said HMM model is stored in an acoustic model memory unit.

**The matching unit 92, referring to the acoustic model
memory unit 93, the dictionary memory unit 94 and
the grammar memory unit 95 in utilizing the feature**

vector from the feature extraction unit 91 as occasion demands, speech recognizes the voice (input speech) entered into the microphone 51 based on such as the Hidden Markov model (HMM) law.

More specifically, **the acoustic model memory unit 93 memorizes acoustic model** (e.g., including the standard pattern to be used in DP (Dynamic Programming) matching, other than HMM) showing the acoustic feature on the sub-words such as phoneme, syllable, and phoneme series in the audio language for identifying the speech. **Here, since the speech recognition is conducted based on the Hidden Markov Model law, the HMM will be used as the acoustic model.**

Aoyama, [0139]-[0140] (emphasis added); *see also Aoyama*, [0150]-[0151].

61. In yet another example, *Stubley* teaches using HMMs as acoustic models.

Advantageously, the models are represented as finite state machines with probability distribution functions attached. Continuously varying signals can be modeled as a series of stationary states, with a probability function at each state, indicating whether the signal moves to the next state or stays at the given state, after a given time interval. An example is the Hidden Markov Model (HMM). This enables the models to contain a representation of patterns

with varying rates of change, yet a common underlying pattern.

Stubley, 2:29-37; *see also Stubley*, 7:23-35, 12:29-44 (“**The modelling paradigm used here is to construct models using hidden Markov models (HMMs)[.]**”) (emphasis added).

62. In another example, *Lee* teaches analyzing input speech represented by feature vectors that correspond to phonemes of words. *Lee* further teaches characterizing phonemes by Hidden Markov Models.

[I]nput speech is analyzed in signal frames, represented by feature vectors corresponding to phonemes making up individual words. The phonemes are characterized by hidden Markov models (HMMs), and a Viterbi algorithm is used to identify a sequence of HMMs which best matches, in a maximum likelihood sense, a respective concatenation of phonemes corresponding to an unknown, spoken utterance.

It is well known that each HMM comprises model parameters, e.g., mixture components which are characterized by Gaussian distributions. **In a learning phase in a speech recognition system, the HMMs are adapted to input speech by a user to adjust to the particular speech characteristics of the user, thereby increasing accuracy of the speech recognition.**

Lee, 1:24-38 (emphasis added).

63. In yet another example, *Toyoda* teaches a voice recognition module (sensor signal processor 31) that uses an HMM. *Toyoda* (Ex. 1029), 7:48-50. Furthermore, *Toyoda* teaches referencing an acoustic model memory that stores an acoustic model representing phonemes while a matching unit 42 recognizes voice input.

Using the feature parameters from the feature parameter extractor 41, the **matching unit 42 recognizes the voice input to the microphone 9 based on the continuous HMM model while referencing an acoustic model memory 43**, a dictionary memory 44, and a grammar memory 45 as necessary.

The acoustic model memory 43 stores an acoustic model that represents an acoustic feature such as phonemes and syllables in a voice to be recognized. Since voice recognition is here carried out using the continuous HMM, an HMM is employed.

Toyoda, 7:61-8:5 (emphasis added).

E. Distance Calculations

64. Distance calculations to compare a feature vector to a template vector, reference model, or states of an acoustic model were well known before the priority date of the '277 Patent. For example, *Rabiner* teaches that one step in speech

recognition is pattern classification, “in which the unknown test pattern is compared with each (sound) class reference pattern and a measure of similarity (distance) between the test pattern and each reference pattern is computed.” *Rabiner*, 82. In general, said distance may refer to a “similarity,” “closeness,” or “probability” between said feature vector and the acoustic state (or template vector, reference vector, model, or portion of a model) to which the feature vector is compared, as I discuss below.

65. One example of computing distance between a feature vector and at least one reference model is taught by *Nadas*. *Nadas* teaches calculating the distance between a prototype vector and a normalized output vector. For example, *Nadas* teaches “[i]n associating prototype vectors with feature vectors, **the feature vector for each time interval is typically compared to each prototype vector. Based on a predefined closeness measure, the distance between the feature vector and each prototype vector is determined and the closest prototype vector is selected.**” *Nadas*, 1:44-49 (emphasis added).

66. It was also known to use probability as a “distance.” For example, when discussing Hidden Markov Models in speech recognition, *Holmes* teaches “[t]he best matching word in a recognition task can be defined as the one whose model is most likely to produce the observed sequence of feature vectors. What we have to calculate for each word is thus not a ‘distance’ from a template, but the *a posteriori*

probability that its model could have produced the observed set of feature vectors.

We do not actually have to make the model produce the feature vectors, but we use the known properties of each model for the probability calculations.” *Holmes*, 31.

67. One method for performing distance calculations was to use a Mahalanobis distance. It was known to use Mahalanobis distance calculations in speech recognition systems before the priority date of the '277 Patent. *Dudemaine* teaches that a Mahalanobis distance “gives a representation of the probability of a particular feature vector matching a given transition in the finite state model of the word in the [] dictionary.” *Dudemaine* (Ex. 1030), 8:44-47. In another example, *Ittycheriah* '447 defines a Mahalanobis distance. *Ittycheriah* '447, 6:60-64 (“[A] distance measure (e.g. along a Mahalanobis distance which is a weighted Euclidian distance in which weights are assigned according to the inverse of the variances of each of the dimensions or parameters of the feature vector[.]”). Additionally, *Kosaka* teaches the “Mahalanbois’ [*sic*] distance between the input pattern and the reference pattern for each class was calculated, and the vowel class giving the minimum distance is selected as the recognition result.” *Kosaka* (Ex. 1031), 23:3-6.

68. In a further example, *Ittycheriah* teaches using a Mahalanobis method for computing distance measures in a speech recognition system.

It is to be understood that any conventional method for computing the distance measure between the distributions of the temporary biometric model and

the distributions of the N models may be implemented in the present method such as the Euclidean, Mahalanobis and Kullback-Leibler methods. It is also understood that these methods for computing distances are typically used for biometric systems in which Gaussian mixtures are utilized (such as speaker recognition).

Ittycheriah, 5:60-67 (emphasis added).

69. In another example, *Yang* teaches using a Mahalanobis distance for distance calculation in a speech recognition system. *Yang* (Ex. 1032), [0035] (“In the similarity calculation portion 40, [M]ahalanobis’ distance algorithm is employed for distance measure[.]”).

70. In a further example, *Maes* teaches calculating the Mahalanobis distance for clustering feature vectors.

Such feature vectors are preferably computed on overlapping 25-30 msec. frames with shifts of 10 msec. Physiologically related (e.g. characterizing vocal tract signatures such as resonances) MEL cepstra, delta and delta-delta feature vectors are preferred as feature vectors for efficiency and effectiveness of speaker identification or verification although other known types of feature vectors could be used. **Such feature vectors** and others, such as LPC cepstra, **are usually thirty-nine dimension vectors, as is well-understood in the art. The resulting feature vectors are clustered into about sixty-five**

codewords (the number is not critical to practice of the invention) in accordance with a Mahalanobis-distance.

Maes, 5:17-29 (emphasis added).

71. It was also known to use neural networks as distance calculations before the priority date of the '277 Patent. For example, *Jelinek* teaches using artificial neural networks (ANNs) as a basis of speech recognition. *Jelinek*, 43. *Jelinek* teaches that “Richard and Lippmann showed (among other properties) that ANNs can be used to estimate a posteriori probabilities [22], and Bourlard and Morgan have taken advantage of this fact in formulating a combined HMM and ANN approach to speech recognition [23].” *Id.* In a further example, *Jelinek* teaches that the “system constructed by Robinson, Hochberg, and Renals achieves state-of-the-art results in speaker independent large vocabulary speech recognition [24].” *Id.*

F. Gaussian Distribution and Probability

72. It was known to use Gaussian distributions and probabilities before the priority date of the '277 Patent. In one example of a speech recognition system using HMM and Gaussian distributions, *Lee* teaches that it was “well known that each HMM comprises model parameters, e.g., mixture components which are characterized by Gaussian distributions.”

[I]nput speech is analyzed in signal frames, represented by feature vectors corresponding to phonemes making up individual words. The phonemes

are characterized by hidden Markov models (HMMs), and a Viterbi algorithm is used to identify a sequence of HMMs which best matches, in a maximum likelihood sense, a respective concatenation of phonemes corresponding to an unknown, spoken utterance.

It is well known that each HMM comprises model parameters, e.g., mixture components which are characterized by Gaussian distributions. In a learning phase in a speech recognition system, the HMMs are adapted to input speech by a user to adjust to the particular speech characteristics of the user, thereby increasing accuracy of the speech recognition.

Lee, 1:24-38 (emphasis added).

73. In another example, *Peters* teaches modeling acoustic characteristics with an HMM (specifically, Continuous Density Hidden Markov Model) that involves parameterization using a Gaussian mixture, which “is a weighted composition of a number of multivariate Gaussian probability density functions (pdfs).”

Commonly, methods for modeling the speaker specific speech pattern make use of Continuous Density Hidden Markov Models (CDHMM) to model the acoustic characteristics. The ability of a CDHMM to model speech depends on the features, the topology, the

parameterization and the training required for the CDHMM. In general, these four factors interact in a complex manner. **Conventionally, the speech models are characterized by a sequence of states, the sequences of states being linked by uni-directional transitions and self-loop transitions that permit hesitation in a same given state. Typically, the parameterization chosen to define a given state is a Gaussian mixture, which is a weighted composition of a number of multivariate Gaussian probability density functions (pdfs).**

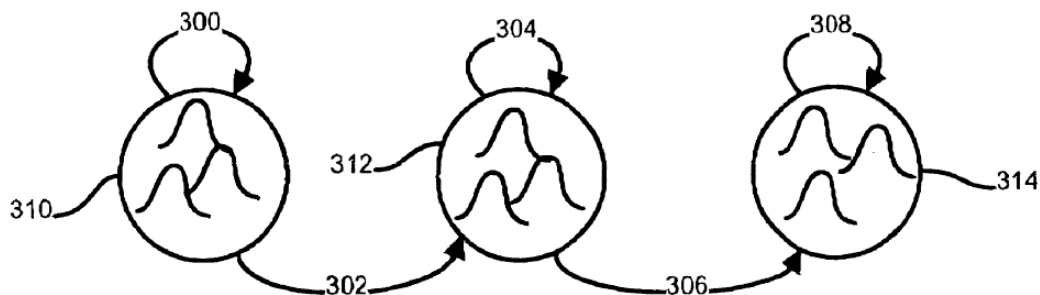
Peters (Ex. 1011), 1:35-48. *Peters* further teaches states of Hidden Markov Models associated with parameters that define Gaussian mixture of probability density functions.

FIG. 3 of the drawings shows a conceptual diagram of a simple speech model in accordance with a specific example of implementation of the invention. In FIG. 3, the states 310, 312 and 314 are depicted as circles and the transitions 302, 306, 300, 304 and 308 are depicted as arrows. Transitions 302 and 306 are inter-state transitions being indicative of transitions between two states. Transitions 300, 304 and 308 are self-loops being indicative of transitions where the originating states and the destination states are the same state. **In this topology, each state 310, 312 and 314 is indicative of a Hidden Markov Model (HMM) state and is associated to a set**

of parameters defining a Gaussian mixture of probability density functions. Similarly, each transition 302, 306, 300, 304 and 308 is associated to a probability value. The set of parameters defining the Gaussian mixture of probability density functions for a given state in a model is an estimate of the relative probability of each component of a given Gaussian mixture. The probabilities associated to a given transition are estimates of the relative probability of hesitating in or leaving a given state.

Peters, 4:63-5:15; see also *Peters*, 5:24-6:1. *Peters*, Fig. 3, which discloses states and transitions, is depicted below.

Figure 3



Peters, Fig. 3.

G. Speech Recognition System Hardware

74. It was known in the art before the priority date of the '277 Patent to use coprocessors to offload processing tasks in a speech recognition system to free up resources for the main processor (e.g., central processing unit (CPU)). For example,

Asghar teaches utilizing a digital signal processor (DSP) to perform mathematical functions offloaded from the CPU to advantageously increase system efficiency.

A digital signal processor is essentially a general purpose microprocessor which includes special hardware for executing mathematical functions at speeds and efficiencies not usually associated with microprocessors. In current computer system architectures, DSPs are used as co-processors and operate in conjunction with general purpose CPUs within the system. For example, current computer systems may include a general purpose CPU as the main CPU and include one or more multimedia or communication expansion cards which include dedicated DSPs. **The CPU offloads mathematical functions to the digital signal processor, thus increasing system efficiency.**

Asghar (Ex. 1033), 2:21-33 (emphasis added). *Asghar* teaches DSPs perform functions including speech recognition. *Asghar*, 1:47-54.

75. It was known in the art before the priority date of the '277 Patent that DSPs were programmable devices. For example, *Newton's Telecom Dictionary* teaches one of the main advantages of a digital signal processor is its programmability. *Newton* (Ex. 1008), 5 ("There are two main advantages of DSPs – first, they have powerful mathematical computational abilities. ... The second

advantage of a DSP lies in the programmability of digital microprocessors.”). *Id.* (“DSPs are used extensively in telecommunications for tasks such as echo cancellation, call progress monitoring, voice processing and for the compression of voice and video signals as well as new telecommunications applications such as wireless LANs and next-generation cellular data and cellular Internet services.”).

76. One example of a well-known DSP that was widely available since the 1980s was the Motorola DSP56000. *Kloker* is an IEEE article from Motorola published in 1986, which teaches the Motorola DSP56000 “brings 10.25-MIPS performance to digital signal processing and retains enough similarities to other Motorola microprocessors to make it easy to learn and program.” *Kloker* (Ex. 1009), 1. *Kloker* teaches the “Motorola DSP56000 is a high-performance, user-programmable digital signal processor implemented in high-density, low-power CMOS technology.” *Id.* Additionally, *Kloker* teaches that “the DSP56000 has hardware and software features specifically added to support digital signal processing” and “it shares enough similarities with other Motorola microprocessors and single-chip microcomputers that it can be used as a very fast microprocessor in any application that requires high-speed calculations or real-time response.” *Id.* Further, *Kloker* teaches the Motorola DSP56000 combines “a core processor, RAM, ROM, peripheral interfaces, and a memory expansion interface on a single, 88-pin chip[.]” *Id.* *Kloker* also teaches that “[m]any instructions allow the programmer to

keep each execution unit busy, thus enhancing program execution speed. The user can easily program the DSP56000 from the programming model without needing to refer to a complicated hardware block diagram.” *Kloker*, 3. *Kloker* further teaches that the “regularity of the programming model also makes possible high-level-language programming of the DSP56000.” *Kloker*, 4. Figure 5 below depicts the memory maps of a DSP56000.

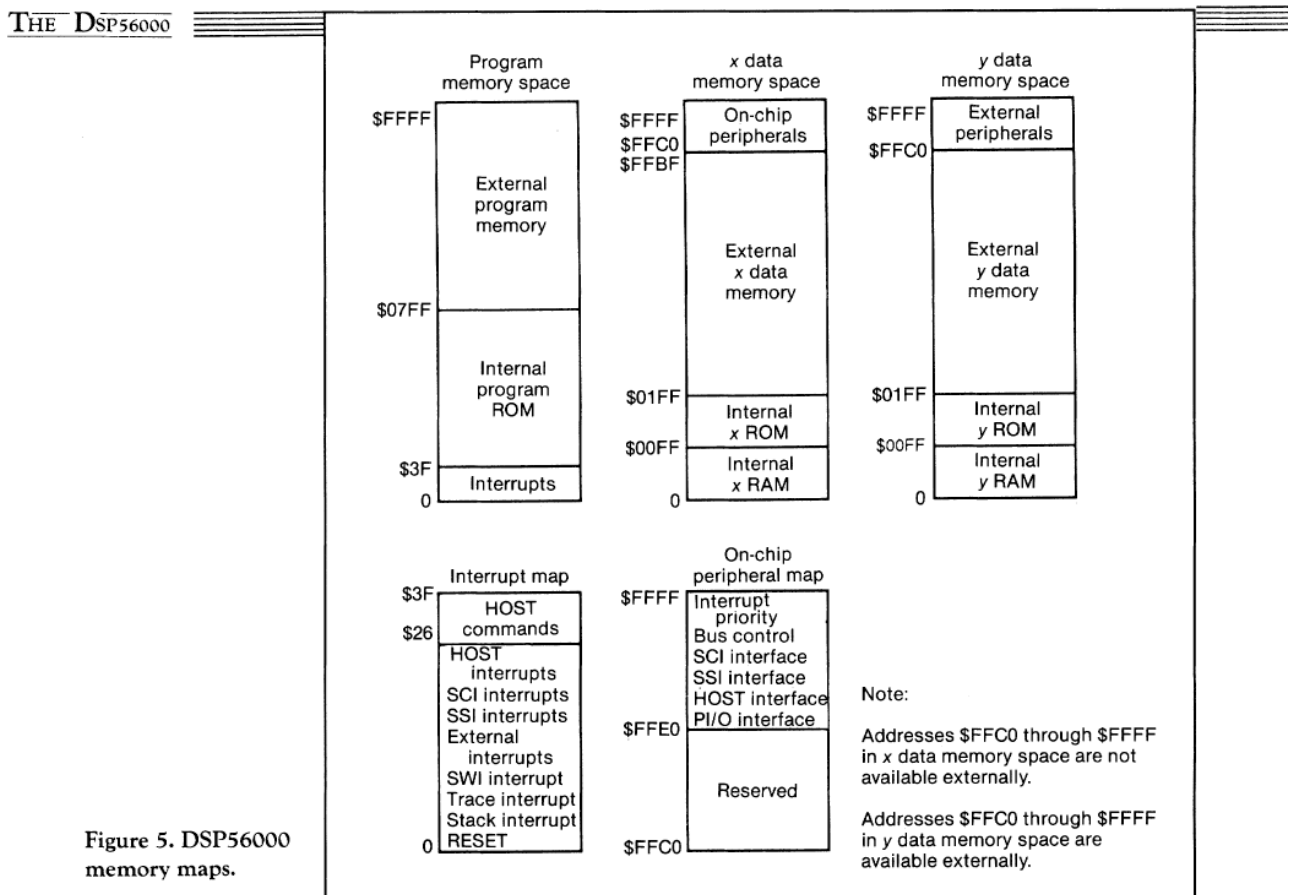


Figure 5. DSP56000 memory maps.

Kloker, 6 (“Figure 5. DSP56000 memory maps”).

77. It was also known to have a calculating apparatus and acoustic model memory on the same integrated circuit before the priority date of the '277 Patent.

For example, *Toyoda* teaches a sensor signal processor 31 also referred to as a “voice recognition module” that includes an acoustic model memory 43 and matching unit 42 (i.e., calculating apparatus), which a POSITA would have understood would have been on a single integrated circuit (sensor signal processor 31) as I discuss below.

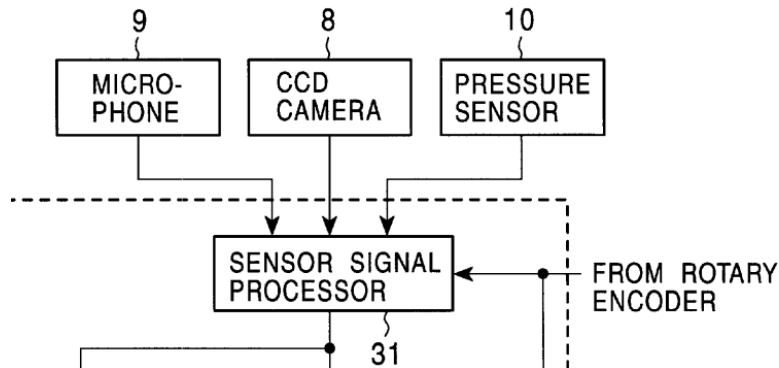
78. *Toyoda* teaches a voice recognition module (sensor signal processor 31) that uses an HMM. *Toyoda*, 7:48-50. Furthermore, *Toyoda* teaches referencing an acoustic model memory that stores an acoustic model representing phonemes while a matching unit 42 recognizes voice input.

Using the feature parameters from the feature parameter extractor 41, the **matching unit 42 recognizes the voice input to the microphone 9 based on the continuous HMM model while referencing an acoustic model memory 43**, a dictionary memory 44, and a grammar memory 45 as necessary.

The acoustic model memory 43 stores an acoustic model that represents an acoustic feature such as phonemes and syllables in a voice to be recognized. Since voice recognition is here carried out using the continuous HMM, an HMM is employed.

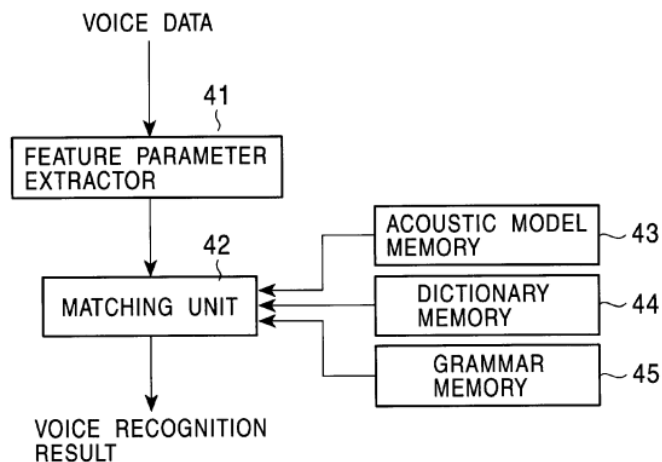
Toyoda, 7:61-8:5 (emphasis added). Sensor signal processor 31 (also referred to as a voice recognition module) is depicted in Fig. 4 below.

FIG. 4



Toyoda, Fig. 4. *Toyoda* teaches that Fig. 7 is a “functional block diagram of a portion of the sensor signal processor 31 shown in FIG. 4[.]” *Toyoda*, 7:43-47. Acoustic model memory 43 and matching unit 42 (i.e., calculating apparatus) are depicted in Fig. 7 below.

FIG. 7



Declaration of Christopher Schmandt
Patent No. 7,979,277

Toyoda, Fig. 7. Therefore, a POSITA would have understood that acoustic model memory 43 and matching unit 42 (both of which are part of sensor signal processor 31) would have been on the same integrated circuit (i.e., sensor signal processor 31).

79. Additionally, it was known to use a single integrated circuit containing multiple digital signal processors (DSPs) before the priority date of the '277 Patent. One example is taught by *Boike*, which was filed October 11, 2001. *Boike* teaches a “system-on-a-chip (SoC) integrated circuit 10 containing multiple digital signal processors (DSPs).” *Boike* (Ex. 1006), 2:52-54. *Boike* depicts such DSP subsystems 18-21 as part of DSP/SoC 10 (i.e., integrated circuit 10). “FIG. 1 is a block diagram depicting a typical configuration of a DSP/SoC multiple digital signal processor integrated circuit.” *Boike*, 2:42-44. Annotated Fig. 1 below depicts a single integrated circuit 10 (annotated blue) and DSP subsystems 18-21 (annotated green).

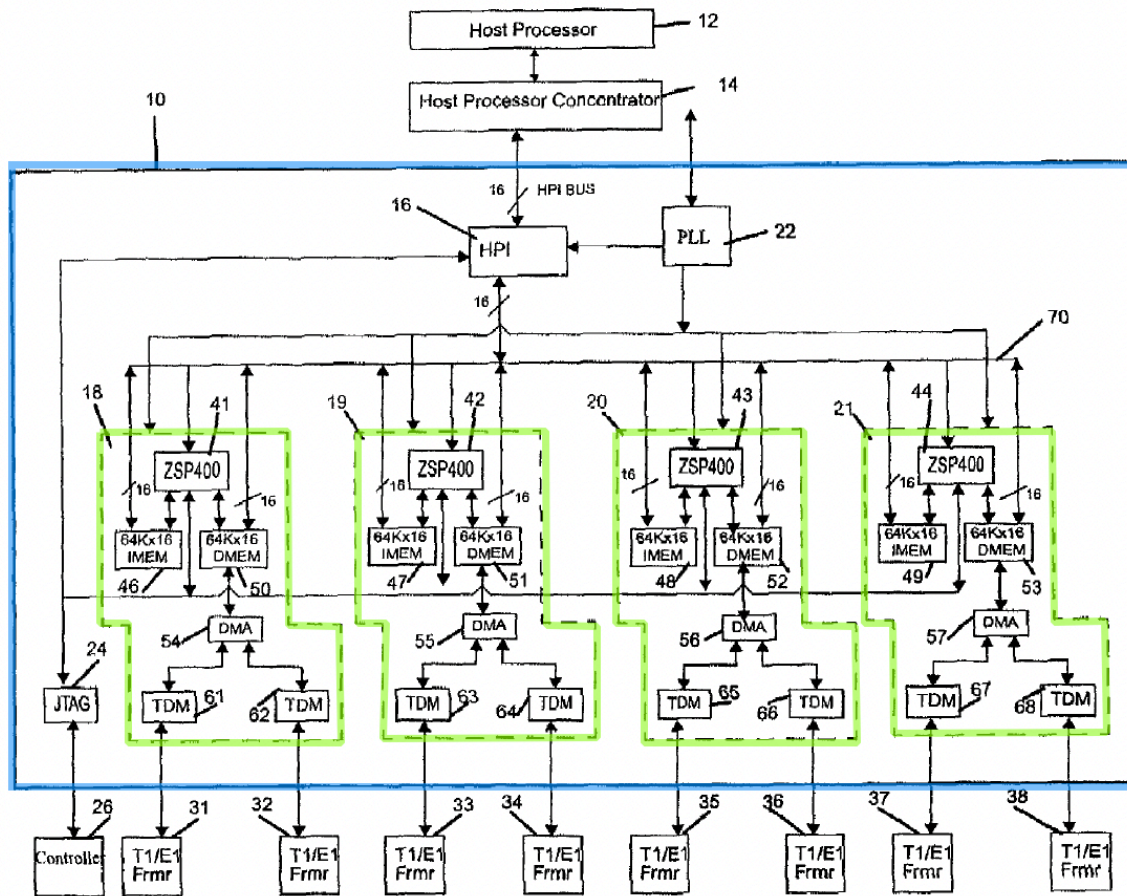


FIGURE 1

Boike, Fig. 1 (annotated). In particular, *Boike* teaches numerous advantages of such an architecture that places multiple DSPs on a single integrated circuit.

Several advantages can be gained by placing multiple DSPs on a single chip rather than having only one DSP on a chip. First, the amount of space on a circuit board taken up by the DSPs is reduced. Under the traditional approach, if four DSPs were needed in a circuit, four separate chips would have to be placed on the circuit board. When four DSPs are placed on a single chip,

only one chip is needed instead of four and the amount of space on the circuit board used by the DSPs is reduced accordingly. Electrical energy tends to be wasted by the random access memory, input/output, and other peripherals on each chip and particularly by the input/output ports. The use of a multi-DSP chip reduces this waste by reducing the number of chips on the board. Connections between the multiple DSPs on one chip do not need input/output circuits, but instead operate at the low internal power levels. **Thus, the amount of power consumed by the circuit board and the amount of heat generated by the board are reduced. The reductions in space and energy consumption contribute to a cost savings for multi-DSP as opposed to single-DSP chips. The use of multiple DSPs on a single chip instead of on separate chips also increases processing speed by reducing the distance between the DSPs and decreasing the number of interconnections among them.**

Boike, 1:19-42 (emphasis added); *see also Boike*, 2:1-9, 2:21-35. *Boike* also teaches that “[p]rior to the development of the present invention, at least one chip was known to exist that improved on the traditional configuration by placing multiple DSPs on a single chip. The Texas Instruments TMS320VC5441 Fixed-Point Digital Signal Processor contains four DSPs in a single integrated circuit.” *Boike*, 1:43-50.

80. In another example of multiple DSPs on a single integrated circuit, *Ozcelik* (filed October 26, 1998 and issued March 21, 2000), teaches a “circuit arrangement and method that utilize a distributed extensible processing architecture to allocate various DSP functions or operations between multiple processing cores disposed on an integrated circuit device.” *Ozcelik* (Ex. 1034), 2:54-59, 3:53-65, 5:26-34. *Ozcelik* teaches that such a configuration is advantageous: “The benefits of integrating multiple functions onto the same chip often include greater performance, lower design and manufacturing costs, reduced component size, and reduced power requirements, among others.” *Ozcelik*, 1:47-51.

H. Pipelining

81. It was well known prior to the '277 Patent to use pipelining to improve data processing. Specifically, it was well known to pipeline data in speech recognition systems by passing data through from feature vector extraction to similarity computations (indicating the similarity between feature vectors and particular states of acoustic models) to search stages used to ultimately determine words and phrases.

82. One example of pipelining data in general is *Luick*. For example, *Luick* teaches embodiments including Arithmetic Logic Units (ALUs) in a pipelined processor.” *Luick*, 1:5-10. *Luick* teaches:

One manner of parallelization is known as “pipelining”,
where instructions are fed into a pipeline for an execution

unit in a processor that performs different operations necessary to process the instructions in parallel. For example, to process a typical instruction, a pipeline may include separate stages for fetching the instruction from memory, executing the instruction, and writing the results of the instruction back into memory. Thus, for a sequence of instructions fed in sequence into the pipeline, as the results of the first instruction are being written back into memory by the third stage of the pipeline, a next instruction is being executed by the second stage, and still a next instruction is being fetched by the first stage. While each individual instruction may take several clock cycles to be processed, since other instructions are also being processed at the same time, the overall throughput of the processor is much greater. With respect to pipelining, the term “stage” generally refers to the combinational logic between registers or latches.

Luick (Ex. 1054), 1:20-37.

83. As another general example of pipelining data, *Simar* teaches “phases of the pipeline” in Figure 11 and “the operation of the pipeline based on clock cycles and fetch packets” in Figure 13:

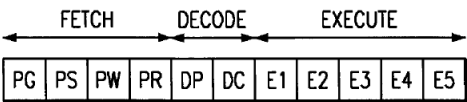


FIG. 11

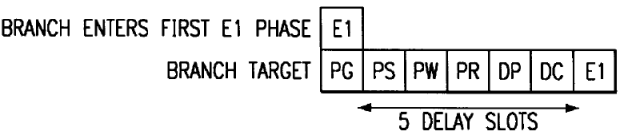


FIG. 12

FETCH PACKET	CLOCK CYCLE																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
n	PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5														
n+1		PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5													
n+2			PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5												
n+3				PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5											
n+4					PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5										
n+5						PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5									
n+6							PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5								
n+7								PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5							
n+8									PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5						
n+9										PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5					
n+10											PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5				
n+11												PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5			
n+12													PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5		

FIG. 13

Simar (Ex. 1039), Figs. 11-13; *see also*, *Simar*, 7:6-9, 13:30-35, 17:30-32, 20:22–23:45, 30:65–31:37 (collectively discussing pipelining data including in reference to DSP processors).

84. I note that although *Simar* discusses internal DSP architectures, the general and well-known advantages of pipelining apply to software architectures as well.

85. As an example of pipelining in speech recognition, *Ravishankar*² discusses pipelining in different aspects of the well-known search stage, pipelining between similarity indications and the search stage, and related concurrency. *Ravishankar*, 106-108.

86. As another example of pipelining in speech recognition, *Baumgartner* teaches pipelining within the similarity indication stage of speech recognition and specifically in *Baumgartner*'s adder circuitry. *Baumgartner*, [0017]-[0019], [0106]-[0107], Figs. 7-9, 24.

87. As an example of pipelining from a feature vector extraction stage to a similarity indication stage to a search stage, *Brown* teaches:

Attention is first directed to FIG. 1 which is a block diagram of a speech recognition system 100 embodying the principles of the invention. The recognition system 100 includes a number of functional blocks including, a linear-predictive-coding-(LPC)-based feature extractor 105, a

² I have reviewed the Declaration of June Ann Munford (Ex. 1035 ("*Munford*")). *Munford* attests to *Ravishankar*'s public availability date and that Ex. 1012 (a scan done by June Ann Munford of a physical copy of *Ravishankar* from Carnegie Mellon University) is a true and correct copy of *Ravishankar*. *Munford* (Ex. 1035), 1-15.

vector quantizer 110, a mixture probability processor 115, a phone probability processor 120, a word probability processor 125, a grammar processor 130, and a traceback processor 135. The functionality of each block of the recognition system may be performed by a respective different processor, or the functionality of several or all blocks may be performed by the same processor. Furthermore, each stage can include multiple processing elements. The stages are pipelined and their operations are synchronized with the rate at which frames of the incoming speech signal are generated.

Brown (Ex. 1036), 3:32-48.

88. These are just a handful of references evidencing the well-known nature of pipelining data in general, including in specific ways in speech recognition.

I. Interrupts

89. Interrupt signals in speech recognition systems to inform different processors when data was finished or ready was well-known in general prior to the '277 Patent. For example, *Atkas 1989* teaches

The inter-DSPbus allows fast communication between the master and slave processors without interfering with the VMEbus. A global memory scheme is realized in order to perform data transfers between the processors. **Control is handled by interrupts.** In order to test, download and control the system, the host machine has read and write

access to the whole memory of the AkuFE system. Because of its configurability and free programmability, the AkuFE system **can be used in a wide range of signal processing applications, such as speech analysis and synthesis, speech coding and speech recognition.** Used as a fast array processor for algorithms with extensive computational requirements, it can highly improve the performance of a standard workstation. The codebook generation /4/ particularly can be accelerated by employing the VQ Chip on the slave board in connection with a host processor. Further applications are spectral analysis, instrumentation and image processing.

Atkas 1989 (Ex. 1058), 1.

90. As another example, *Atkas 1991* teaches a multi-DSP architecture for acoustic signal processing and further teaches “[a] global memory scheme is realized in order to perform data transfers between the processors. For a flexible VMEbus and inter-DSP communication an interrupt control is provided[.]” *Atkas 1991* (Ex. 1059), 2.

91. As a further example, *Wu* teaches

An audio bus interface system and method interfaces a **plurality of digital signal processing devices** to an audio bus to facilitate **variable processing loading on the DSPs.** The audio bus contains frames with synchronization data. **The system utilizes a**

programmable interrupt controller for each digital signal processing unit. The programmable interrupt controller controls the rate at which a given DSP can be interrupted. The respective digital signal processor controls the programmable interrupt controller to maximize its throughput. Also, an audio format translator allows differing audio format protocols to be processed by the same audio DSP.

Wu (Ex. 1060), Abstract, 2:50–3:21.

92. Accordingly, interrupt signals, including interrupt signals between processors in speech recognition systems, were well-known.

J. Prior Art Speech Recognition Systems

93. Numerous speech recognition systems were well known before the priority date of the '277 Patent. For example, IBM's Tangora classifies acoustic input that "is then matched to words using discrete Hidden Markov Models of phoneme realization. A linguistic decoder then classifies the output of the acoustic front end producing a word for output." *Schmandt*, 64.

94. Another well-known speech recognition system was CMU's Sphinx. Sphinx "operates in nearly real time using hardware multiprocessor peripherals to aid in the search process." *Id.* Further, "Sphinx makes extensive use of Hidden Markov Models: each phone is represented by an HMM, each word is a network of phones, and the language is a network of words." *Schmandt*, 64-65.

95. Yet another well-known speech recognition system was MIT's SUMMIT. SUMMIT is a "phoneme-based connected speech recognizer[.]" *Schmandt*, 65. Transformed speech is segmented by "finding acoustic landmarks in the signal. Regions between the landmarks are grouped into segments on the basis of similarity. A set of phonetic classifiers assigns probabilistic phonetic labels to each of these portions of the speech signal." *Id.* Further, "[w]ords are represented by a network of phonemes; several networks may be used to encode alternate pronunciations. A pattern matching algorithm similar to dynamic programming matches the phoneme labels of the input against the word networks, deducing word boundaries in the process." *Id.*

VI. OVERVIEW OF THE '277 PATENT

96. I have reviewed the '277 *Patent* (Ex. 1001). Descriptions of the '277 Patent that may be pertinent to my analysis are provided herein, and my descriptions of the '277 Patent are intended only as a summary of the '277 Patent and/or intended as being pertinent to the Challenged Claims of the '277 Patent. My opinions provided herein are not intended as a discussion of my entire understanding of the '277 Patent or any other claims in any related patent to the '277 Patent. The '277 Patent describes a speech recognition system comprising separate processors for performing steps of the speech recognition process. '277 *Patent*, 5:11-12, 5:34-43, 29:16-17, Fig. 21, Claim 1. A first processor implements the feature vector

calculation (or front end) and search stage aspects of the speech recognition system (or circuit), and a second processor implements the distance (or probability) computation of the speech recognition system (or circuit). Data is pipelined from the front end, to the distance calculator, to the search stage. '277 Patent, Claims 1, 14-16.

VII. SUMMARY OF UNPATENTABILITY

97. In my opinion, it was known that speech recognition systems utilized pipelining and generally included three main steps: (1) extracting feature vectors from the audio signal (i.e., speech utterances); (2) computing a probability or likelihood that a particular feature vector is modeled by a particular subword unit (i.e., phoneme, allophone) that is associated with an acoustic model; and (3) performing a word identification process or search using a prefix tree to indicate the most likely spoken speech. I note that the '277 Patent's claims recite these three steps that were all well-known in the art. Pipelining said speech recognition steps and using multiple processors to perform said steps—and specifically performing the distance (or probability) computation on its own processor—was also well known. I note that even the '277 Patent admits likelihood (or probability or distance) determination is “computationally expensive.” '277 Patent, 1:38-46.

98. As I opine below, *Jiang* teaches the three well-known speech recognition steps recited in the independent claims. Specifically, *Jiang* teaches

feature vector extraction, distance (or likelihood or probability) calculation, and word identification—where feature vector extraction is performed by a first processor and distance calculation and word identification are performed by a “tree search engine 74” that may be its own dedicated processor. *Jiang*, 6:53-57 (disclosing feature extraction module 66 is a “conventional array processor”), 6:38-42 (disclosing tree search engine 74 may be a “dedicated speech recognition processor”). Thus, *Jiang* teaches two separate processors for performing the three well-known speech recognition steps. I note that *Jiang* does not expressly teach pipelining data or that the likelihood calculations are executed on a processor distinct from the processors perform feature vector extraction and a search stage for word identification. However, *Jiang* expressly suggests using multiprocessor systems and different hardware architectures, including expressly teaching its speech recognition system may comprise various computer configurations.

99. I note that *Baumgartner*, another prior art reference, expressly teaches performing the distance (or probability) computation step on a own processor distinct from a processor performing both the feature vector extraction and word identification.

100. I further note that *Brown*, another prior art reference, expressly teaches that data is pipelined through the three well-known speech recognition steps.

101. *Smyth* is another prior art reference that also expressly teaches performing the likelihood (or probability) calculations on its own processor.

VIII. OVERVIEW OF THE PRIOR ART

A. Overview of Jiang

102. *Jiang* teaches a speech recognition system having a plurality of acoustic models that performs feature vector extraction, distance/probability calculations, and word identification. *Jiang* (Ex. 1004), Abstract, 4:8-27, 4:51-5:3, 6:46-7:10, 7:30-8:51.

103. In my opinion, *Jiang* teaches a “system for recognizing speech based on an input data stream indicative of the speech [which] provides possible words represented by the input data stream as a prefix tree[,]” and therefore *Jiang* is in the same field of endeavor as the '277 Patent. *Compare, Jiang*, Abstract, 4:8-27 with '277 Patent, Abstract. Further, because *Jiang* is directed to a problem of performing speech recognition and *Jiang* teaches using various hardware configurations to perform speech recognition tasks, *Jiang* is reasonably pertinent to a problem faced by the inventors of the '277 Patent. *Compare Jiang*, 4:51–5:3, 6:46-7:10, 7:30-47 with '277 Patent, 1:4-14, 2:3-6.

B. Overview of Baumgartner

104. *Baumgartner* teaches accelerators for speech recognition systems. *Compare '277 Patent*, Abstract, Fig. 21, 30:40–65 with *Baumgartner*, Abstract, [0002], [0007]-[0010], Fig. 19. Therefore, in my opinion, *Baumgartner* is in the

same field of endeavor as the '277 Patent. Further, because *Baumgartner* is directed to a problem of distributing the computational load in a large vocabulary speech recognition system, *Baumgartner* is reasonably pertinent to a problem faced by the inventors of the '277 Patent. Compare '277 Patent, 1:39-46, 2:3-6 with *Baumgartner*, [0005].

C. Overview of Brown

105. *Brown* teaches speech recognition methods using pipelining. Compare *Brown*, Abstract, 1:9-11, 3:35-48 with '277 Patent, Abstract, 4:8-15, 30:41-42, Fig. 21. Therefore, in my opinion, *Brown* is in the same field of endeavor as the '277 Patent. Further, because *Brown* is related to a problem of reducing computation requirements in speech recognition systems, it is my opinion that *Brown* is reasonably pertinent to a problem faced by the inventors of the '277 Patent. Compare '277 Patent, 1:15-35 with *Brown*, 1:34-40.

D. Overview of Kazeroonian

106. *Kazeroonian* teaches processing data in speech recognizers. Compare '277 Patent, Abstract, 1:4-7, 2:41-44 with *Kazeroonian*, Abstract, 5:8-12, 6:31-33, 15:33–16:3. Therefore, in my opinion, *Kazeroonian* is in the same field of endeavor as the '277 Patent. Further, because *Kazeroonian* is related to a problem of improving data processing in speech recognition systems, it is my opinion that *Kazeroonian* is reasonably pertinent to a problem faced by the inventors of the '277

Patent. *Compare* '277 Patent, 2:41-44; 29:29-41 with *Kazeroonian*, Abstract, 5:8-12, 6:31-33, 15:33–16:3, 19:26–20:5, 21:9-11, Fig. 2.

E. Overview of Vensko

107. *Vensko* teaches a speech recognition system. *Compare* '277 Patent, Abstract, with *Vensko*, Abstract. Therefore, in my opinion, *Vensko* is in the same field of endeavor as the '277 Patent. Further, because *Vensko* is related to a problem of processor allocation in speech recognition systems—including the specific use of interrupts—it is my opinion that *Vensko* is reasonably pertinent to a problem faced by the inventors of the '277 patent. *Compare* '277 Patent, 1:4–2:6, 27:43-46 with *Vensko*, Abstract, 1:6-11.

F. Overview of Smyth

108. *Smyth* discloses a token-based speech recognition system with a recognition processor 3. *Smyth* (Ex. 1005), Abstract, 4:38-45, Fig. 1. *Smyth* discloses the various steps of speech recognition including a feature extractor extraction, distance (or probability or likelihood determination), and word search and identification. *Smyth* at generally 5:1-6:63, Fig. 2.

109. In my opinion, *Smyth* teaches a speech recognition system using processors to perform different portions of a speech recognition process. Therefore, *Smyth* “relates to speech recognition circuits and methods” and is thus in the same field of endeavor as the '277 Patent. *Compare* *Smyth*, 5:20-46, 5:58-6:4, 6:36-63

with '277 Patent, 1:4-5, 12:4-10, 12:15-50, 23:61-67. It is also my opinion that *Smyth* is reasonably pertinent to a problem faced by the inventors of the '277 Patent because both recognize challenges with performing speech recognition tasks while using a limited set of computing resources. Compare '277 Patent, 1:30-32, with *Smyth*, 2:40-42.

IX. OPINIONS REGARDING GROUND 1: CLAIMS 1, 5, 7, 12, AND 14-16 ARE OBVIOUS OVER *JIANG*, *BAUMGARTNER*, AND *BROWN*

A. Independent Claim 1

1. Claim 1(Pre): “A speech recognition circuit, comprising”

110. Though *Jiang* does not expressly use the word “circuit,” a POSITA would have recognized that *Jiang* teaches or at least renders obvious a circuit for performing speech recognition. *Jiang*, 1:15-19, 6:18-61, 4:51-5:3, Abstract, Figs. 1, 2, 7. *Jiang* describes in detail a number of computer-implemented structures and hardware components including a microphone, A/D converter, RAM, ROM, input and output devices, microprocessor-based electronics, PCs, ports, and program modules that would have comprised “circuitry” as understood by a POSITA. *Jiang*, 5:1-6:45, Figs. 1-2. Further, a POSITA would have found it obvious to implement the structures and computerized methods described by *Jiang* using “circuitry,” especially since *Jiang* expressly teaches its invention may be employed in a variety of “computer system configuration[s].” *Jiang*, 4:60-62.

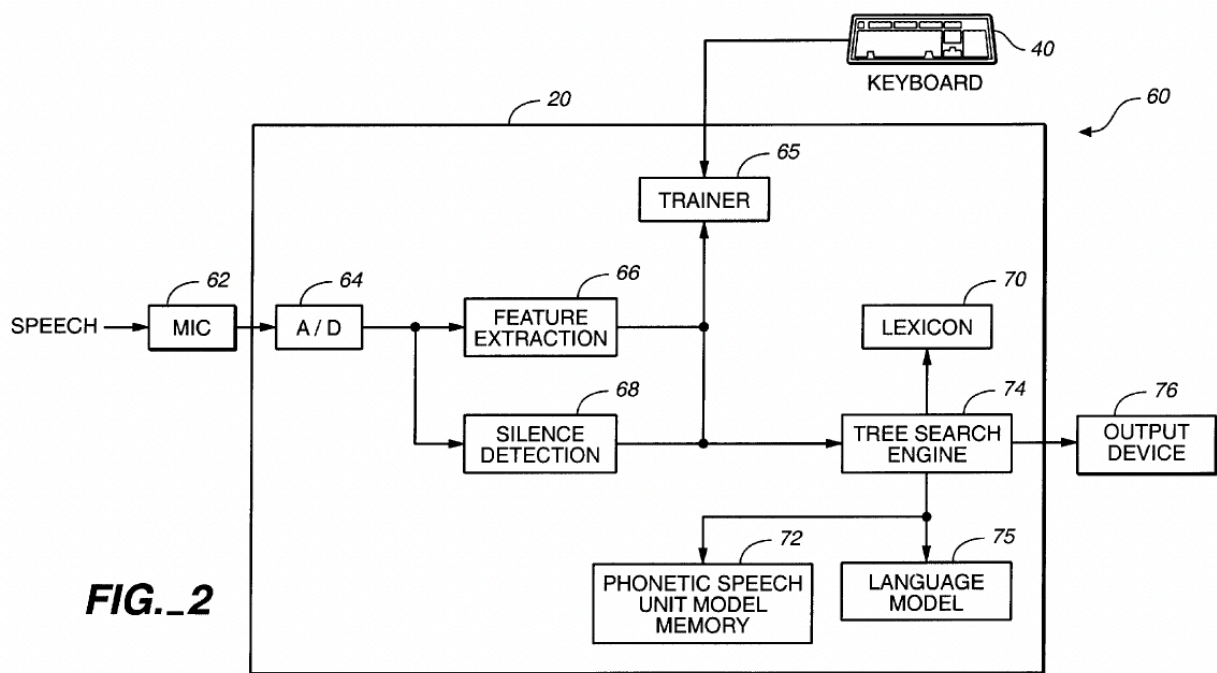
111. In my opinion, it would have been obvious to a POSITA to implement the computerized systems and methods taught by *Jiang*, to the extent not otherwise taught, using “circuitry,” as recited in Claim 1. I note the ’277 Patent does not appear to ascribe any structural features or requirement to the claimed “circuit.” A POSITA would have recognized that computers are electronic devices comprising at least one or more computer processors, memory, a power source, and electrical connections therebetween and operating on electricity, thus teaching or at least rendering obvious “circuitry” performing the computerized steps taught by *Jiang* (and/or taught by *Baumgartner*, *Brown*, *Kazeroonian*, *Vensko*, and *Smyth*, as discussed for the collective claims of the ’277 Patent). To the extent any particular citation from one of these references describes computerized methods stored as software, a POSITA would have physically implemented those computerized methods using electronic circuitry such as a processor core and associated memory. Thus, in my opinion, a POSITA would have found it obvious and been motivated to embody the various electronic components disclosed in the references as “circuitry” without undue experimentation.

2. *Claim 1(a): “an audio front end for calculating a feature vector from an audio signal,”*

112. *Jiang* teaches or renders obvious “an audio front end for calculating a feature vector from an audio signal.” As I explain in detail below, *Jiang* teaches

feature extraction module 66 where feature extraction module 66 calculates a feature vector from an audio signal. *Jiang*, 1:58-67, 6:46-7:10.

113. Regarding the claimed “audio signal,” *Jiang* teaches a speech recognition system 60 including a microphone 62 and analog-to-digital (A/D) converter 64. *Jiang*, 6:18-21, 6:46-7:1. Speech recognition system 60 is depicted in Figure 2 below.



Jiang, Fig. 2; see also *Jiang*, Fig. 7.

114. *Jiang* teaches inputting speech into system 60 “in the form of an audible voice signal provided by the user to microphone 62. Microphone 62 converts the audible speech signal into an analog electronic signal which is provided to A/D converter 64. A/D converter 64 converts the analog speech signal into a sequence of

digital signals which is provided to feature extraction module 66.” *Jiang*, 6:46-53. *Jiang* further teaches the “signals are, in one preferred embodiment, provided to feature extraction module 66 by A/D converter 64 at a sample rate of approximately 16 kHz, implementing A/D converter 64 as a commercially available, well known A/D converter.” *Jiang*, 6:57-61. Additionally, *Jiang* teaches “[f]eature extraction module 66 divides the digital signal received from A/D converter 64 into frames which include a plurality of digital samples. Each frame is approximately 10 milliseconds in duration.” *Jiang*, 6:62-65, 7:16-17 (“As feature extraction module 66 is processing the digital samples from A/D converter 64[.]”), 7:23-24 (“silence detection module 68 processes the digital samples provided by A/D converter 64[.]”), 11:25-29. *Jiang* further discloses frames are sequential time intervals during which an acoustic signal is sampled.

When actually processing an acoustic signal, the signal is typically sampled in sequential time intervals called frames. The frames typically include a plurality of samples and may overlap or be contiguous. Each frame is associated with a unique portion of the speech signal. The portion of the speech signal represented by each frame is analyzed to provide a corresponding acoustic vectors. During speech recognition, a search is performed for the state sequence most likely to be associated with the sequence of acoustic vectors.

Jiang, 1:58-67. Therefore, in my opinion, *Jiang* teaches inputting speech as an analog signal into system 60 via microphone 62 and converting said analog signal to digital samples via A/D converter 64. Thus, *Jiang* teaches the claimed “audio signal.”

115. Regarding the claimed “audio front end,” in my opinion, *Jiang*’s feature extraction module 66 teaches the claimed “audio front end.” *Jiang*, 1:58-67, 6:46–7:10, Fig. 2. *Jiang* teaches A/D converter 64 converts the analog speech signal into a sequence of digital signals “which is provided to feature extraction module 66.” *Jiang*, 6:50-53. *Jiang* teaches feature extraction module 66 “performs spectral analysis on the digital signals and computes a magnitude value for each frequency band of a frequency spectrum. The signals are, in one preferred embodiment, provided to feature extraction module 66 by A/D converter 64 at a sample rate of approximately 16 kHz[.]” *Jiang*, 6:54-59. In detail, *Jiang* teaches feature extraction module 66 encodes frames that include a plurality of digital samples into feature vectors, and “also preferably encodes the feature vectors into one or more codewords using vector quantization techniques and a codebook derived from training data.”

Feature extraction module 66 divides the digital signal received from A/D converter 64 into **frames** which include **a plurality of digital samples**. Each frame is approximately 10 milliseconds in duration. The frames are then preferably **encoded by feature extraction module**

66 into a feature vector reflecting the spectral characteristics for a plurality of frequency bands. In the case of discrete and semi-continuous hidden Markov modeling, feature extraction module 66 also preferably encodes the feature vectors into one or more **codewords** using vector quantization techniques and a codebook derived from training data. Thus, **feature extraction module 66** provides, at its output the **feature vectors (or codewords)** for each spoken utterance. The feature extraction module 66 preferably provides the feature vectors (or codewords) at a rate of one codeword approximately every 10 milliseconds.

Jiang, 6:62-7:10.

116. In my opinion, *Jiang*'s teachings of encoding frames of digital audio samples into feature vectors reflecting the spectral characteristics for a plurality of frequency bands teaches or at least renders obvious "for calculating a feature vector," as claimed. In particular, *Jiang* teaches a feature vector is encoded, which a POSITA would have understood to mean "calculate[ed]," as claimed. I note that the '277 Patent does not provide express descriptions of steps "for calculating a feature vector" by the audio front end, nor do the claims recite anything more than "an audio front end for calculating a feature vector from an audio signal." See '277 Patent, 15:6-7 ("At the start of each time frame, the audio input is **converted** into a feature vector"), 23:17-18 ("At the start of the time frame, the feature vector is **generated**"),

12:4-10 (“For each audio input time frame, the audio signal is converted into a feature vector. This may be done by splitting the audio signal into spectral components.”). In my opinion, it was well known to a skilled artisan to use an audio front end for calculating (i.e., generating, converting, encoding) a feature vector from an audio signal, and *Jiang*’s teachings of feature extraction module 66 which encodes a feature vector from a digital audio signal provided by A/D converter 64 teaches or at least renders obvious Claim 1(a).

117. In my opinion, *Jiang*’s feature extraction module 66 teaches, or at least renders obvious, the claimed “audio front end” because *Jiang*’s feature extraction module 66 receives a digital audio signal from *Jiang*’s A/D converter and encodes said digital audio signal into a feature vector. *Jiang*, 6:46–7:1.

3. *Claim 1(b): “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”*

a) The ’277 Patent’s Discussion of “extracted and/or derived quantities”

118. In my opinion, a POSITA would have understood that the claimed “plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame” would have included spectral components of the feature vector and the component derivatives. For example, the ’277 Patent describes converting an audio signal into a feature vector by “splitting the audio signal into spectral components”:

For each audio input time frame, the audio signal is converted into a feature vector. This may be done by splitting the audio signal into spectral components. For example, in the present embodiment, the audio input signal is split into 12 frequency bands and one energy band, and a feature vector is constructed using these 13 components, plus their first and second derivatives, a total of 39 components.

'277 Patent, 12:4-10.

119. Further, a POSITA would have understood that Claim 1(b) references the *'277 Patent*'s description of the spectral components of the feature vector and the component derivatives and that cepstral coefficients are spectral components of an audio signal that indicate extracted and derived quantities. As I discussed above, it was well known that components of feature vectors included spectral components such as cepstral coefficients. *See* V.B (including my discussion of *Ittycheriah '447*, which, like the *'277 Patent*, discloses thirty-nine dimension feature vectors. "Such feature vectors and others, such as LPC cepstra, are thirty-nine dimension vectors ...[.]" *Ittycheriah '447*, 4:54-67; *'277 Patent*, 12:35-37). Thus, in my opinion, cepstral coefficients are a well-known representation of spectral components (e.g., signal frequency, energy) that are extracted and derived from an audio signal. *See* V.A (discussing spectrograms), V.B (discussing feature vectors). Therefore, a

POSTIA would have understood it was well-known that audio signals included spectral components or characteristics split among frequency bands.

b) *Jiang's Teachings*

120. In my opinion, *Jiang* teaches or renders obvious “wherein the feature vector comprises a plurality of extracted and/or derived quantities,” as claimed. *Jiang* teaches a feature vector “reflect[s] the spectral characteristics for a plurality of frequency bands.” *Jiang*, 6:22–7:10 (“The frames are then preferably encoded by feature extraction module 66 into **a feature vector reflecting the spectral characteristics for a plurality of frequency bands**”), 6:53-57 (“In a preferred embodiment, feature extraction module 66 is a conventional array processor which performs spectral analysis on the digital signals and computes a magnitude value for **each frequency band** of a frequency spectrum.”). I note the ’277 Patent also describes “the audio signal is converted into a feature vector. This may be done by splitting the audio signal into spectral components.” ’277 Patent, 12:4-10. Therefore, in my opinion, *Jiang* teaches feature extraction module 66 (i.e., the claimed “audio front end”) extracts quantities from a digital audio signal where the quantities are indicative of the spectral characteristics of said digital audio signal. Further, *Jiang* teaches such quantities are encoded into a feature vector.

121. In my opinion, it was also well known that speech input was commonly captured as a series of audio samples that were often 10 milliseconds each in

duration. *Ravishankar*, 28 (“Frames are typically spaced at 10msec intervals”); *Bahl* (Ex. 1052), 16:41-46 (“taken at preferably 10 msec intervals.”); *Nguyen* (Ex. 1047), 7:39-40 (“frame time (e.g., 10 milliseconds)”), 7:9-10; *Baumgartner* (Ex. 1007), [0205] (“the observation frequency (e.g., every 10 ms)”); *Smyth* (Ex. 1005), 1:24-31 (“segmented into frames of length 10-20 ms”); *Vensko* (Ex. 1038), 4:40-51 (“10 milliseconds to A/D convert[.]”), 4:60-5:3 (“each frame corresponding to 10 milliseconds[.]”); *Ney* (Ex. 1053), 7 (“10-ms time frame.”), 11 (“every 10 ms of the input speech[.]”).

122. It was further known that spectral components represent how fast an audio signal is changing over time. Further, it was known that Fast Fourier Transforms (FFTs) were used to extract metrics (i.e., the claimed “quantities”) from “raw” signal changes as evidenced by the spectral components. *Baumgartner*, [0037] (“Those skilled in the art will realize that, to convert the speech signal 105 into frames 115, the pre-processing block 110 can contain such devices as an analog-to-digital converter, a window/filter, and a Fast Fourier transform.”).

123. In my opinion, *Jiang* teaches, or at least renders obvious, feature vectors that comprise “extracted and/or derived quantities from said audio signal during a defined audio time frame,” as claimed. For example, *Jiang* teaches feature extraction module 66 divides the digital audio signal received from A/D converter 64 into frames that include a plurality of digital samples and where each frame is

“approximately 10 milliseconds in duration.” *Jiang*, 6:62-65. A POSTIA would have understood that 10 milliseconds in duration was a “defined audio time frame,” as claimed. Feature extraction module 66 (i.e., the claimed “audio front end”) encodes the frames “into a feature vector reflecting the spectral characteristics for a plurality of frequency bands.” *Jiang*, 6:54–7:1. Therefore, *Jiang*’s feature vectors are encoded from frames representing 10 milliseconds of audio speech input, as was well-known in the art as I discussed above.

4. Claim 1(c): “a calculating circuit for calculating distances indicating the similarity between a feature vector and a plurality of predetermined acoustic states of an acoustic model; and”

124. In my opinion, *Jiang* alone or in combination with *Baumgartner* teaches or renders obvious Claim 1(c). *Jiang* teaches or render obvious a “calculating circuit,” as claimed, in addition to teaching the claimed functionality of the “calculating distances” limitation. *Jiang* teaches tree search engine 74 performs both the functionality of Claim 1(c), i.e., “calculating distances indicating the similarity ...,” and the functionality of Claim 1(d), i.e., “identify[ing] words within a lexical tree” *Jiang*’s tree search engine 74 may be a dedicated speech recognition processor. *Jiang*, 6:39-42. The proposed mapping for Claim 1(c) relies on the functionality performed by *Jiang*’s tree search engine 74 but further proposes modifying *Jiang*’s tree search engine to be on separate processors, where the functionalities of Claims 1(a)-1(b) (feature vector calculation) and 1(d) (word

identification) are implemented on a first processor, and the functionality of Claim 1(c) (distance calculation) is implemented on a second processor, as claimed by Claim 1(e) and taught by *Baumgartner*. *Jiang* already contemplates using multiple processors in its speech recognition system. *Jiang*, 4:60–5:3 (“those skilled in the art will appreciate that the invention may be practiced with other computer system configurations ...”), 6:31-45. Accordingly, and as further discussed in Section IX.A.6.f, a POSITA would have been motivated and found it obvious to modify *Jiang*’s tree search engine such that the functionality of Claim 1(c) is implemented on its own processor.

125. To the extent argued *Jiang* does not teach a “calculating circuit,” *Baumgartner* teaches a calculating circuit for determining probabilities. As discussed in Section IX.A.6, *Baumgartner*’s SLA processor (where the “heart of the SLA is a circuit”) teaches the claimed “calculating circuit [] implemented using a second processor,” as claimed in Claim 1(e). It would have been obvious to a POSITA to modify *Jiang*’s tree search engine 74 to perform the distance calculations using a calculating circuit on a second processor. *See* Section IX.A.6.f. The claimed calculating circuit implemented using a second processor as recited in Claim 1(e) relies on *Baumgartner*’s SLA processor circuit for performing distance calculations, as discussed further below.

a) The '277 Patent's Discussion of "distances"

126. I note the '277 Patent describes "distances" in a broad sense and encompasses "probabilities," "likelihoods," and "probability distributions." '277 *Patent*, 3:51-56, 3:57-60 ("There are a wide variety of probability distributions that can be used for the distance calculation state of a speech recogniser, and a wide variety of distance metrics used. These are widely documented in the literature."), Fig. 16. The '277 Patent additionally expressly describes that the "distances" are not limited to either a probability or a likelihood. '277 *Patent*, 4:13-17 ("In this context a distance is an indication of the probability or likelihood that a feature vector corresponds to a particular state. An important class of distance computation speech recognition is the calculation of output state probabilities and recognizers using Hidden Markov Models."), 12:38-49 ("The distance calculator stage of the recognition process computes a probability or likelihood that a feature vector corresponds to a particular state."). The '277 Patent explains that "[a] common choice is to use Gaussian Distributions and correspondingly the Mahalanobis Distance [MHD] metric" for calculating distances. '277 *Patent*, 3:62-63, 12:51-53, 23:58-24:5 (stating other types of distance measurements other than MHD may be used), 12:11-14 (referring to the MHD engine as computing the "likelihood" for each state); 12:51-13:4 (describing the MHD).

127. I note that the '277 Patent intimates a distinction between a “probability” versus a “likelihood”: “In alternative embodiments, the likelihoods rather than probabilities are computed as it leads to simpler implementations of computations in the search stage.” *'277 Patent*, 13:2-4. Additionally, the '277 Patent states: “It is preferable to use likelihoods rather than probabilities, because likelihoods can be added together to give a total likelihood. Probabilities must be multiplied, which can give rise to rounding errors.” *'277 Patent*, 13:27-31. In my opinion, the '277 Patent draws a mathematical distinction between a probability and a likelihood. However, I understand from these passages and other collective discussions in the '277 Patent that regardless of any mathematical distinction between a probability and a likelihood, both measurements are a similarity measurement, thus describing the claimed “distances.”

128. In my opinion, the '277 Patent's description of “distance” as a probability or likelihood of a feature vector compared to a particular state of an HMM is consistent with well-known use of HMMs in speech recognition. *See* Section V.D-E; *Ravishankar*, 28-29. In the well-known context, HMM represent states. A distance (e.g., likelihood, probability) is then computed for each state of an HMM and the distance indicates the similarity between the uttered speech and the state sequence or path. In my opinion, a POSITA would have understood that the probability reflects a “distance” or similarity. Specifically a low probability indicates

a larger distance (meaning the feature vector is not likely the HMM state it is being compared to) and a high probability indicates a smaller distance (meaning the feature vector is likely the HMM state it is being compared to). Accordingly, a POSITA would have understood that a probability or likelihood reflects a “distance” or similarity.

b) *Jiang*’s Teachings

(1) “a calculating circuit”

129. In my opinion, *Jiang* teaches or renders obvious a “calculating circuit,” as claimed. *Jiang* teaches a tree search engine 74 for performing both the functionality of Claim 1(c) (i.e., calculating distance) as well as the functionality of Claim 1(d) (i.e., using said calculated distances to identify words). *Jiang* teaches tree search engine 74 “is preferably implemented in CPU 21 (which may include one or more processors) or maybe performed by a dedicated speech recognition processor employed by personal computer 20.” *Jiang*, 6:39-42.

130. Because *Jiang* teaches tree search engine 74 may be a dedicated processor, it is my opinion that *Jiang*’s tree search engine 74 is, includes, or renders obvious a circuit. I again note the ’277 Patent does not ascribe any specific structural features or requirements to the claimed “circuit.” A POSITA would have understood that a “circuit” is an aspect of a processor used to perform tasks by carrying electrical current through paths. See *Microsoft Computer Dictionary* (Ex. 1056), 6 (a “circuit”

is “1. Any path that can carry electrical current. 2. A combination of electrical components interconnected to perform a particular task. At one level, a computer consists of a single circuit; at another, it consists of hundreds of interconnected circuits.”), 8 (citing to “central processing unit” for the definition of a “processor”), 5 (citing to “CPU” for the definition of “central processing unit”), 7 (a “CPU” is “The computational and control unit of a computer. The CPU is the device that interprets and executes instructions. Mainframes and early minicomputers contained circuit boards full of integrated circuits that implemented the CPUs”); *Dictionary of Electrical and Computer Engineering* (Ex. 1055), 6, 7. Accordingly, a POSITA would have understood or found obvious that the claimed “circuit” would have been met by a processor disclosure.

131. A POSITA would have further understood that *Jiang* teaches a speech recognition “circuit” through *Jiang*’s various disclosures of electrical components performing executing speech recognition on electronic hardware, as explained above in Section IX.A.1. Because *Jiang* teaches a speech recognition “circuit,” contemplates various hardware architectures, and specifically teaches certain speech recognition tasks may be performed on distinct processors, a POSTIA would have understood *Jiang* teaches a “calculating circuit,” as claimed.

(2) **“a plurality of predetermined acoustic states of an acoustic model”**

132. I note the '277 Patent describes examples of acoustic states, “including states corresponding to [a] phone or group of phones.” '277 Patent, 6:63-64, 7:56-58. The '277 Patent further describes acoustic models include HMMs and “alternative types of acoustic models, rather than HMM models.” '277 Patent, 37:53-55.

133. Regarding the claimed “acoustic model,” in my opinion, *Jiang*’s teachings of HMMs teach the claimed plurality of predetermined acoustic states of “an acoustic model.” *Jiang* teaches “[m]emory 72 stores phonetic speech unit models, such as hidden Markov models, which represent speech units to be detected by system 60.” *Jiang*, 7:33-36, 2:53-56 (“In such a prefix tree, each subword unit (such as a phoneme) is typically represented by a branch which is associated with a particular phonetic model (such as an HMM).”). A POSITA would have understood that the phonetic speech unit models referenced by *Jiang* are “acoustic model[s],” as claimed. As I explained above in Section V.D, HMMs were extremely well known to be acoustic models. *See* Section V.D; *see also* *Ravishankar*, 19-20, 27-28, 41. HMMs as acoustic models is consistent with how *Jiang* describes HMMs. *Jiang*, 1:19-57, 2:53-56. A POSITA would have also understood that *Jiang*’s reference to phonetic speech teaches that the claimed models are “acoustic,” as speech input represents audio (or acoustic) signals. *Jiang*, 1:19-57, 2:53-56. Further, *Jiang*’s

teachings of HMMs are consistent with the '277 Patent, which describes HMMs as acoustic models. *Jiang* repeatedly teaches utilizing HMMs for speech recognition. '277 Patent, 3:4-5, 10:56-67.

134. Regarding the claimed “plurality of predetermined acoustic states” of an acoustic model, in my opinion *Jiang* teaches each HMM includes a plurality of predetermined acoustic states. *Jiang*, 1:21-22 (“A hidden Markov model includes a plurality of states”), 1:43-48 (“HMMs have been employed to model observed sequences of speech spectra, where specific spectra are probabilistically associated with a state in an HMM. In other words, for a given observed sequence of speech spectra, there is a most likely sequence of states in a corresponding HMM.”), 1:49-57 (further describing states and associating states with a sub-word unit).

135. As I explained above, it was well known that HMMs include a plurality of predetermined states. See Section V.D; see also *Ravishankar*, 19-20, 27-28, (“An HMM is a set of states connected by transitions (see Figure 3.2 for an example). Transitions model the emission of one frame of speech.”), 41. A POSITA would have understood that the speech recognition taught in *Jiang*, 7:30-47 employs HMMs and teaches the HMMs include a plurality of predetermined acoustic states. For example, *Jiang* provides a discussion of speech recognition. *Jiang*, 7:30–8:51. At the end of this cited discussion, *Jiang* expressly teaches “[t]his is all done in a known manner.” *Jiang*, 8:51. This known manner includes the discussion in *Jiang*’s

Background section that expressly teaches each HMM has a plurality of predetermined acoustic states. *Jiang*, 1:15–2:15, 8:51. *Jiang*’s “known manner” where HMM contain a plurality of predetermined states in consistent with well-known nature of HMMs as I explained above. *See* Section V.D.

136. The plurality of acoustic states is “predetermined” because HMMs are generated and stored in system memory prior to the claimed distance calculation. A POSITA would have understood that HMMs in many cases are generated in a training or learning phase and therefore exist before the three well-known speech recognition steps occur to actually determine speech (as opposed to training the system). *See* Section V.D (quoting *Lee*, 1:24-38). For example, *Jiang* provides a discussion explaining how the HMMs are created by training module 65:

In order to train the system shown in FIG. 2, training words are spoken into system 60 by the user. The training words comprise a set of training data which is converted to digital samples by A/D converter 64 and then to codewords by feature extraction module 66. The codewords (or output probability distributions) are provided to training module 65. Training module 65 also receives a phonetic transcription of each training word from the user by a user input device, such as keyboard 40. The training module 65 can be implemented on either the same processor, or a separate processor, from the remainder of the speech recognition system. One or more

HMMs are created by training module 65 for each phoneme of each word in a desired vocabulary represented by the training data. The HMMs thus represent model output distributions associated with the phonemes in the vocabulary. The prefix tree is then formed, based upon the desired vocabulary, such that phonemes are structured to provide a leaf which is associated with each word in the desired vocabulary. It should also be noted that the training words (or data set) can either be received one word at a time from the microphone, as described above, or input, in its entirety by a conventional computer input device, such as a floppy disk which contains a previously produced data set.

Jiang, 11:25-47; *see also Jiang*, 7:30-47 (explaining how HMMs are stored in system memory). It was well known that HMMs are determined prior to distance calculations (and thus “predetermined”) because HMMs exist in the system prior to distance (or probability or likelihood or similarity) calculations/determinations. *See* Section V.D; *see also Ravishankar*, 19-20, 27-28).

(3) “a feature vector”

137. As I explained above, it is my opinion that *Jiang* teaches encoding the frames of spoken speech into a feature vector. *See* Section IX.A.2-IX.A.3. For example, *Jiang* teaches a feature extraction module 66 that receives frames from

A/D converter 64 and then module 66 encodes said frames “into a feature vector reflecting the spectral characteristics for a plurality of frequency bands.”

Feature extraction module 66 divides the digital signal received from A/D converter 64 into frames which include a plurality of digital samples. Each frame is approximately 10 milliseconds in duration. The frames are then preferably encoded by feature extraction module 66 into a feature vector reflecting the spectral characteristics for a plurality of frequency bands. In the case of discrete and semi-continuous hidden Markov modeling, feature extraction module 66 also preferably encodes the feature vectors into one or more codewords using vector quantization techniques and a codebook derived from training data. Thus, feature extraction module 66 provides, at its output the feature vectors (or codewords) for each spoken utterance. The feature extraction module 66 preferably provides the feature vectors (or codewords) at a rate of one codeword approximately every 10 milliseconds.

Jiang, 6:62-7:10.

138. I also note that *Jiang* teaches an optional additional step involving vector quantization. *Jiang*, 7:1-10. In my opinion, vector quantization was a well-known technique that clusters feature vectors into similar codewords to reduce the size of the search space. For example, *Brown* discloses such.

In particular, feature extractor 105 derives a set of LPC parameters from an incoming speech signal or speech input which defines the identifying characteristics of the speech input. The incoming speech signal is sampled in a sequence of overlapping frames during a corresponding sequence of frame periods. Feature extractor 105 thereupon produces so-called cepstral features from the LPC parameters which represent the frame being processed. Illustratively, 12 cepstral and 12 delta cepstral features are produced. **Vector Quantizer (VQ) 110 is an optional stage which is included in the recognition system 100 to reduce computation requirements. Specifically, VQ 110 represents the various cepstral features in an efficient manner.**

Brown (Ex. 1036), 10:24-34, 3:53-65. *Ravishankar* also discloses such. *Ravishankar*, 48-50 (“VQ stands for *vector quantization*. In this function, the Gaussian densities that make up each feature codebook are evaluated at the input feature vectors. In other words, we compute the Mahalanobis distance of the input feature vector from the mean of each Gaussian density function.”). I also note that a codeword is an index into a codebook and a codebook stores a value representative of a cluster of data in the original form such as a representative feature vector. *See* ’277 *Patent*, 35:11-12, 4:46-48.

139. In my opinion, *Jiang* teaches speech recognition that determines one or more phonemes represented by codewords. *Jiang*, 7:38-43. It is also my opinion that a skilled artisan would have understood that since codewords are feature vectors that have undergone vector quantization, *Jiang*'s teachings of determining the phonemes of codewords also teaches or at least renders obvious determining the phonemes of feature vectors because codewords are encoded feature vectors.

(4) "calculating distances ..."

140. Regarding the claimed "calculating distances ...," in my opinion, *Jiang* teaches a distance calculation that utilizes HMMs. *Jiang*, 7:30-47. Indeed, *Jiang* discloses that the speech recognition process at 7:30-8:51 is done in a known manner. A POSITA would have understood that *Jiang*'s distance calculation determines the similarity between a feature vector and a plurality of states of an acoustic model (for example, the states of one or more HMMs). *Jiang*, 7:30-47. I discuss such in more detail below and note that this is consistent with the well-known manner referenced by *Jiang*. See Section V.E.

141. *Jiang* teaches determining a score for each phone branch of a prefix tree based on determining a most likely phoneme for a particular codeword/feature vector in comparison to a plurality of predetermined acoustic states of at least one HMM (i.e. the acoustic model). *Jiang*, 8:29-33. Specifically, *Jiang* teaches:

**Upon receiving the codewords from feature extraction
module 66, and the boundary detection signal provided by**

silence detection module 68, tree search engine 74 accesses information stored in the phonetic speech unit model memory 72. **Memory 72 stores phonetic speech unit models, such as hidden Markov models, which represent speech units to be detected by system 60.** In one preferred embodiment, the phonetic models stored in memory 72 **include HMMs which represent phonemes. Based upon the HMMs stored in memory 72, tree search engine 74 determines a most likely phoneme represented by the codeword received from feature extraction module 66, and hence representative of the utterance received by the user of the system.**

Jiang, 7:30-47.

142. In particular, I note that a POSITA would have understood from *Jiang*'s teaching at 7:45-46 that "a phonetic HMM tree search can be performed in order to find the proper phoneme" indicates a phonetic HMM may be used to identify the most likely phoneme.

143. I note that *Jiang* discusses two different "scores" in describing the word identification via the prefix tree. The first score is discussed at *Jiang*, 8:21-37 and is "indicative of a likelihood that the particular phoneme being examined is the actual phoneme indicated by the codeword received from the feature extraction module." This score is the distance indicating the similarity between a feature vector and a plurality of predetermined acoustic states of an acoustic model, as I discuss above in

Section IX.A.4.b.2. However, *Jiang* discusses a second score that is based on the first score and any scores calculated prior in the prefix tree. In particular, *Jiang* discusses that as the tree is traversed, a score is assigned to each node. This second score is “based on the likelihood that the present codeword (output probability distributions) under analysis is represented by the phoneme corresponding to the branch in tree 77 then being considered, and based on the score [i.e., the first score referenced at 8:16-37] assigned to nodes further up the tree which are connected by phoneme branches to the present node.” *Jiang*, 8:44-51. Because *Jiang* expressly states the second score is based on both the likelihood of the present codeword under analysis, i.e., the codeword associated with the node currently being scored, and based on the scores of prior nodes, in my opinion a POSITA would have reasonably understood *Jiang* is referring to two different scores. Indeed, such a scoring method for lexical trees is well known, as *Jiang* states at 8:51. It was well understood that as the node is traversed, later node scores are dependent on what is occurring at the present node and what occurred in prior nodes. Finally, I note that *Jiang* does not refer to the score discussed at 8:16-37 as a first score or the score discussed at 8:44-51 as a second score, but I use these indicators for ease of reference and to differentiate between the two differently-calculated scores.

144. I further note that *Ravishankar* teaches “[o]rganizing the HMMs to be searched as a phonetic tree instead of the flat structure of independent linear HMM

sequences for each word is probably the most often cited improvement in search techniques in use currently.” *Ravishankar*, 33. I note that *Jiang*’s teachings of traversing a tree and determining a score for each node and related phoneme branch is similar to *Ravishankar*’s teachings. Specifically, *Jiang* teaches that using HMMs (phonetic speech unit models), the likelihood of a phoneme is computed and “a word which most likely represents the codewords received by feature extraction module” is determined. *Jiang*, 7:30-47, 7:49-54. In my opinion, *Jiang* teaches speech is input into a speech recognition system, the speech is sampled, feature vectors are extracted, and a tree search engine traverses a lexical tree, namely a prefix tree, and a score is calculated for a codeword based on said HMMs.

145. It is also my opinion that *Jiang* teaches a “distance” that is referred to as a “score” or a “probability score.” *Jiang*, 1:19-2:15. The Background section in *Jiang* provides a discussion of the well-known HMMs to calculate a distance indicating the similarity between a feature vector and a plurality of predetermined HMM states. *Jiang*, 1:19-2:15. For example, *Jiang* teaches utilizing HMMs to “model observed sequences of speech spectra, where specific spectra are probabilistically associated with a state in an HMM.” *Jiang*, 1:41-48. *Jiang* also teaches a “probability score is computed for each state in the state sequences (i.e., the HMMs) being considered.” *Jiang*, 2:5-7. A POSITA would have understood *Jiang*’s collective teachings indicate that *Jiang*’s “score [] indicative of a likelihood

that the particular phoneme being examined is the actual phoneme indicated by the codeword received from the feature extraction module” is a “distance” as recited in Claim 1(c). *Jiang*, 8:20-23.

146. A POSITA would have also understood *Jiang*’s Background indicates the calculated “score” is a probability or likelihood representative of a similarity or distance measure. *Jiang*, 1:15–2:15. For example, *Jiang*’s Background describes the HMMs as “probabilistic models” and expressly states “[a]n observation is probabilistically associated with each unique state.” *Jiang*, 1:20-25.

147. A POSITA would have understood *Jiang*’s teaching of calculating a probability score for each state sequence teaches or renders obvious calculating at least one distance. *Jiang* uses the term “probability score” to refer to a measurement that indicates the similarity or likelihood of the feature vectors of an utterance to the state, namely “representative of the utterance received by the user of the system.” *Jiang*, 7:41-43. “The highest probability score provides the most likely state sequence for the entire utterance,” which is then converted into the phoneme, word, or word sequence, as applicable. *Jiang*, 2:10-15. In my opinion, because the probability score scores the likelihood the state sequence is representative of the utterance, the probability score is a distance indicating the similarity between the feature vector (or the codeword that encodes the feature vector) and the

predetermined acoustic states of the HMM (i.e., the state sequence of the HMM). Therefore, *Jiang* teaches the claimed “calculating distances.”

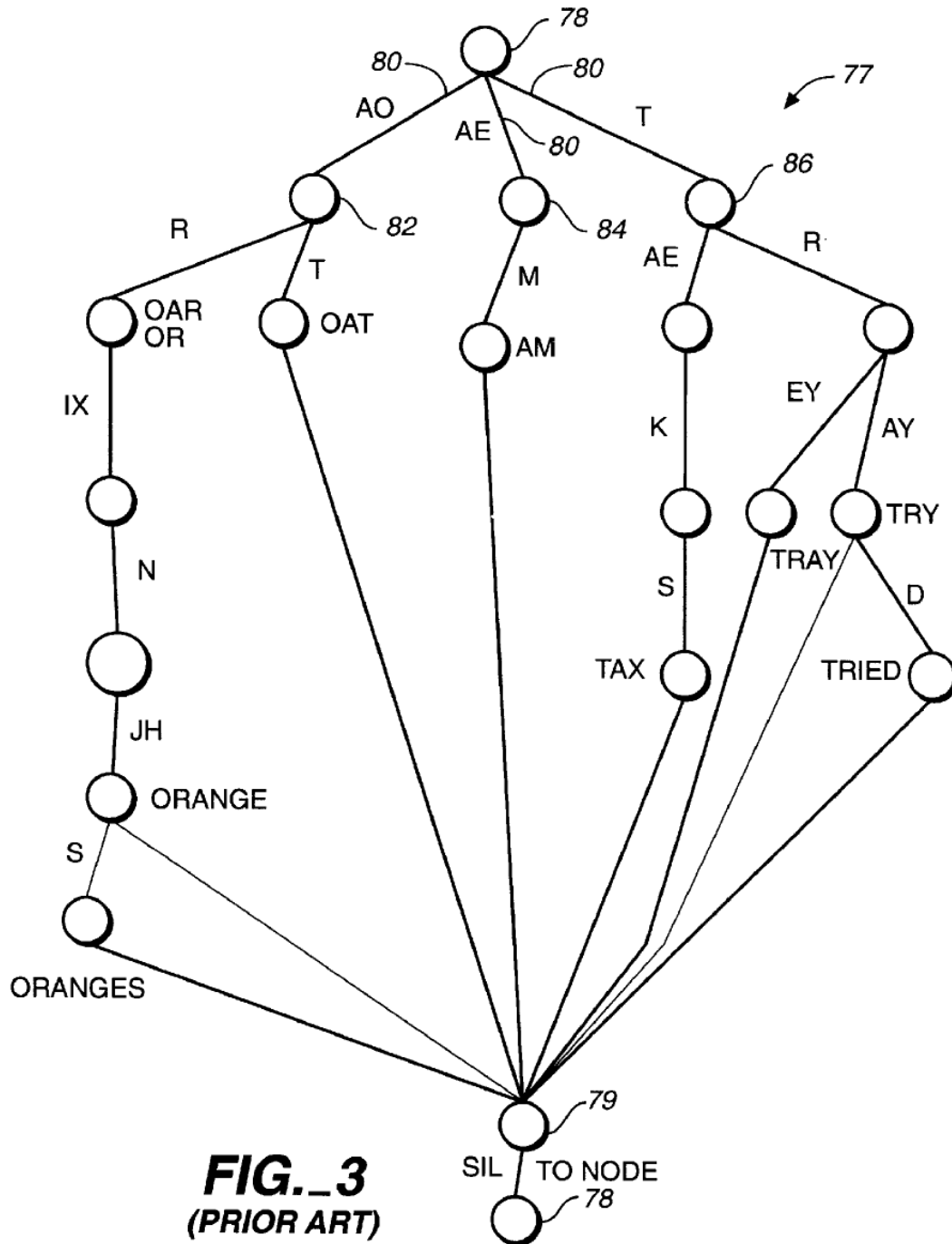
148. Further, in my opinion, *Jiang*’s teachings of determining the likelihood that a codeword is a particular phoneme teaches calculating distances that indicate the similarity between a feature vector and HMMs states. *See Jiang*, 6:64-7:10, 1:58-67, 7:38-43, 8:29-36. I note that because *Jiang* teaches determining the likelihood of a given phoneme is indicated by a codeword, *Jiang* also teaches determining the likelihood of the given phoneme is indicated by at least one feature vector. *See* Section IX.A.2 regarding my opinions of Claim 1(a). Because *Jiang* teaches computing a probability score for each state of an HMM, it is my opinion that *Jiang* teaches computing a probability score for a plurality of states of an acoustic model such as an HMM, as claimed. *See Jiang*, 7:37-42. It is also my opinion that because *Jiang* teaches HMMs are used for determining a most likely phoneme from a feature vector, *Jiang* teaches the calculated score indicates the probability (i.e., likelihood, similarity) between states of an HMM and a feature vector/codeword.

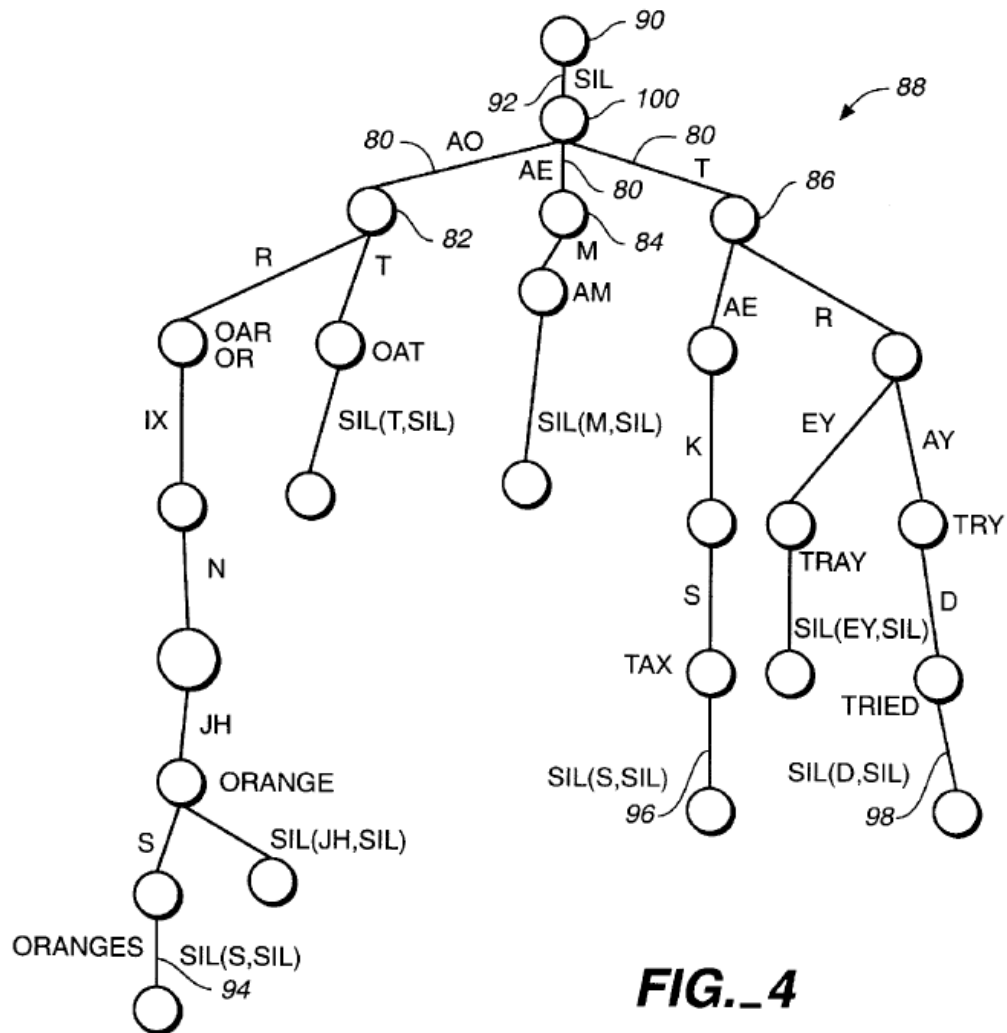
149. It is also my opinion that *Jiang* teaches or at least renders obvious calculating a plurality of distances. For example, *Jiang* teaches determining the similarity between the feature vector (or codeword) and "a plurality of predetermined acoustic states of an acoustic model," as claimed. *Jiang*, 7:30-45, 8:16-51. As mentioned above, *Jiang* teaches a “probability score is computed for

each state in the state sequences (i.e., the HMMs) being considered.” *Jiang*, 2:5-7. Because *Jiang* teaches computing a probability score for each state of an HMM, *Jiang* teaches computing a probability score for a plurality of predetermined acoustic states of an acoustic model. Further, *Jiang* teaches “[a]s the tree search engine traverses tree 77, it preferably computes a score, for each phoneme branch considered in tree 77, wherein the score represents the likelihood that the particular phoneme encoded by the codeword corresponds to the phoneme for the branch under consideration.” *Jiang*, 8:29-33. Thus, a score is computed for each node where the score indicates the likelihood that a phoneme for each node is the phoneme encoded by a codeword. *Jiang*, 8:16-42. In my opinion, *Jiang* thus teaches computing a score for each phoneme branch under consideration and since each phoneme branch is scored, a plurality of distances are calculated. Indeed, a POSITA would have understood that because the prefix tree has a plurality of nodes, multiple scores or distances are calculated.

150. *Jiang* teaches that the word search via the prefix tree begins, in the Fig. 3 example, with three possible phonemes, and the Fig. 3 prefix tree is “[f]or the sake of clarity, only a portion of a prefix tree.” *Jiang*, 8:5-15; *Ravishankar*, 33. In speech recognition, the most likely phonemes starting the speech utterance are calculated and other phonemes are pruned. *Jiang* discloses the Fig. 3 prefix tree begins with the phonemes represented by the letters AO, AE, and T. Because *Jiang* calculates at

least three scores for respective phonemes AO, AE, and T, *Jiang* teaches calculating distances (plural):





Jiang, Figs. 3-4, *see also Jiang*, Figs. 3, 5, 6. A POSITA would have understood that because *Jiang* teaches HMMs are used to determine the most likely phoneme from a feature vector (*Jiang*, 7:30-45), *Jiang* teaches the calculated score indicates the similarity (referred to in *Jiang* as the probability or likelihood) between a feature vector (the codeword) and the HMM states.

151. Specifically, as the user starts speaking, the feature vectors (or codewords) are scored based on the HMMs. *Jiang*, 7:38-41. *Jiang* explains that as the prefix tree is traversed: a score is assigned to each node connected to a phoneme branch then under consideration by the speech recognition system. The score is indicative of a likelihood that the particular phoneme being examined is the actual phoneme indicated by the codeword received from the feature extraction module 66. *Jiang*, 8:16-24. The tree search engine 74 computes the score for each phoneme branch. *Jiang*, 8:28-32. Because (1) the score is “indicative of a likelihood that the particular phoneme being examined is the actual phoneme indicated by the codeword received from the feature extraction module,” and (2) the tree search engine “determines a most likely phoneme represented by the code word received from feature extraction module,” the score is or includes a distance calculation.

152. *Jiang* teaches the tree search engine 74 is the component that traverses tree 77. *Jiang*, 8:28-32 (“As the tree search engine traverses tree 77, it preferably computes a score, for each phoneme branch considered in tree 77”). A POSITA would have understood that *Jiang*’s tree search engine 74 performs the claimed calculating distances and therefore teaches the claimed “calculating circuit.” As I explained above in Section IX.A.4.b.1, *Jiang* teaches the claimed “calculating circuit.” Because *Jiang* teaches tree search engine 74 determines a score indicative of the most likely phoneme and further teaches the most likely phoneme is

determined by computing a probability score indicating the similarity of a feature vector to an HMM state, a POSTIA would have understood *Jiang* teaches or renders obvious Claim 1(c).

153. *Jiang*'s threshold teachings further teach calculating more than one distance. For example, *Jiang* teaches

FIG. 6 illustrates tree 102 (shown in FIG. 5) with a number of the silence branches (and the branches connected thereto) pruned from the tree. After considering each of the silence branches, representing the context dependent silence phones at the input side of tree 102, **a score is assigned to each node** connected to those silence branches. Then, in one preferred embodiment, **each of the scores for each node are compared to the maximum score assigned to any node on that level of the tree. A predetermined threshold level can be set, or an adaptive threshold level can be implemented**, for the comparison. **If the score for the node being compared is less than the maximum score by the threshold level, all subsequent branches connected to that node are pruned from the tree**, thereby drastically reducing the search space for any given search. FIG. 6 illustrates that the score assigned to the node for silence branch 108 is sufficiently low that the remainder of the branches of the tree have been pruned from the tree.

Jiang, 10:65–11:15. A POSITA would have recognized *Jiang*'s threshold teachings expressly state “scores” (plural) are assigned and compared to a threshold value. *Id.* Because more than one score is compared to the threshold value, a POSITA would

have understood that *Jiang* teaches multiple distances (i.e., probabilities or likelihoods) are calculated.

154. As will be explained in greater detail with respect to Claim 1(d), *Jiang* goes on to teach that a prefix tree is traversed to identify spoken words, where the possible words are constrained by lexicon 70, and where the tree comprises a plurality of phoneme branches corresponding to the constrained set of words from the lexicon. *Jiang*, 7:49-54, 8:5-13, Fig. 3. As the tree is traversed, a score is determined for each node and associated phoneme branch to determine the likelihood the phoneme being examined is the phoneme indicated by the codeword. *Jiang*, 8:16-32. As discussed above, this is similar to *Ravishankar*'s teachings. *Ravishankar*, 33. The likelihood of the phoneme is computed based on the phonetic speech unit models (HMMs) discussed in *Jiang*, 7:30-46. Accordingly, a POSITA would have understood that as the speech is input, sampled, and the feature vectors extracted (or encoded), *Jiang*'s tree search engine traverses the prefix tree, scoring the codeword based on the HMMs, in a known manner.

c) *Baumgartner* Teaches a “calculating circuit”

155. In my opinion, *Baumgartner* teaches the claimed “calculating circuit.” As is explained below in detail, *Baumgartner* teaches a Speech Label Accelerator (SLA) (i.e., the claimed “second processor” in Claim 1(e)). See Section IX.A.6. Specifically, *Baumgartner* teaches

The SLA operates on this sequence of observation vectors, one vector at a time. The SLA performs the work of labeling block 130. The labeling block 130 performs a computation that is discussed below and determines probabilities 135 for a number of phones. Phones are small, indivisible acoustic elements of speech.

Baumgartner, [0038]. Thus, *Baumgartner* teaches determining a probability for a particular phone, where the probability is determined by the SLA.

156. A POSITA would have understood that probability—such as that discussed by *Baumgartner*—indicates how good a match the observed data from the feature vector is to a particular HMM state thereby showing how closely the HMM state matches the feature vector (i.e., the claimed “distance[]”). For example, a POSITA would have understood that it was well-known in the art that such probability indicates how good the feature vector matches a particular HMM state. *See, e.g., Ravishankar*, 21 (“In the case of HMM-based systems, the [acoustic probability computation] refers to the computation of the probability of a given HMM state emitting the observed speech at a given time.”); *Rabiner*, 82 (“in which the unknown test pattern is compared with each (sound) class reference pattern and a measure of similarity (distance) between the test pattern and each reference pattern is computed.”); *Nadas*, 1:44-49 (“[i]n associating prototype vectors with feature vectors, the feature vector for each time interval is typically compared to each

prototype vector. Based on a predefined closeness measure, the distance between the feature vector and each prototype vector is determined and the closest prototype vector is selected.”). *Baumgartner* further teaches: “the heart of the SLA is a **circuit** that permits the fast approximate computation of a large collection of values $P=\{P(v|l)\}$ for a fixed observation vector v .” *Baumgartner*, [0043].

157. In my opinion, a POSITA would have understood that *Baumgartner’s* SLA teaches the claimed “calculating circuit,” because *Baumgartner* expressly teaches “the heart of the SLA is a circuit” that performs the functionality described above related to probabilities, likelihoods, and similarities. A POSITA would have understood that *Baumgartner’s* “observation vectors” are the feature vectors commonly described in the art. As further explained below, *Baumgartner’s* observation vectors are feature vectors because they are created by converting input speech signal into a vector by mel-frequency cepstral estimation. *Baumgartner*, [0037]-[0038]; *See* Section IX.A.6.b.

158. In my opinion, *Baumgartner* expressly teaches the use of acoustic models. For example, *Baumgartner* teaches an “acoustic model is a set of d-dimensional probability density functions or pdfs.” *Baumgartner*, [0040]. *Baumgartner* goes on to teach that the probability density functions (which *Baumgartner* cites as “pdf[s]”) are “typically implemented as a mixture of d-dimensional Gaussians, given by a P function described in *Baumgartner’s* Equation

(1). *Baumgartner*, [[0040]-[0041]. *Baumgartner* further explains that “such [P] functions are distributed among the leaves of a family of decision trees” and that such P functions are “sometimes also called a leaf.” *Baumgartner*, [0042]. Specifically, *Baumgartner* teaches

The computation that the SLA speeds up (or in fact approximates, by a procedure detailed below) is the following. **Given a single observation vector, v , associated with a single frame of speech, the top-ranking K leaves $l_1, l_2, \dots, l_K$ in the set M are to be determined, as ordered by the likelihoods $\{P(v|l)\}_{l=1}^L$.** Although in principle this involves computing the value of all L leaves present in the acoustic model M , in fact a hierarchical scheme, which is described below in Section 5.3, makes it possible to compute just a subset of leaf values. In any event, the heart of the SLA is a circuit that permits the fast approximate computation of a large collection of values $P=\{P(v|l)\}$ for a fixed observation vector v .

Baumgartner, [0043]. A POSITA would have understood these disclosures to mean *Baumgartner* teaches that each observation vector is associated with a probability or likelihood. Accordingly, in my opinion, *Baumgartner* teaches the SLA determines a probability for a particular phone. *Baumgartner*, [0037]-[0043].

159. *Baumgartner* also teaches its SLA determines probabilities for a number of phones. For example, *Baumgartner* teaches

The SLA operates on this sequence of observation vectors, one vector at a time. The SLA performs the work of labeling block 130. The labeling block 130 performs a computation that is discussed below and determines probabilities 135 for a number of phones. Phones are small, indivisible acoustic elements of speech. Trellis match 140 then performs a search and selects a best word or phrase 145. Now that an overall system has been described, SLA 130 will be described in more detail.

Baumgartner, [0038]. A POSITA would have understood that when *Baumgartner* teaches its SLA “performs the work of labeling block 130” and labeling block 130 “determines probabilities 135 for a number of phones,” *Baumgartner* teaches its SLA indeed determines probabilities that a given input speech vector corresponds to a given state (or phone). *Baumgartner*, [0037]-[0038]. Such probability was well-known in the art to determine the “distance” from a feature vector to a given HMM state, as I explained above. *See* Section IX.A.4.b (and additional Sections cited therein). A POSITA would have understood that the probability (or distance) computation determines how closely an HMM state matches an input feature vector.

160. In my opinion, *Baumgartner*’s SLA is or otherwise comprises a circuit that performs the computation of values (probabilities) for each observation vector (feature vector) and therefore teaches the claimed “calculating circuit.” *Baumgartner* expressly teaches “[i]n any event, the heart of the SLA is a circuit that permits the fast approximate computation of a large collection of values $P=\{P(v|l)\}$ for a fixed

observation vector v .” *Baumgartner*, [0043]. A POSITA would have indeed understood *Baumgartner*’s SLA as comprising a circuit. *See Microsoft Computer Dictionary*, 5-8,; *Dictionary of Electrical and Computer Engineering*, 6-7; *see also* Section IX.A.1 (explaining a POSITA would have understood a processor comprises a circuit). *Baumgartner* further teaches an SLA comprises adder circuitry, which adds “atoms” (akin to vector quantization) to determine approximate values for “kernels” (akin to Gaussians or probability density functions) and “allows fast and efficient computation of kernels” that are later “used to determine a word or phrase for a segment of speech.” *Baumgartner*, [0008], [0062]–[0080], [0095]. A POSITA would have therefore understood that *Baumgartner*’s SLA is a calculating circuit for the reasons explained above and as I explain in greater detail in Section IX.A.6.

d) Motivation to Combine *Jiang* and *Baumgartner*’s Teachings

161. *See* Section IX.A.6.f, which provides a detailed motivation to modify *Jiang* to include *Baumgartner*’s SLA (the claimed “second processor”) including circuitry for performing the functionality of Claim 1(c)—namely, performing the distance or likelihood calculations.

5. *Claim 1(d): “a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words;”*

a) “a lexical tree”

162. Regarding the claimed “lexical tree,” *Jiang* teaches a prefix tree. *Jiang* explains that its prefix tree is created by organizing HMMs from trained data, by specifically teaching:

One or more HMMs are created by training module 65 **for each phoneme** of each word in a desired vocabulary represented by the training data. **The HMMs thus represent model output distributions associated with the phonemes** and the vocabulary. **The prefix tree is then formed**, based upon the desired vocabulary, such that **phonemes are structured to provide a leaf** which is associated with each word in the desired vocabulary.

Jiang, 11:25-47, 7:30–8:51. A POSITA would have understood that HMMs represent phones or groups of phones, and that it was very well-known to organize the HMMs into a lexical tree structure. *Ravishankar*, 28 (explaining how “[m]ost systems use hidden Markov models (HMMs) to represent the basic units of speech. ... almost all systems use HMMs for modelling triphones[,]” and further explaining HMM modelling), 33 (“Organizing the HMMs to be searched as a phonetic tree instead of the flat structure of independent linear HMM sequences for each word is

probably the most often cited improvement in search techniques in use currently.

This structure is referred to as *tree-structured lexicon* or *lexical tree*.”).

163. *Jiang*’s teachings of prefix trees are consistent with lexical tree searches as well known in the art. For example, *Ravishankar* explains

Organizing the HMMs to be searched as a phonetic tree instead of the flat structure of independent linear HMM sequences for each word **is probably the most often cited improvement in search techniques in use currently**. This structure is referred to as tree-structured lexicon or lexical tree.

Ravishankar, 33. A POSITA would have understood *Jiang*’s prefix tree teachings are consistent with the lexical tree teachings in the art, such as *Ravishankar*. For example, *Jiang* teaches “[t]he words are preferably presented to tree search engine 74 in the form of a prefix tree which can be traversed from a root to a leaf (or to an internal word node) to arrive at the word most likely indicative of the utterance of the user.” *Jiang*, 7:67–8:4. *Jiang* also teaches “[e]ach leaf on the tree (which represents a word) is connected to a silence branch, such as branches 94, 96, and 98, each of which are associated with a silence phone. By employing prefix tree 88, the present system provides a silence bracketed lexicon wherein every word in the lexicon is bracketed by silence phone.” *Jiang*, 9:37-42. Accordingly, *Jiang* teaches

leaves are words in the prefix tree, leaves are comprised of collections of phonemes, and each phoneme is associated with a phoneme branch.

164. A POSITA would have accordingly understood the connection between *Jiang's* prefix tree and acoustic model (or HMMs). For example, in addition to the disclosures cited in the preceding paragraph, *Jiang* teaches

In such a prefix tree, each subword unit (such as a phoneme) is typically represented by a branch which is associated with a particular phonetic model (such as an HMM). The phoneme branches are connected, at nodes, to subsequent phoneme branches. All words in the lexicon which share the same first phoneme share the same first branch. All words which have the same first and second phonemes share the same first and second branches. By contrast, words which have a common first phoneme, but which have different second phonemes, share the same first branch in the prefix tree but have second branches which diverge at the first node in the prefix tree, and so on. The tree structure continues in such a fashion such that all words likely to be encountered by the system are represented by the end nodes of the tree (i.e., the leaves on the tree).

Jiang, 2:53-67. In other words, *Jiang* teaches acoustic models, such as HMMs, are associated with branches in *Jiang's* prefix tree. Accordingly, a POSITA would have understood *Jiang's* prefix tree utilizes a plurality of HMMs and that each node and associated branch of the prefix tree is associated with one or more HMMs.

165. Therefore, a POSITA would have understood *Jiang*'s prefix tree is a "lexical tree." Specifically, a POSITA would have recognized *Jiang* teaches "[a] prefix tree represents the lexicon of the speech recognition system as a tree structure wherein all of the words likely to be encountered by the system are represented in the tree structure." *Jiang*, 2:47-52. A POSITA would have understood that because the words of lexicon 70 are presented to tree search engine 74 in the form of a prefix tree, *Jiang* teaches a "lexical tree" in the conventional sense. See, e.g., *Ravishankar*, 33-34. A POSITA would have understood *Jiang*'s prefix tree is simply a particular form of a lexical tree. *Ravishankar*, 33-34. It was well-known for lexical trees to comprise nodes and branches. *Ravishankar*, 33-34. Accordingly, in my opinion, *Jiang* teaches or at least renders obvious the claimed "lexical tree."

b) "the lexical tree comprising a model of words"

166. Regarding the claimed "lexical tree comprising a model of words," *Jiang* teaches the prefix tree comprises all words likely to be encountered by the speech recognition system. *Jiang*, 2:49-52 ("A prefix tree represents the lexicon of the speech recognition system as a tree structure wherein all of the words likely to be encountered by the system are represented in the tree structure."), 7:65-8:4 ("lexicon 70 contains information which is representative of all of the words in the vocabulary of speech recognition system 60. The words are preferably presented to tree search engine 74 in the form of a prefix tree which can be traversed from a root

to a leaf (or to an internal word node) to arrive at the word most likely indicative of the utterance of the user.”), 7:48-54. I note that the ’277 Patent describes a “model of words” may comprise words. ’277 Patent, 37:56-59. *Jiang*’s prefix tree comprises a lexicon 70 which is all words in the system’s vocabulary. Therefore, in my opinion, *Jiang* teaches or at least renders obvious the claimed “lexical tree comprising a model of words.”

c) “a search stage for using said calculated distances to identify words within a lexical tree”

167. Regarding the claimed “search stage for using said calculated distances to identify words within a lexical tree,” *Jiang* teaches search engine 74 includes a search stage of traversing a prefix tree to identify words within the prefix tree. *Jiang*, 8:16-43. A POSITA would have understood that *Jiang* uses the calculated distances (i.e., the first score indicative of the most likely phoneme for a particular node and associated phoneme branch) to identify words. *Jiang*, 6:67–7:4, 7:48-54 (“the information received by tree search engine 74 based on its accessing of the phonetic speech unit models in memory 72 is used in searching lexicon 70 to determine a word which most likely represents the codewords received by feature extraction module 66”), 7:65-67 (“The lexicon “contains information which is representative of all of the words in the vocabulary of speech recognition system 60.”), 8:16-43. I note this is similar to how *Ravishankar* describes “[o]rganizing the HMMs to be searched as a phonetic tree,” as was well-known to a POSITA. *Ravishankar*, 33.

168. *Jiang* teaches its system obtains information from the “phonetic speech unit models” (for example, the HMMs discussed in *Jiang* and well known in the art) and this information includes likelihood scores. *Jiang*, 7:30-47, 8:16-42. A POSITA would have understood *Jiang* to mean that *Jiang* uses HMMs to calculate scores indicative of the likelihood (similarity) that a phoneme being examined in *Jiang*’s prefix tree is the actual phoneme indicated by the codeword. *Jiang*, 7:30-47, 8:16-42. Because the determination of the codeword utilizes the HMM states of *Jiang*, the “information” referenced in *Jiang* is received by the tree search engine based on accessing the phonetic speech unit model and is “used in searching lexicon 70 to determine a word” representative of the utterance. *Jiang*, 7:34-51. Accordingly, a POSITA would have understood such “information” includes at least the likelihood score using HMMs. *Jiang*, 7:30-47, 8:16-42.

169. *Jiang*’s search stage can further be illustrated by *Jiang*’s exemplary traversal of the word “ORANGE.” *Jiang*, 8:25-43.

For example, if the word ORANGE were input into system 60 by the user, feature extraction module 66 would likely divide the word ORANGE into codewords indicative of the phonemes represented as follows: AO, R, IX, N, JH. As the tree search engine traverses tree 77, it preferably computes a score, for each phoneme branch considered in tree 77, wherein the score represents the likelihood that the particular phoneme encoded by the codeword corresponds

to the phoneme for the branch under consideration. Thus, tree search engine 74 computes a score for node 82 which indicates that the first codeword under consideration is highly likely to be represented by the AO phoneme corresponding to the branch under consideration. Tree search engine 74 also preferably computes a score for each of the other nodes 84 and 86 in tree 77, wherein the score is indicative of the likelihood that the codeword being analyzed is represented by the phonemes AE, and T. Under ideal circumstances, the score assigned to nodes 84 and 86 is lower than the score assigned to node 82.

Jiang, 8:25-43. A POSITA would have understood *Jiang*'s identification of the word ORANGE (i.e., phonemes AO-R-IX-N-JH) uses one or more distances (i.e., the score for node 82 indicating the phoneme corresponding to the codeword is highly likely to be AO) where the scores are determined sequentially. After the score for the phoneme AO is determined to be a likely match (low distance value equating to a high probability or likelihood match), *Jiang*'s prefix tree uses the AO-identified phoneme to perform the identification of the word ORANGE. *Jiang*, 8:5-28.

170. As I explained above, there are a finite number of phonemes in the English language. See Section V.A (citing *Schmandt*, 23-24 (“the set of [phonemes] defines all the sounds from which words can be constructed in a particular language”)). A POSITA would therefore have understood that there are a relatively “small number of initial branches of the tree,” as expressly taught by *Jiang*. *Jiang*,

3:1-21. The scores for the initial branches are, or are based on, the scores used to determine the initial phoneme of the codeword (i.e., the feature vector). *Jiang*, 8:5-43 (discussing that the initial nodes include nodes for the phonemes AO, AE, and T). Each of these nodes would have an assigned score representing the distance calculation. Accordingly, the initial branches of the prefix tree used the calculated distances which indicate the similarity between phonemes and the feature vector (for example the AO, AE, and T shown in *Jiang*'s Figure 3). A POSITA would have understood this prefix tree traversal as *Jiang*'s tree search engine searching the tree to identify words within the tree, thus teaching the claimed "search stage." Therefore, in my opinion, *Jiang* teaches, or at least renders obvious the claimed "search stage for using said calculated distances to identify words within a lexical tree."

171. A POSITA would have understood that *Jiang*'s tree search engine is *searching Jiang's* prefix tree in order to identify words within the tree. A POSTIA would have understood that based on the disclosures cited and explained above, the traversal of *Jiang*'s prefix tree by *Jiang*'s tree search engine teaches or renders obvious the claimed "search stage," as claimed. *Jiang*, 7:67-8:4.

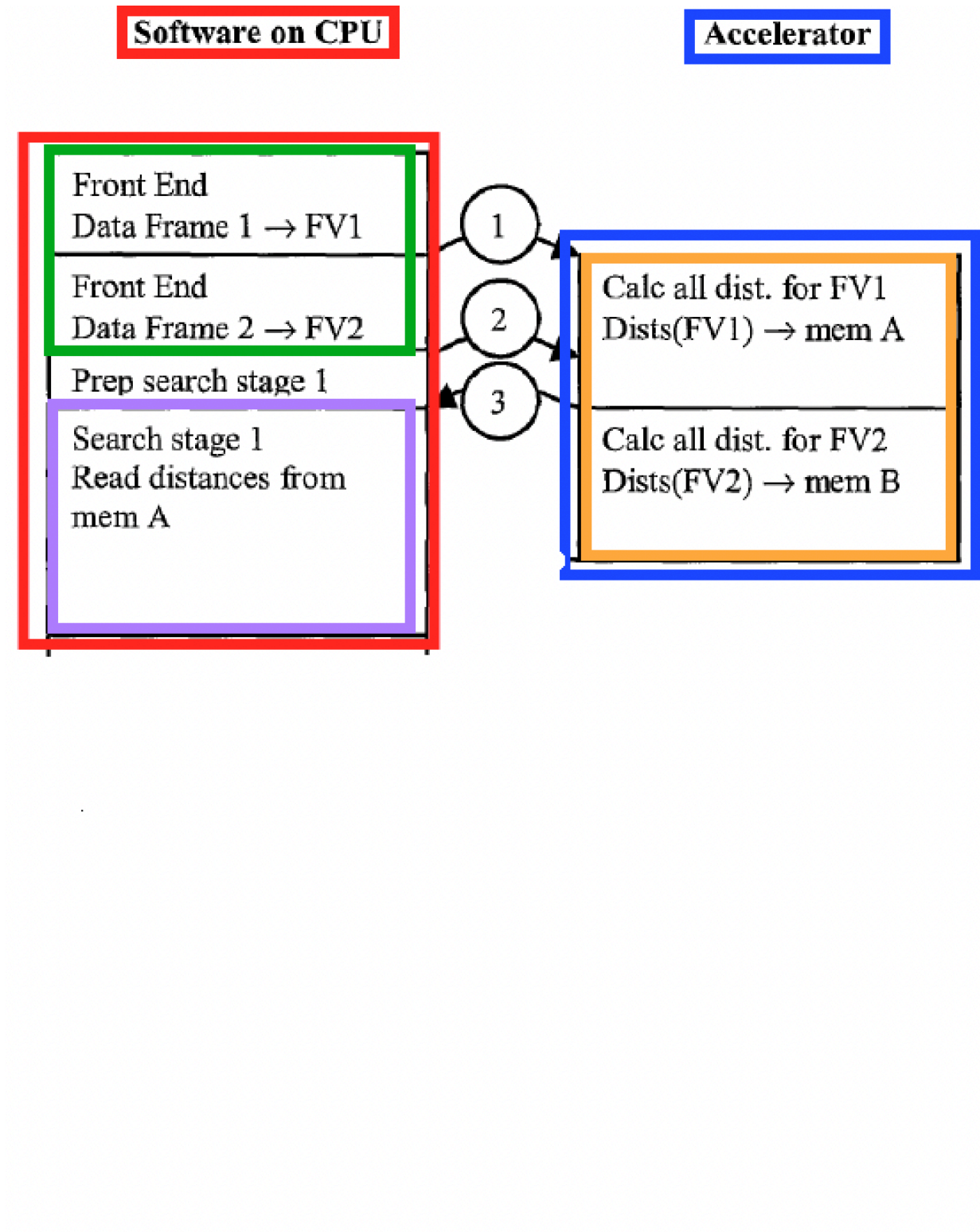
6. Claim 1(e): “wherein said audio front end and said search stage are implemented using a first processor, and said calculating circuit is implemented using a second processor, and”

172. In my opinion, the teachings of *Jiang* modified to include the hardware architecture taught by *Baumgartner* render obvious Claim 1(e). As I explained above, *Jiang*, either alone or in combination with *Baumgartner*, teaches or renders obvious the functionalities of the claimed “audio front end,” “calculating circuit,” and “search stage.” See Sections IX.A.1-IX.A.5. As explain in detail below, *Baumgartner* teaches a similar front end and search stage “implemented using a first processor,” as claimed and a similar calculating circuit “implemented using a second processor,” as claimed. As I also explain below, a POSITA would have been motivated to modify *Jiang* to include the hardware architecture taught by *Baumgartner*.

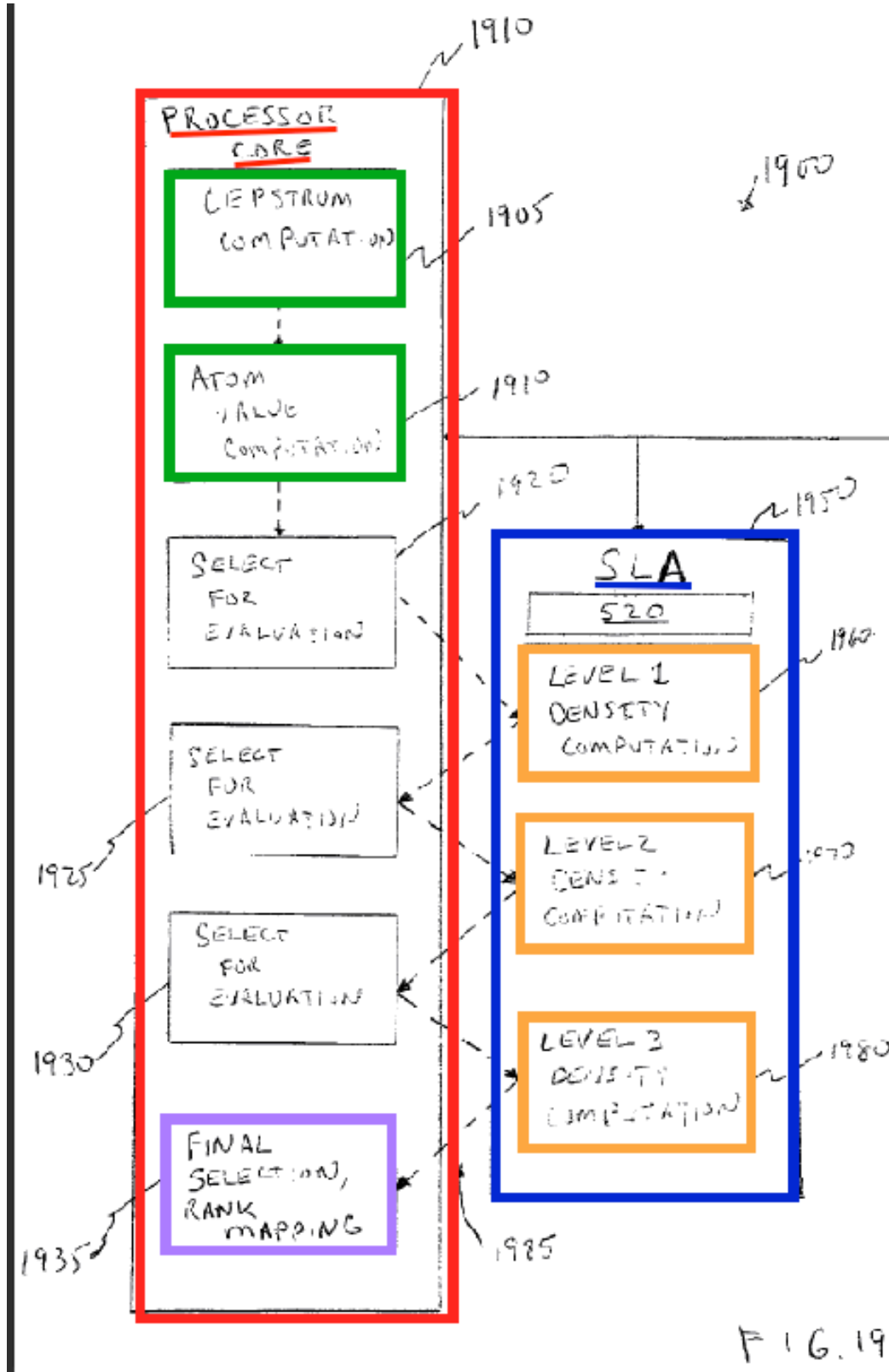
a) Overview of Mapping

173. I note that the '277 Patent's Figure 21 illustrates “a speech recognition circuit having a shared processor for the front end and search stages.” '277 Patent, 8:54-57, Fig. 21.

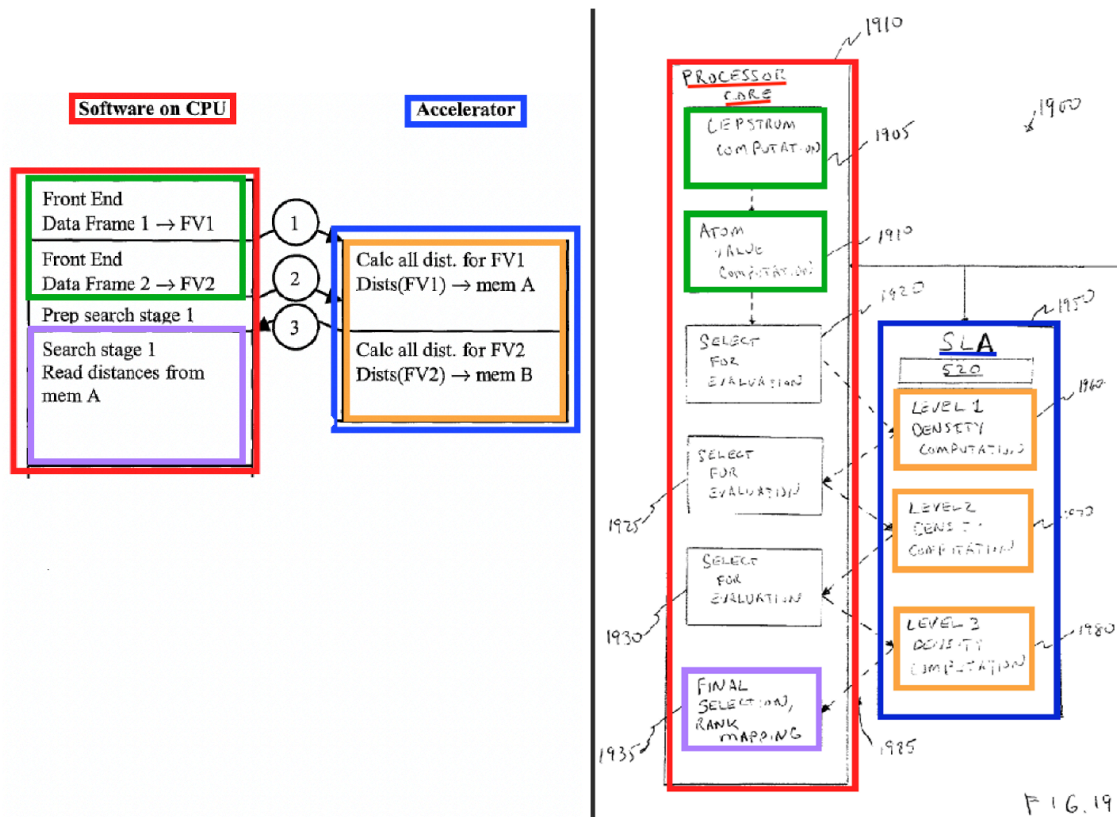
174. As I explain in detail below, *Baumgartner*'s Figure 19 teaches the hardware architecture of Claim 1(e) and Figure 21 of the '277 Patent. Comparing the '277 Patent's Figure 21 with *Baumgartner*'s Figure 19 evidences how *Baumgartner* teaches the claimed hardware architecture:



'277 Patent, Fig. 21 (annotated, excerpted).



Baumgartner, Fig. 19 (annotated, excerpted).



'277 Patent, Fig. 21 (annotated, excerpted, left); Baumgartner, Fig. 19 (annotated, excerpted, right).

175. In the annotated figures above the claimed “audio front end” is annotated green, the claimed “search stage” is annotated purple, and the claimed “first processor” is annotated red. Additionally, the claimed “calculating circuit” is annotated orange, and the claimed “second processor” is annotated blue.

176. Baumgartner’s Figure 19 teaches a hardware architecture of a speech recognition system. Baumgartner, [0031]-[0033], [0156]-[0158], Fig. 19. Baumgartner’s cepstrum computation 1905 and atom value computation 1910 (annotated green) serve as an audio front end. Baumgartner’s final selection rank

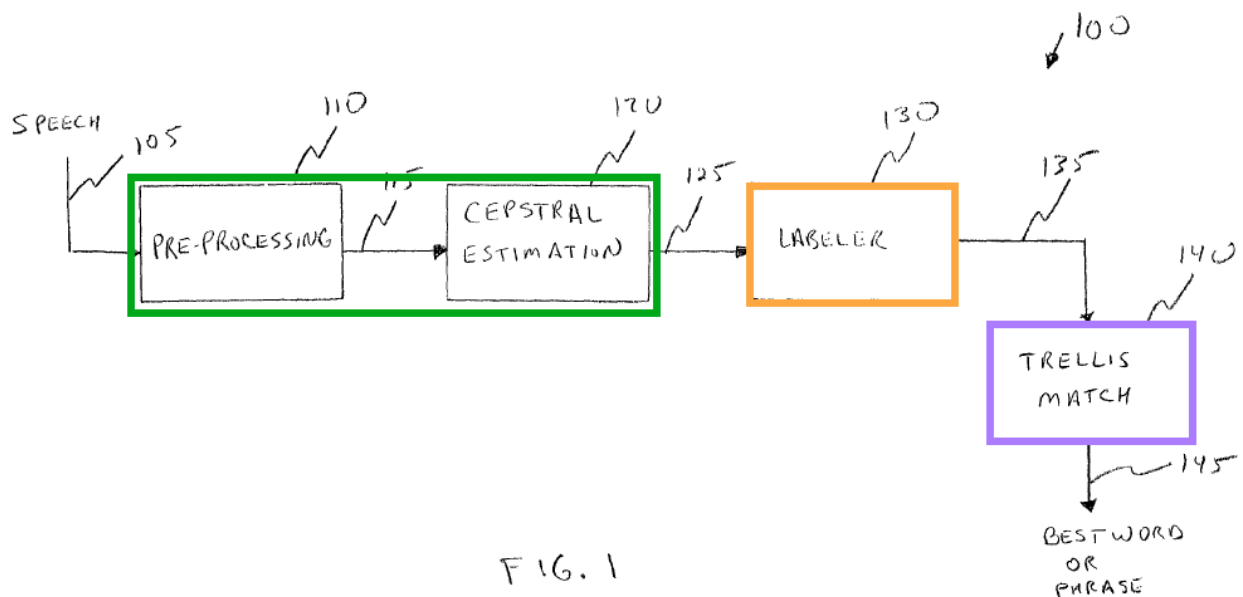
mapping 1935 (annotated purple) at least serves as a search stage. As seen in the annotated Figure 19, the cepstrum computation 1905, atom value computation 1910, and rank mapping 1935 are implemented on processor core 1910 (annotated red) which serves as or comprises a first processor. *Baumgartner's* density computations 1960-1980 (annotated orange) serve as distance calculations. As also seen in *Baumgartner's* annotated Figure 19, the density computations are implemented on Speech Label Accelerator (SLA) 150 (annotated blue) which serves as or comprises a second processor. I note that *Baumgartner's* Figure 19 also includes intervening steps 1920, 1925, and 1930 and a zig-zag line 1985 that are not annotated as they are not included in the mapping for Claim 1.

b) *Baumgartner* Teaches an “audio front end,” a “calculating circuit,” and a “search stage”

177. In my opinion, *Baumgartner* the well-known speech recognition stages, including a front end, a calculating circuit, and a search stage. Specifically, *Baumgartner* teaches a Speech Label Accelerator (SLA) for speech recognition and techniques for using such SLA. *Baumgartner*, [0002] (“The present invention relates to speech recognition and, more particularly, to Speech Label Accelerators (SLAs) and techniques for using such SLAs.”).

178. As I explained above it was well known in speech recognition systems to utilize signal processing, feature vectors, acoustic states, acoustic models, similarity indications between feature vectors and acoustic states of acoustic models,

and searches to identify words. *See* Sections V.A-V.E. In my opinion, *Baumgartner* teaches these well-known speech recognition aspects. For example, *Baumgartner*'s Figure 1 depicts the well-known speech recognition aspects. *Baumgartner*, [0011] ("FIG. 1 shows an exemplary system for converting speech to text"), Fig. 1. *Baumgartner*'s Figure 1 provides:



Baumgartner, Fig. 1 (annotated), [0033]-[0043] (with pinpoint, [0037]-[0038]). *Baumgartner*'s annotated Figure 1 above shows the well-known aspects of speech recognition. *Baumgartner*, [0033]. For example, *Baumgartner* teaches

Turning now to FIG. 1, an exemplary speech recognition system 100 is shown. Speech recognition system 100 operates on a speech signal 105 to produce a best word or phrase 145. System 100 comprises a pre-processing block

110, a cepstral estimation block 120, a labeler 130, and a trellis match block 140.

Let $s(t)$ represent a continuous-time, continuous-amplitude audio signal 105, which is assumed to contain some speech that is to be decoded into text. Let this signal be quantized in time and in amplitude by pre-processing block 110, yielding a sequence of samples denoted $s_0, s_1, \dots, s_i$. In the first stage of the speech recognition computation, this stream of samples is divided into equal-sized, overlapping blocks of contiguous samples. Each block is called a frame. A typical frame contains 256 samples. Thus, signal 115 is a series of overlapping frames. Those skilled in the art will realize that, to convert the speech signal 105 into frames 115, the pre-processing block 110 can contain such devices as an analog-to-digital converter, a window/filter, and a Fast Fourier Transform.

By applying selected signal-processing operations, for instance, mel-frequency cepstral estimation 120, each frame from signal 115 is converted into an observation vector. Signal 125 comprises a series of observation vectors. The relation between frames and observation vectors is one-to-one. The sequence of frames 115 is written as $x_0, x_1, \dots, x_j$, and the associated sequence of observation vectors 125 is written as $v_0, v_1, \dots, v_j$. The SLA operates on this sequence of observation vectors, one vector at a time. The SLA performs the work of labeling

block 130. The labeling block 130 performs a computation that is discussed below and determines probabilities 135 for a number of phones. Phones are small, indivisible acoustic elements of speech. Trellis match 140 then performs a search and selects a best word or phrase 145. Now that an overall system has been described, SLA 130 will be described in more detail.

Baumgartner, [0036]-[0038].

179. Regarding *Baumgartner*'s similar "audio front end," in my opinion, *Baumgartner*'s Figure 1 shows pre-processing 100 and cepstral estimation 120 (annotated green) and teaches an "audio front end," as claimed. *Baumgartner*'s pre-processing 110 and cepstral estimation 120 convert audio signal 105 into frames and apply well-known signal processing operations (for example, cepstral estimation) to obtain an observation vector. *Baumgartner*, [0037]-[0038] ("Those skilled in the art will realize that, to convert the speech signal 105 into frames 115, the pre-processing block 110 can contain such devices as an analog-to-digital converter, a window/filter, and a Fast Fourier Transform. By applying selected signal-processing operations, for instance, mel-frequency cepstral estimation 120, each frame from signal 115 is converted into an observation vector."); *Jiang*, 6:46-7:1; *Ravishankar*, 27-28 (describing how sampled speech is preprocessed into a cepstrum), 40-41; *Dictionary of Electrical and Computer Engineering*, 5 ("Cepstrum vocoder" is "A digital device for reproducing speech in which samples of the cepstrum of speech,

together with pitch information, are transmitted to the receiver, and are then converted into an impulse response that is convolved with an impulse train generated from the pitch information.”). I note the ’277 Patent explains the same well-known concepts. ’277 *Patent*, 11:65–12:10 (describing how digital signals are converted into feature vectors). I further note the similarity to *Jiang*. *Jiang*, 6:46–7:1 (also describing how digital signals are converted into feature vectors). *Baumgartner*’s teachings are similar to *Jiang*’s teachings mapped above as the claimed “audio front end” and consistent with the well-known signal processing and feature extraction techniques used in speech recognition.

180. Regarding *Baumgartner*’s similar “calculating circuit,” in my opinion *Baumgartner*’s Figure 1 shows labeler 130 (annotated yellow) and teaches a “calculating circuit,” as claimed. *Baumgartner* teaches “[t]he SLA operates on this sequence of observation vectors, one at a time. The SLA performs the work of labeling block 130. The labeling block 130 performs a computation that is discussed below and determines probabilities 135 for a number of phones. Phones are small, indivisible acoustic elements of speech.” *Baumgartner*, [0038]. *Baumgartner* also teaches the SLA determines “likelihoods” of a “[g]iven observation vector, v , associated with a single frame of speech.” *Baumgartner*, [0043]. *Baumgartner* further teaches “the heart of the SLA is a circuit that permits the fast approximate computation of a large collection of values $[\]$ for a fixed observation vector v .”

Baumgartner, [0043]. *Baumgartner*'s description of labeler 130 performed by the SLA is similar to the probability computation well-known in the art. See Section V.E. For example, *Ravishankar* describes a similar probability computation "of observing a given speech input x in HMM state s ." *Ravishankar*, 29. *Jiang* also describes a similar process as I explained above. See Section IX.A.4.b (citing *Jiang*, 7:30-8:51, describing the functionality of *Jiang*'s tree search engine 74, and noting *Jiang* teaches this is done in a well-known manner). Accordingly, a POSITA would have recognized that *Baumgartner*, *Ravishankar*, and *Jiang* teach similar probability computations to indicate the similarity between an input feature vector and a given state in an acoustic model, as was well-known in the art. I note the same concept is described in the '277 Patent. '277 Patent, 3:51-56, Fig. 16.

181. Regarding *Baumgartner*'s similar "search stage," in my opinion *Baumgartner*'s Figure 1 shows trellis match 140 (annotated purple) and teaches a "search stage," as claimed. *Baumgartner* teaches that after the SLA (labeling block 130) determines probabilities, "[t]rellis match 140 then performs a search and selects a best word or phrase 145." *Baumgartner*, [0038]. This is similar to the well-known manner referenced in *Jiang*. *Jiang*, 1:33-40 ("A sequence of state transitions can be represented, in a known manner, as a path through a trellis diagram that represents all of the states of the HMM over a sequence of observation times. Therefore, given an observation sequence, a most likely path through the trellis diagram (i.e., the most

likely sequence of states represented by an HMM) can be determined using a Viterbi algorithm.”). This is also similar to the ’277 Patent. ’277 Patent, 13:33–15:57. Accordingly, in my opinion, *Baumgartner*’s trellis search is similar to *Jiang*’s tree search engine which identifies a word and is mapped as the “search stage,” as claimed, and is consistent with the well-known search techniques used in speech recognition.

182. *Baumgartner*’s Figure 1 illustrates and teaches the well-known speech recognition functionality of the three speech recognition stages—feature extraction, distance or probability calculation, and word search. *Baumgartner*, [0033], Fig. 1. *Baumgartner*’s Figure 1 does not expressly disclose hardware but provides the functionality of these three well-known speech recognition stages. *Baumgartner*, [0033], Fig. 1. *Baumgartner* teaches that the well-known functionality of the three speech recognition stages described for Figure 1 “may be designed as an apparatus” including a “wide variety of hardware designs that are possible for the same basic idea.” *Baumgartner*, [0033]. *Baumgartner* further describes “various extensions and refinements” of such hardware designs are possible and described in *Baumgartner*’s specification, including *Baumgartner*’s Figure 19. *Baumgartner*, [0033]. In my opinion, a POSITA would have understood *Baumgartner*’s Figure 19 is one such hardware design of the well-known speech recognition system described for *Baumgartner*’s Figure 1.

**c) *Baumgartner* Teaches an Audio Front End
“implemented using a first processor”**

183. In my opinion, *Baumgartner* teaches an audio front end “implemented using a first processor,” as claimed. In *Baumgartner*’s Figure 19—which as explained above is a hardware architecture of a well-known speech recognition system—*Baumgartner* teaches cepstrum computation 1905 and atom value computation 1910 (annotated green) on processor core 1910 (annotated red). *Baumgartner*, [0037]-[0038], [0156]-[0157], Fig. 19. A POSITA would have understood that *Baumgartner*’s reference to “cepstrum” in Figure 19 is akin to *Baumgartner*’s “cepstral” reference in Figure 1. For example, a POSITA would have understood that “cepstral” frames undergo “cepstral” normalization resulting in “cepstrum” vectors and that such occurs in signal processing front ends in speech recognition systems. See *Ravishankar*, 40-41, Fig. 3.1. Ravishankar’s Figure 3.1 provides:

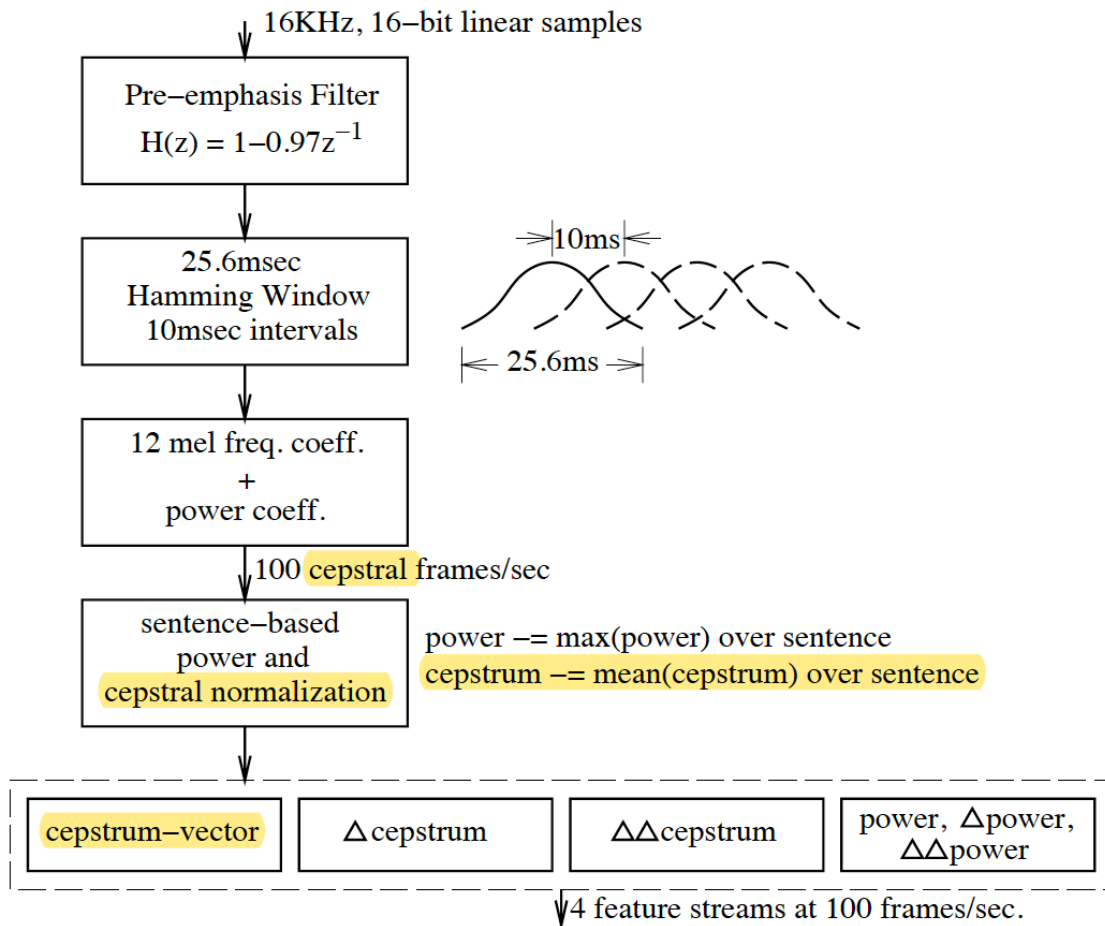


Figure 3.1: Sphinx-II Signal Processing Front End.

Ravishankar, Fig. 3.1 (annotated), 40-41.

184. I note that *Baumgartner*'s Figure 19 shows an "atom value computation 1910" occurring on *Baumgartner*'s processor core. *Baumgartner*, Fig. 19. A POSITA would have understood *Baumgartner*'s atom computation is an indexing function, which is akin to other techniques known in the art such as clustering or vector quantization, which is separate from the claimed distance calculation. *Baumgartner*, [0062]–[0079]. For example, *Ravishankar* teaches a similar vector

quantization technique. *Ravishankar*, 48-50. The '277 Patent also describes a decompression method and clustering. '277 Patent, 4:45-50. A POSITA would have understood that *Baumgartner*'s "atom value computation" (i.e., a vector quantization method), is a front-end process. As I explained above, *Jiang* teaches an optional codeword functionality where the codewords undergo a similar vector quantization process during front-end processing and feature vector generation. See Sections IX.A.2, IX.A.4.b.3 (together describing *Jiang*'s feature vector codewords as a vector quantization process in *Jiang*'s front-end).

185. A POSITA would have understood *Baumgartner*'s processor core 1910 comprises a processor, as recited in Claim 1(e), that performs the functionalities related to signal processing and vector extraction, as recited in Claims 1(a)-1(b). For example, the '277 Patent describes the claimed "first processor" as an exemplary CPU performing the audio front end functions. '277 Patent, 30:40–33:67, Fig. 21. In the '277 Patent's discussion of Figure 21, the '277 explains "[a]t the front-end processing for frame 1, the CPU passes the feature vectors for frame 1 to the accelerator, allowing distance calculations to start." '277 Patent, 30:50-53. The '277 Patent next explains that after the accelerator completes distance calculations, "[t]he CPU is now ready to start the search stage for frame 1." '277 Patent, 30:57-61. The '277 Patent goes on to explain how this process occurs for additional frames of speech data in an exemplary pipelined system. '277 Patent, 30:53–31:45. The '277

Patent describes the process by referring the “CPU task schedule.” ’277 Patent, 31:47. A POSITA would have understood these disclosures related to the ’277 Patent’s “CPU” performing various functionality and/or tasks and understood that such CPU comprised a processor. Specifically, a POSITA would have recognized such CPU’s functions or tasks related to front-end processing and feature vector extraction. I also note that *Baumgartner* describes processor core 1910 as executing a series of steps. *Baumgartner*, [0156].

**d) *Baumgartner* Teaches a Calculating Circuit
“implemented using a second processor”**

186. I note that the ’277 Patent describes an exemplary accelerator may be used as the claimed “second processor.” ’277 Patent, 30:40–33:67, Fig. 21.

187. As I explained above in Section IX.A.4.c, *Baumgartner* teaches a “calculating circuit,” as claimed. *Baumgartner* teaches “the heart of the SLA is a circuit that permits the fast approximate computation of a large collection of values [] for a fixed observation vector *v*.” *Baumgartner*, [0043]. *Baumgartner* teaches the SLA computations are later used in *Baumgartner*’s trellis match to “perform[] a search and select[] a best word or phrase 145.” *Baumgartner*, [0038].

188. A POSITA would have understood that *Baumgartner*’s SLA comprises a “processor,” as recited in Claim 1(e). For example, *Baumgartner* teaches

Broadly, Speech Label Accelerators (SLAs) are provided
that comprise an indirect memory, atom value memory,

and adder circuitry. Optionally, the SLAs also comprise an accumulator, a load/accumulate multiplexer (mux), and a control unit. As explained in more detail below, the indirect memory acts to select atom values from the atom value memory. The atoms are added to determine an approximate value for a kernel or the logarithm of a kernel. Consequently, the SLA allows fast and efficient computation of kernels. The kernels may be subsequently used to determine leaf values, which are then used to determine a word or phrase for a segment of speech.

Baumgartner, [0008], [0043]. A POSITA would have understood the exemplary components and functionality described in relation to *Baumgartner*'s SLA to mean *Baumgartner*'s SLA comprises a processor. I also note that a POSITA would have understood *Baumgartner* teaches the computationally heavy mathematical processing would have been performed by a processor. *Baumgartner*, [0003], [0005], [0007], [0033]. *Baumgartner* teaches speeding processing during the labeling stage of speech recognition and to free "one or more processors from performing labeling." *Baumgartner*, [0007]. Therefore, the SLA—including the circuit for determining probabilities—comprises a "processor," as claimed. Further, for all the reasons explained in the Section, *Baumgartner* teaches the claimed "calculating circuit is implemented using a second processor."

e) ***Baumgartner* Teaches a Search Stage “implemented using a first processor”**

189. In my opinion, *Baumgartner* teaches a search stage implemented on the same “first processor” as the front end, as claimed. Specifically, *Baumgartner*’s Figure 19 teaches final selection and rank mapping 1935 performed by processor core 1910. *Baumgartner*, Fig. 19.

190. As I explained above, *Baumgartner* teaches a search stage (i.e., trellis match 140) as was well-known in speech recognition systems. *Baumgartner*, [0033], [0038], Fig. 1. This trellis match takes the output of *Baumgartner*’s SLA (i.e., labeler 130) and performs a search to select the best word or phrase. *Baumgartner*, [0038], Fig. 1. As I also explained above, *Baumgartner* teaches Figure 19 is an “extension and refinement[.]” of one “of the wide variety of hardware designs that are possible” for the basic, well-known speech recognition system depicted in *Baumgartner*’s Figure 1 (which as explained above, teaches a front end process, a distance calculation process, and a word search process). *Baumgartner*, [0033], Figs. 1, 19. Therefore, a POSITA would have understood that *Baumgartner*’s Figure 19 is a specific hardware architecture of the well-known speech recognition functionality described in *Baumgartner*’s Figure 1. In *Baumgartner*’s Figure 19, *Baumgartner* teaches that after the density computation (i.e., the calculated distances) on the SLA are calculated, “processor core 1910 performs step 1935 where it performs a final selection and rank mapping.” *Baumgartner*, [0157]. A POSITA would have

understood that *Baumgartner's* search functionality, as expressly taught by *Baumgartner's* Figure 1, occurs on processor core 1910 in *Baumgartner's* Figure 19. *Baumgartner*, [0033], [0206] (“The final summation values written back to system memory by the SLA core are used by software to determine the best match or to define the next set of indices to be applied to further refine the search for the best match.”), [0210] (“the summation computations will be done multiple times for up to three search level”). *Baumgartner's* Figure 19 shows a processor core and an SLA (which as explained above teach the claimed “first processor” and “second processor,” respectively), where all speech recognition functionality occurs on processor core 1910 with the exception of the density (i.e., “distance”) computations occurring on the SLA. Accordingly, a POSTIA would have understood step 1935 describes the functionality of trellis match 140 (i.e., the claimed “search stage”) and selects a word or phrase, or renders such obvious. *Baumgartner*, [0033], [0038], [0157], Figs. 1, 19.

191. Because a POSITA would have understood *Baumgartner's* processor core 1910 performs a search through a trellis and because *Baumgartner's* processor core 1910 teaches the claimed “first processor,” it is my opinion that *Baumgartner* teaches or renders obvious a search stage “implemented using a first processor,” as claimed.

f) Motivation to Combine *Jiang* and *Baumgartner*

192. In my opinion, a POSITA would have been motivated and found it obvious to modify *Jiang* such that *Jiang*'s functionality—which maps to the claimed “audio front end,” “calculating circuit,” and “search stage” (as explained above in Sections IX.A.1-IX.A.5)—would be implemented on the hardware architecture taught by *Baumgartner*'s Figure 19. Specifically, as explained below, a POSITA would have been motivated to implement *Jiang*'s signal processing (feature extraction) and word identification (search) functionalities on a first processor and also implement *Jiang*'s probability computations on a second processor.

193. A POSITA would have been motivated to make this simple modification for several reasons. For example, *Jiang* teaches its speech recognition system may be “practiced with other computer system configurations, including ... multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, main-frame computers, and the like.” *Jiang*, 4:60-65. A POSITA would have understood and recognized these disclosures to mean that *Jiang* contemplated different hardware modifications to its system, including, for example, the use of two separate processors.

194. A POSITA would have understood *Jiang*'s threshold and pruning discussion and recognized that *Jiang* identified reasons to “increase in computational savings.” *Jiang*, 10:65–11:25. Specifically, *Jiang* explains that after computing

scores for each node, the scores are compared to a predetermined threshold level and removed (or pruned) if they fall below said threshold level. *Jiang*, 10:65–11:25. Even though *Jiang* still describes calculating all scores initially (for each state for example), *Jiang* explains that pruning thereafter saves on computational resources. *Jiang*, 10:65–11:25. *Jiang*'s threshold and pruning discussion on computational savings would thus further motivate a POSITA to implement a hardware architecture that provides additional computing power. The reason is due to the tradeoff between pruning and accuracy. While pruning saves on computational resources, pruning also decreases accuracy, as branches of the prefix tree are removed. Therefore, a POSITA would have been motivated to find another way to save on computational resources while pruning, thus achieving both increased accuracy and saving computational resources. An obvious consideration would have been a hardware architecture which utilizes a separate processor for the computationally heavy distance (or probability) computations.

195. *Jiang* already contemplates that *Jiang*'s feature extraction module and *Jiang*'s tree search engine (performing distance calculations and tree searching) are executed on separate processors. *Jiang*, 6:32-36, 6:39-42, 6:53-57. Regarding *Jiang*'s feature extraction, *Jiang* specifically teaches feature extraction module 66 “may be either [a] hardware module[] in computer 20, or [a] software module[] stored in any of the information devices disclosed in FIG. 1 and accessible by CPU

21 or another suitable processor.” *Jiang*, 6:32-36, 6:39-42, 6:53-57. A POSITA would have realized that *Jiang* envisions dedicated or separate processors for performing specific speech recognition functionality.

196. Given that a POSITA would have understood *Jiang*’s disclosures to contemplate various hardware architectures, a POSITA looking to design a speech recognition system at the time of the ’277 Patent would have looked to additional sources—leading a POSITA to *Baumgartner*.

197. Regarding *Jiang*’s tree search engine performing distance calculations (as I explained above in Section IX.A.4.b.4), a POSITA would have been motivated to modify *Jiang* such that *Jiang*’s tree search engine performing distance calculations was performed on its own processor—distinct from the processor executing *Jiang*’s feature extractor and *Jiang*’s tree search engine searching for the best word. For example, *Jiang* repeatedly refers to distinct processors for performing the functionalities of speech recognition. *See, e.g., Jiang*, 6:39-42 (“Further, tree search engine 74 is preferably implemented in CPU 21 (which may include one or more processors) or may be performed by a dedicated speech recognition processor employed by personal computer 20.”), 6:53-57 (stating feature extraction module 66 is a “conventional array processor”), 7:19-21 (stating silence detection module 68 “can either be implemented on the same, or a different processor as that used to implement the feature extraction module 66”).

198. A POSITA would have also been aware of *Ravishankar*. A POSITA would have understood *Ravishankar* emphasized the need for efficiency, reduced computational footprint, speed, vocabulary size, and accuracy in commercial speech recognition systems. *Ravishankar*, 5, 17-18, 65-67, 105-107. A POSITA looking to build a commercial speech recognition system would have been acutely aware of CMU's Sphinx-II system, which *Ravishankar* used as a baseline. *Ravishankar*, 5.

199. A POSITA would have recognized the Sphinx-II system operated on university lab general purpose computers. *Ravishankar*, 5, 18, 23. A POSITA would have also recognized *Ravishankar's* teachings and been interested in taking the lab-version Sphinx-II and cutting down its computational footprint—without sacrificing vocabulary size, speed, or accuracy—to create a commercial system. *Ravishankar*, 5. For example, *Ravishankar* teaches the practical need for sufficient computational resources in large vocabulary continuous speech applications as then-commonly available computers were insufficient given an increase in computational loads for large vocabulary applications. *Ravishankar*, 5, 33-34, 55-56, 65-66. A POSITA would have particularly been interested in *Ravishankar's* teachings on concurrency and would have noted the reference to multiprocessor architectures. *Ravishankar*, 66-67, 105-109. A POSITA would have recognized that *Ravishankar* exemplified the speech recognition landscape in the 1990s—where systems were evolving from speaker-dependent small vocabulary systems to speaker-independent continuous

large vocabulary systems—which was occurring not only because humans speak with a larger vocabulary which varies from speaker to speaker, but also because it would open up a wide range of applications including dictation, browsing, and vehicle navigation. *Ravishankar*, 5, 17-18.

200. A POSITA would have understood that calculating distances of a feature vector as compared to an acoustic state was well-known to be a computationally-intensive task. *Baumgartner*, [0003]-[0007], [0200]-[0205]; *Mathew I* (Ex. 1050), 1, 4; *Mathew II* (Ex. 1051), 1, 2. Specifically, A POSITA would have been aware of the ongoing discussion of implementing a Gaussian probability accelerator in the Sphinx 3—the successor to the Sphinx-II discussed in *Ravishankar*. For example, two articles from the University of Utah’s School of Computing in 2003 were expressly directed to offloading the computationally expensive calculations used in speech recognition systems to an accelerator: “A Gaussian Probability Accelerator for SPHINX 3” by Binu K. Mathew et al. (“*Mathew I*”) and “A Low-Power Accelerator for the SPHINX 3 Speech Recognition System” by Binu K. Mathew et al. (“*Mathew II*”). As their titles suggest, Mathew investigated and contemplated the advantages of using an accelerator to improve the efficiency of speech recognition systems. *Mathew I*, 1; *Mathew II*, 1. Specifically, Mathew found the Gaussian probability calculation to be a computationally expensive task in relation to the front end processing. *Mathew I*,

2-4; *Mathew II*, 2. Accordingly, a POSITA would have been motivated to use a separate processor to calculate distances, as taught by *Baumgartner* and *Baumgartner's* SLA, for the distance calculation of *Jiang's* speech recognition system.

201. Given that a POSITA would have understood *Jiang* already contemplates different and various hardware architectures and also understood the current landscape of speech recognition systems as explained by *Ravishankar* and additional sources, a POSITA looking to design a speech recognition system at the time of the '277 Patent would have looked to additional sources—leading a POSITA to *Baumgartner*. *Baumgartner* explains “[s]peech recognition ... is currently a computationally intensive task,” that “it would be beneficial to reduce the load on the processors of the system when the system performs speech recognition,” and “a need exists for techniques to reduce the computational load of a processor during speech recognition.” *Baumgartner*, [0003]-[0006]. A POSITA would have recognized *Baumgartner's* “invention provides Speech Label Accelerators (SLAs), ... speeds processing during the labeling stage of a speech recognizer[,] and frees one or more processors from performing labeling.” *Baumgartner*, [0007]. A POSITA would have further understood that *Baumgartner's* hardware architecture—utilizing a separate processor for the computationally-expensive distance calculations—provides an efficient solution to the computational load

issues identified by the prior art and that may have arisen due to pruning. *Ravishankar*, 5, 33-34, 55-56, 65-66; *Baumgartner*, [0200]–[0205] (“Because the SLA implements the time-consuming portion of the labeling algorithm in hardware, it is possible to contemplate recognition systems with much more computationally demanding acoustic models ... it is believed the SLA may enable the construction of speech recognition systems that are at once faster, more power efficient, and substantially more accurate ... the SLA 500 will be used to speed up the processing required in a speech label algorithm for acoustic modeling in speech recognition.”).

202. A POSTIA would have had a reasonable expectation of success without undue experimentation in modifying *Jiang*’s system to utilize *Baumgartner*’s SLA for calculating distances and implementing both *Jiang*’s feature extraction and search on a separate processor at least because *Baumgartner*, *Mathew I*, and *Mathew II*, illustrated such feasibility. Further, the use of separate processors, such as *Baumgartner*’s SLA, co-processors, or math-processors, was a well-known technique to offload computational tasks in general prior to the ’277 Patent, as I explained above in Section V.G. The determination of whether to use a separate processor for computationally expensive tasks—as opposed to using a sole processor for the entire system—would have been a simple design choice well within the skill set of a POSITA. Overall, given the wide-spread use of separate processors for computationally expensive tasks at the time, and the specifically known use of

separate processors for distance calculations in speech recognition system, a POSITA would have been able to implement *Baumgartner's* SLA into *Jiang's* speech recognition system without undue experimentation.

7. *Claim 1(f): "wherein data is pipelined from the front end to the calculating circuit to the search stage."*

203. The *Jiang-Baumgartner* system teaches or at least renders obvious transmitting data from a front end to a calculating circuit to a search stage. *See* Sections IX.A.1-IX.A.5. For example, *Jiang* teaches feature extraction module 66 takes an audio speech signal and converts the audio speech signal into a feature vector. *Jiang*, 6:50–7:10. “Thus feature extraction module 66 provides, at its output the feature vectors (or codewords) for each spoken utterance.” *Jiang*, 7:5-7. *Jiang* next teaches calculating distances (i.e., calculating a score) based on the feature vector (as output by feature extraction module 66) based on the similarity between the feature vector and states of an acoustic model. *Jiang*, 8:29-33 (the “score represents the likelihood that the particular phoneme encoded by the codeword corresponds to the phoneme for the branch under consideration.”); *see also Jiang*, 7:30-47 (providing additional explanation of how the score for each phoneme branch is determined based on determining a most likely phoneme for a codeword (i.e., feature vector) as compared to a plurality of acoustic states of one or more HMMs). *Jiang* next teaches using those calculated distances (scores) to indicate the mostly

phoneme for a particular node and associated phoneme branch to identify words as *Jiang*'s tree search engine traverses *Jiang*'s prefix tree. *Jiang*, 8:16-43.

204. I note the *Jiang-Baumgartner* system does not expressly teach data is pipelined. However, as discussed in detail below, the *Jiang-Baumgartner* system in further view of *Brown* teaches Claim 1(f).

205. I note the '277 Patent describes pipelining data in at least one embodiment as a "time axis is divided into equal 'pipeline step times'." '277 Patent, 26:60. "Once the pipeline has filled (which takes two step times), the DSP, the Accelerator, and the CPU are all operating in parallel. In this idealized example, the total time available for processing a frame is three step times—one step time in the DSP, one step time in the accelerator, and one step time in the search stage." '277 Patent, 27:3-8. Further, the '277 Patent describes a "non-pipelined system": "[a] non-pipelined system would be one in which the processing of frame n+1 would not start until all stages of the processing of frame n was already complete." '277 Patent, 27:17-20, 26:52-27:38, 29:16-28, 29:42-67, 30:1-20. The '277 Patent also explains "FIG. 21 shows a pipelined system[.]" '277 Patent, 30:41. The '277 Patent's discussion of pipelining is consistent with the pipelining that was well-known in the art, both in general and specifically in speech recognition. See Section V.H above.

a) Brown's Teachings

206. I note that *Brown* teaches a speech recognition system 100 that includes functional blocks performing speech recognition, such as feature extraction, phone probability processing, and word probability processing. *Brown*, 3:32-41. *Brown* teaches these functional blocks may be pipelined:

The functionality of each block of the recognition system may be performed by a respective different processor, or the functionality of several or all blocks may be performed by the same processor. ... The **stages are pipelined and their operations are synchronized with the rate at which frames of the incoming speech signal are generated.**

Brown, 3:41-48.

207. In my opinion, a POSITA would have understood that *Brown*'s teachings of pipelining teaches or renders obvious Claim 1(f). Specifically, a POSITA would have recognized *Brown*'s pipelining "synchronize[s] with the rate at which frames of the incoming speech signal are generated." *Brown*, 3:41-48.

208. In my opinion, *Brown*'s teaching of pipelining and synchronizing operations at the rate at which frames of the incoming speech signal are generated teaches or renders obvious the claimed "data is pipelined from the front end to the calculating circuit to the search stage," as claimed and as described in the '277 Patent. *Brown* discloses that as the incoming speech signal is generated, frames are

responsively generated for each segment of speech, commonly 10 ms. *Brown*, 3:41-48. *Brown* discloses that “[t]he incoming speech signal is sampled in a sequence of overlapping frames during a corresponding sequence of frame periods.” *Brown*, 3:56-69. Feature extraction is then performed. *Brown*, 3:58-61. Then a probability score indicating the probability a respective phoneme was uttered during the speech signal frame period is computed. *Brown*, 3:66–4:12. Then, words are determined using HMMs. *Brown*, 4:13-25.

209. *Brown* discloses the functional stages are pipelined (in addition to other functional stages referenced at *Brown*, 3:32-40). In my opinion, this means that that as the frames of data are generated, the data is processed and passed from one stage to another so that the operations are synchronized with the rate of frame generation. The '277 Patent explains pipelining in relation to a “time axis” and “step times,” as I quoted above. '277 *Patent*, 26:60, 27:3-8. A POSITA would have recognized that *Brown*'s time synchronization for pipelined steps describes the same concept. Both can be thought of in terms of time axes and step times, where the time axes are divided into equal portions, each representing the time it takes for data to be processed in the three well-known speech recognition steps. This equal division allows for the speech recognition system to pipeline the three well-known speech recognition steps into equal pipelined steps. Further, such a pipelined system allows different processors (performing the different steps of speech recognition) to operate

concurrently or in parallel. In *Brown*, for example, the operations of each of the functional stages (e.g., feature extraction, phone probability processing, etc.) are synchronized with the frame generation rate. Thus, for example, after feature extraction is performed on a first frame *N*, the next functional stage begins the next processing step on frame *N*, and in parallel (i.e., at the same time because the operations are synchronized with the frame generation rate), feature extraction is performed on a second frame, *N*+1. This process continues in a synchronized manner. I note this is consistent with the well-known method of pipelining data as I described above in Section V.H.

b) Motivation to Combine *Brown* with *Jiang-Baumgartner*

210. In my opinion, a POSITA would have been motivated and found it obvious to modify the *Jiang-Baumgartner* system to pipeline data as taught by *Brown*. For example, a POSITA would have understood and found obvious that pipelining data offered numerous, well-known advantages. As I explained above in Section V.H, these advantages include greater overall throughput of the processor, improved performance, decreased cost, and simplified programming. *See* Section V.H (citing *Luick*, *Simar*, and *Brown*). A POSITA would have understood that utilizing *Brown*'s pipelining teachings would have improved the *Jiang-Baumgartner* system in the same way that pipelining improved *Brown*'s speech recognition system and would have improved a computer system in general as was well-known in the

art. *See* Section V.H (citing *Luick*, *Simar*, and *Brown*). Such well-known improvements would have included keeping the phases of processing data in sync as well as allowing for concurrent and parallel processing. *Id.*

211. Further, a POSITA would have realized that *Jiang*, *Baumgartner*, and *Brown* teach similar speech recognition systems including utilizing distinct processors for specific speech recognition system tasks. *Jiang*, 4:60-65, 6:39-57, 7:19-21; *Baumgartner*, [0033]-[0043], [0155]-[0158], Figs. 1, 19; *Brown*, 3:35-48.

212. In my opinion, a POSITA would have recognized that *Jiang* teaches the three distinct stages for feature extraction, probability computation, and word search, as explained above. *See* Sections IX.A.1-IX.A.5. A POSITA would have also understood *Brown* teaches pipelining separate speech recognition stages where the output from a prior stage is used as the input to the next stage. *Brown*, 3:32-48. Because speech data is received in exemplary 10 ms increments, a POSITA would have recognized a steady data input and need to process data efficiently. Indeed, a POSITA would have understood and found obvious pipelining data between the three well-known speech recognition stages occurring on different processors (for example as taught by the *Jiang-Baumgartner* system) to keep overall processing efficient with respect to the exemplary 10 ms time intervals. In other words, pipelining between the speech recognition stages on separate processors keeps each processor working efficiently. For example, assume a frame N. As feature vector

extraction completes on frame N, data is passed to the processor performing probability (i.e. distance) computations. While probability computations are being performed on frame N, the processor extracting feature vectors is operating on data for the next frame N+1. This allows the first processor performing feature vector extraction and word identification to remain busy and avoid idle, wasted time.

213. As such, a POSITA would have understood the need for and been motivated to design systems that keep all processors busy whenever possible, as time is important in pipelined systems with defined time intervals per frame. As previously explained, *Jiang* already discusses the importance of time with respect to pruning. *Jiang*, 10:65–11:25. As also explained, the *Jiang-Baumgartner* system accounts for increased processing when utilizing pruning, thus achieving both higher accuracy and more efficient processing. Pruning saves on computational resources but may reduce accuracy. Distributing the computational load between different processors reduces the need to save on computational loads in any given processor thus allowing for greater accuracy than would be achieved on a sole processor. A POSITA would have realized pipelining keeps all processors operating efficiently in a multi-processor system. Given a defined frame time (for example 10ms), pipelining keeps all processors working efficiently by avoiding potentially wasted processing time, or idle/busy wait time. This ensures that idle-processing time is minimized, thus improving the speech recognition system and

functionality/accuracy of the speech recognition system. In my opinion, a POSITA would have recognized pipelining, as taught by *Brown*, is one such way to avoid wasted computational resources as *Brown* expressly teaches synchronization “with the rate at which frames of the incoming speech signal are generated.” *Brown*, 3:44-48.

214. A POSITA would have had a reasonable expectation of success without undue experimentation in modifying the *Jiang-Baumgartner* system to utilize pipelining as taught by *Brown*. As I explained above, pipelining was well-known in the art in general and specifically in speech recognition systems. *See* Section V.H. Because pipelining was well-known in processing in general and specifically in speech recognition for the advantages described above, a POSITA would have had a reasonable expectation of success in further modifying the *Jiang-Baumgartner* system to utilize pipelining as taught by *Brown*. A POSITA would have further realized that *Jiang*, *Baumgartner*, and *Brown* teach speech recognition systems using similar hardware, such as CPUs, processors, accelerators, to perform similar speech recognition functions. *Jiang*, 4:60-65, 6:39-57, 7:19-21; *Baumgartner*, [0033]-[0043], [0155]-[0158], Figs. 1, 19; *Brown*, 3:11-15 (“[T]he illustrative embodiments of the present invention are presented as comprising individual functional blocks. The functions these blocks represent may be provided through the use of either shared or dedicated hardware, including, but not limited to, hardware capable of

executing software. Illustrative embodiments may comprise, for example, digital signal processor (DSP) hardware[.]”), 3:35-48. Given the similarity of these speech recognition systems and specific hardware utilized to perform the well-known functional stages of speech recognition, a POSITA would have had a reasonable expectation of success to further modify the *Jiang-Baumgartner* system to utilize the pipelining functionality taught by *Brown*. Additionally, as I explained above, a POSITA would have understood and found obvious that implementing pipelining would simplify programming. *See* Section V.H (citing *Simar*, 20:22-23). Because pipelining was well-known both in general and specifically in speech recognition and because pipelining was known to have the advantages described above, a POSITA would have had a reasonable expectation of success without undue experimentation in modifying the *Jiang-Baumgartner* system to utilize pipelining as taught by *Brown*.

B. Dependent Claim 5

1. ***“A speech recognition circuit as claimed in claim 1, wherein the said calculating circuit is configured to autonomously calculate distances for every acoustic state defined by the acoustic model.”***

215. *Jiang* teaches “[a] probability score is computed for **each state** in the state sequences.” *Jiang*, 2:4-6. *Jiang* also teaches its HMMs “include a plurality of states, wherein a transition probability is defined for **each transition from each**

state to every state, including transitions to the same state.” *Jiang*, 1:21-25. In reference to *Jiang*’s prefix tree, *Jiang* teaches:

This type of structure lends itself to a number of significant advantages. For example, given the small number of initial branches in the tree, it is possible to consider the beginning of all words in the lexicon, even if the vocabulary is very large, **by evaluating the probability of each of the possible first phonemes**.

Jiang, 3:10-15. And as I explained above in Section IX.A.4 regarding Claim 1(c), *Jiang* and *Baumgartner* each teach a “calculating circuit for calculating distances,” as claimed. A POSITA would have understood that *Jiang* teaches calculating distances (or probabilities) for each state.

216. A POSITA would have also understood that *Jiang* teaches calculating such distances “autonomously.” For example, *Jiang*’s tree search engine 74 first receives codewords from feature extraction module 66. *Jiang*, 7:30-42. *Jiang*’s tree search engine 74 next calculates probability scores, performing the likelihood indication I explained above in Section IX.A.4.b. *Jiang*’s tree search engine 74 next access HMMs and determines a most likely phoneme based on the probability score for each state in the state sequence. *Jiang*, 7:30-42. I note that this occurs on two separate processors in the *Jiang-Baumgartner-Brown* system as explained in detail in Section IX.A.6. Accordingly, a POSITA would have understood that *Jiang*’s tree

search engine 74 autonomously performs the distance calculation for every state after receiving codewords.

217. A POSITA would have understood that because *Jiang* teaches calculating distances (i.e., *Jiang*'s probability score) for every state after receiving the codewords, the *Jiang-Baumgartner-Brown* system of Claim 1 further teaches, or at least renders obvious, Claim 5.

C. Dependent Claim 7

- 1. “*The speech recognition circuit of claim 1, wherein the feature vector comprises a plurality of spectral components of an audio signal for a predetermined time frame.*”**

218. As I explained in Sections IX.A.2-IX.A.3 above, *Jiang*'s feature vector that is output by *Jiang*'s feature extraction module 66 comprises a plurality of spectral components, such feature vector and associated spectral components are from an audio signal, and such feature vector and associated spectral components relate to frames corresponding to predetermined time periods of 10 milliseconds. *See* Sections IX.A.2-IX.A.3 (citing *Jiang*, 6:62–7:1, 6:64-65, 7:7-10).

219. In my opinion, a POSITA would have understood that because *Jiang*'s digital audio signal is sampled at a rate of 10 milliseconds per frame, the 10 millisecond time frame is “predetermined,” as claimed, such that *Jiang*'s feature extraction module 66 performs feature extraction every 10 milliseconds.

220. Accordingly, in my opinion, the *Jiang-Baumgartner-Brown* system of Claim 1 further teaches, or at least renders obvious, Claim 7.

D. Dependent Claim 12

1. ***“The speech recognition circuit of claim 1, wherein the audio front end is configured to input a digital audio signal.”***

221. As I explained above, *Jiang*’s feature extraction module 66 teaches the claimed “audio front end.” See Section IX.A.2. I note that *Jiang* further teaches speech is input into system 60 and converted into a digital signal by *Jiang*’s A/D converter 64 where such digital signal is then provided to *Jiang*’s feature extraction module 66. *Jiang*, 6:46-53.

222. Accordingly, in my opinion, the *Jiang-Baumgartner-Brown* system of Claim 1 further teaches, or at least renders obvious, Claim 12.

E. Independent Claim 14

1. ***Claim 14(Pre): “A speech recognition circuit, comprising:”***

223. See Section IX.A.1 regarding Claim 1(Pre).

2. ***Claim 14(a): “an audio front end for calculating a feature vector from an audio signal,”***

224. See Section IX.A.2 regarding Claim 1(a).

3. ***Claim 14(b) “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”***

225. See Section IX.A.3 regarding Claim 1(b).

4. Claim 14(c): “calculating means for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and”

226. See Section IX.A.4 regarding Claim 1(c), which explains the mapping for the claimed functionality of Claim 14(c). As I explained above, I have been informed that “calculating means” should be construed as a means-plus-function limitation and provided my opinion that the structure for performing the functionality claimed in Claim 14(c) is at least a processor. See Section III.D.1.

227. In my opinion, *Jiang’s* dedicated processor and/or CPU, which comprises a processor, teaches the claimed “calculating means” by teaching a processor performing the functionality of Claim 14(c). *Jiang’s* tree search engine may be implemented by a dedicated (as explained above in Section IX.A.4 for Claim 1(c)) and therefore teaches or renders obvious Claim 14(c). *Jiang*, 6:39-42, 8:16-24. *Jiang* also teaches that *Jiang’s* tree search engine is implemented on a CPU, and a POSITA would have understood a CPU comprises a processor. *Jiang*, 6:39-42, 8:16-24.

228. Additionally, as I explained in Sections IX.A.4 and IX.A.6, in my opinion, the *Jiang-Baumgartner* system teaches a processor for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model. In my opinion, *Baumgartner’s* SLA, which comprises a processor, teaches the claimed “calculating means.”

5. ***Claim 14(d): “a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words;”***

229. See Section IX.A.5 regarding Claim 1(d).

6. ***Claim 14(e): “wherein said audio front end, said calculating means, and said search stage are connected to each other to enable pipelined data flow.”***

230. See Section IX.A.7 explaining Claim 1(f) regarding the “pipelined data flow.”

231. Additionally, in my opinion, the *Jiang-Baumgartner-Brown* system further teaches Claim 14(e). *Baumgartner* teaches the components of Figure 19 are “all interconnected through bus 1948.” *Baumgartner*, [0156]. In my opinion, a POSITA would have understood that the components of *Baumgartner*’s Figure 19 (which, as explained above in Section IX.A.6, teach a front end, a calculating circuit/processor, and a search stage) which are interconnected through *Baumgartner*’s bus 1948 teaches or at least renders obvious “wherein said audio front end, said calculating means, and said search stage are connected to each other,” as claimed.

F. Independent Claim 15

1. ***Claim 15(Pre): “A speech recognition method, comprising:”***

232. See Section IX.A.1 regarding Claim 1(Pre). A POSITA would have understood that *Jiang*’s discussion of computerized steps or functionalities of speech

recognition teaches a “method,” as claimed. *Jiang*, 1:15-19, 2:8-10, 4:8-9, 4:30-32, 4:51–6:20, Abstract, Figs. 1-2, Claim 1.

2. *Claim 15(a): “calculating a feature vector from an audio signal using an audio front end,”*

233. *See* Section IX.A.2 regarding Claim 1(a). In my opinion, a POSITA would have understood the disclosures in Section IX.A.2 regarding in Claim 1(a) teach Claim 15(a).

3. *Claim 15(b): “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”*

234. *See* Section IX.A.3 regarding Claim 1(b).

4. *Claim 15(c): “calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model using a calculating circuit; and”*

235. *See* Section IX.A.4 regarding Claim 1(c) (explaining the proposed mapping teaches both distances (plural) and a distance (singular) to acoustic states and an acoustic state, respectively). A POSITA would have understood the disclosures in Section IX.A.4 regarding Claim 1(c) teach Claim 15(c).

5. *Claim 15(d): “using a search stage to identify words within a lexical tree using said calculated distances, the lexical tree comprising a model of words;”*

236. *See* Section IX.A.5 regarding Claim 1(d). A POSITA would have understood that the disclosures in Section IX.A.5 regarding Claim 1(d) teach Claim 15(d).

6. ***Claim 15(e): “wherein data is pipelined from the front end, to the calculating circuit, and to the search stage.”***

237. See Section IX.A.7 regarding Claim 1(f).

G. Independent Claim 16

1. ***Claim 16(Pre): “A non-transitory storage medium storing processor implementable code for controlling at least one processor to implement a speech recognition method, the code comprising:”***

238. See Section IX.A.1 regarding Claim 1(Pre) (discussing speech recognition), Section IX.A.6 regarding Claim 1(e) (discussing processors), and Section IX.F.1.a for Claim 15(Pre) (discussing a speech recognition method). I note that *Jiang* teaches its invention may be implemented “in the general context of computer-executable instructions,” and “[t]he drives and the associated computer-readable media provide nonvolatile storage of computer readable instructions.” *Jiang*, 4:54-56, 5:5-29. A POSITA would have understood *Jiang*’s references to system memory 22, ROM 24, and RAM 25 represent examples of “non-transitory medium,” as claimed as such was well-known in the art. *Jiang* also teaches tree search engine 74 “may be performed by a dedicated speech recognition processor employed by personal computer 20.” *Jiang*, 6:39-42. A POSITA would have understood that to be “employed by personal computer 20,” the speech recognition functionality of *Jiang* would utilize code to control the processor. *Jiang*, 6:39-42. Additionally, a POSITA would have understood that speech recognition methods utilize storage mediums and computer code to implement functions on processors.

Ravishankar, 5, 18-20, 49, 56, 79, Figs. 3.5, 4.6 (collectively showing that speech recognition functionality code would be stored in memory and executed by a processor).

239. Accordingly, *Jiang*'s examples of non-transitory storage medium (e.g., system memory 22, ROM 24, RAM 25) teach the claimed "non-transitory storage medium," and *Jiang*'s computer-executable instructions stored thereon teach the Claim 16(Pre), to the extent Claim 16(Pre) is limiting.

2. *Claim 16(a): "code for controlling the processor to calculate a feature vector from an audio signal,"*

240. See Section IX.G.1 regarding Claim 16(Pre), Section IX.A.2 regarding Claim 1(a), and Section IX.A.6 regarding Claim 1(e).

241. A POSITA would have understood that *Jiang*'s speech recognition system includes "code for" performing the functionality of Claims 16(a)-16(d) because *Jiang* teaches computer-executable instructions for performing the disclosed functionality, and it was well-understood in the art to use code for controlling processors to implement functionality in speech recognition methods. See, e.g., *Ravishankar*, 5, 18-20, 49, 56, 79, Figs. 3.5, 4.6.

3. *Claim 16(b): "wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;"*

242. See Section IX.G.1 regarding Claim 16(Pre), Section IX.A.3 regarding Claim 1(b), and Section IX.A.6 regarding Claim 1(e).

4. *Claim 16(c): “code for controlling the processor to calculate a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and”*

243. See Section IX.G.1 regarding Claim 16(Pre), Section IX.A.4 regarding Claim 1(c), and Section IX.A.6 regarding Claim 1(e).

5. *Claim 16(d): “code for controlling the processor to identify words within a lexical tree using said calculated distances, the lexical tree comprising a model of words,”*

244. See Section IX.G.1 for Claim 16(Pre), Section IX.A.4 regarding Claim 1(d), and Section IX.A.6 regarding Claim 1(e).

6. *Claim 16(e): “wherein data is pipelined by the processor pursuant to the code from the feature calculation, to the distance calculation, and to the word identification.”*

245. See Section IX.G.1 regarding Claim 16(Pre), Section IX.A.6 regarding Claim 1(e), and Section IX.A.7 regarding Claim 1(f).

X. OPINIONS REGARDING GROUND 2: CLAIM 4 IS OBVIOUS OVER JIANG, BAUMGARTNER, BROWN, AND KAZEROONIAN

A. Dependent Claim 4

1. *Claim 4: “A speech recognition circuit as claimed in claim 1, wherein the first processor supports multi-threaded operation, and runs the search stage and front ends as separate threads.”*

246. I have been informed by counsel to interpret the claimed “front ends” in Claim 4 as the claimed “front end” in Claim 1, because Claim 1 does not recite plural front ends.

247. *Kazeroonian* teaches a pipelined and multithreaded system which processes audio and video data in real time. *Kazeroonian* (Ex. 1037), Abstract, 15:33-16:1, 19:30–20:5, 21:9-11, Fig. 2.

248. Specifically, a POSITA would have understood that *Kazeroonian* teaches a single processor performing multi-threaded operations with separate threads in the speech recognition context. For example, *Kazeroonian* teaches an audio-video processing server (AVPS) 101 performing speech recognition. *Kazeroonian*, 15:33–16:1 (“The system can also include a speech recognizer. This speech recognizer takes, as an input, the audio portion of an audio-video program and produces a word sequence corresponding to that audio data.”), Fig. 2 (including “speech recognizer 226”). *Kazeroonian* teaches:

[AVPS] 101 can be a multi-processor allowing parallel processing of the modules and tasks shown in Figs. 2 and 3. The tasks and processing steps are divided between separate processing threads[.] ... AVPS 101 uses threads to achieve real-time video encoding throughput, high segmentation speed, and other characteristics that affect user interface responsiveness. **The pipelined architecture** supported by these multiple threads provides the end-users with access to the indexed videos only moments after the videos are received. [¶] Even on a **single processor system multi-threading can improve performance** by judicious assignment of various tasks (stages) to **separate threads**.

Kazeroonian, 19:26–20:5. *Kazeroonian* also teaches “different processor threads are assigned to different stages of processing.” *Kazeroonian*, 21:9-11. Accordingly, *Kazeroonian* teaches a single processor performing multi-threaded operations with separate threads.

249. A POSITA would have understood the *Kazeroonian* disclosures relate to an audio-video processing server. In my opinion, a POSITA would have understood that such disclosures are also applicable to systems only processing audio (and not processing video), for example such as speech recognition systems. In my opinion it would have been less complex to implement *Kazeroonian*’s audio-video teachings on an audio-only system as there would be less overall data to process. For example, when only processing audio, there is no need to “compress[] or encod[e] the video data,” detect or analyze “both the audio and video portions,” compare color distribution, or “comput[e] a statistic of the rate of change of the video signal.” *Kazeroonian*, 5:22, 6:18-21. In my opinion, implementing the multithreading and pipelining teachings of *Kazeroonian* (to the extent such are taught in reference to video processing) to audio processing would have been performed in a simpler fashion.

250. In my opinion, a POSITA would have been motivated to modify the *Jiang-Baumgartner-Brown* system to use a multithreading operation (and specifically the assignment of various tasks described by *Kazeroonian*) in order to

avoid data loss. For example, the front end of the speech recognition system would need to run in real-enough time to complete processing a current frame (i.e., feature vector extraction) before the next frame arrives (i.e., in the 10 millisecond interval) or data would be lost. In other words, there would be real time data at a given frame rate that must be processed by the front end in real time in order to keep the additional tasks (particularly the word search also occurring on the same processor) processing at the correct time. Multitasking using multithreaded operations and prioritization of tasks for these reasons have been known for decades—in the 1980s, I built a non-preemptive multitasking system to run under DOS for precisely the reasons discussed here.

251. A POSITA would have been further motivated and found it obvious to modify the *Jiang-Baumgartner-Brown* system to utilize a multi-threaded operation—as taught by *Kazeroonian*—for the claimed “first processor” to run the claimed “search stage” and “front end” as separate threads. For example, *Kazeroonian* expressly teaches that such utilization of multithreaded operations provides high throughput (which a POSITA would have understood from *Kazeroonian*’s reference to “achiev[ing] real-time [] encoding throughput”), high segmentation speed, and results “only moments after the videos are received.” *Kazeroonian*, 19:30–20:2. *Kazeroonian* also teaches such multi-threading improves the assignment of processing stages to separate threads. *Kazeroonian*, 20:3–5.

Indeed, *Kazeroonian* teaches “[u]sing threads is especially advantageous when a process consists of multiple tasks of varying urgency and processing needs.” *Kazeroonian*, 20:10-12. A POSITA would have understood and found obvious that when modifying *Jiang* to implement its audio front end and search stage using a first processor as taught by *Baumgartner*, and when modifying the *Jiang-Baumgartner* system such that data is pipelined through the components, as taught by *Brown*, such a modification would have included performing audio front end and search stage tasks in parallel on a single “first processor” via a multi-threaded operation. Use of *Kazeroonian*’s known multi-threading technique (i.e., a well-known technique) would have advantageously improved the similar *Jiang-Baumgartner-Brown* system in the same way it improved *Kazeroonian*’s system.

252. A POSITA would have had a reasonable expectation of success without undue experimentation in modifying the *Jiang-Baumgartner-Brown* system to utilize multithreading of a front end and search stage on the same (first) processor as taught by *Kazeroonian*. For example, multi-threading operations were well known and commonly used in pipelined operations, as exemplified and taught by *Kazeroonian*. *Kazeroonian*, 19:33–20:2 (“The pipelined architecture supported by these multiple threads provides the end-users with access to the indexed videos only moments after the videos are received.”). Further, as mentioned, I built a

multitasking/multithreading system in the 1980s, further evidencing the reasonable expectation of success.

XI. OPINIONS REGARDING GROUND 3: CLAIMS 9-10 ARE OBVIOUS OVER *JIANG, BAUMGARTNER, BROWN, AND VENSKO*

A. Dependent Claim 9

1. *Claim 9: “The speech recognition circuit of claim 1, wherein the speech accelerator has an interrupt signal to inform the front end that the accelerator is ready to receive a next feature vector from the front end.”*

253. In my opinion, *Vensko* teaches “an interrupt signal to inform the front end that the [calculating circuit implemented on the second processor] is ready to receive a next feature vector from the front end.” I have been informed by counsel to interpret “the speech accelerator” in Claim 9 as the “calculating circuit [] implemented using a second processor” in Claim 1, because Claim 1 does not recite a “speech accelerator.”

254. I note the '277 Patent describes the “distance calculation engine may be included within an accelerator.” *'277 Patent*, 3:34-37.

255. I note *Vensko* teaches template processors that perform a dynamic programming algorithm (DPA) including performing a distance calculation between a character distance vector (i.e., feature vector) and an assigned template. *Vensko*, 6:22-29. A master processor causes character distance vectors to be transferred to each of the template processors for comparison. *Vensko*, 6:22-26. “The master processor via an interrupt commands the template processors to begin processing the

character distance vectors on a frame basis.” *Vensko*, 6:26-29. Next, each template processor “**generate[s] an interrupt** to the master processor 160 informing it that the results are ready for transfer.” *Vensko*, 6:29-33, 10:10-19. The template processors, which each generate an interrupt to the master processor, thus signal to the master processor that each template processor is ready to receive the next feature vector. A POSITA would have understood that *Vensko* teaches an interrupt signal informing the master processor results are ready. *Vensko*, 6:48-51. *Vensko* next references a syntax analysis. *Vensko*, 6:48-51. Then *Vensko* teaches “a new character distance vector is sent to the template processors and the process begins all over again until recognition is complete.” *Vensko*, 6:48-51. I note that a POSITA would have understood that in *Vensko*’s speech recognition system, the “character distance vector” is a “feature vector,” as claimed. *Vensko*, 5:41-43 (explaining *Vensko*’s character distance vector is created in *Vensko*’s front end processor), 5:67–6:3 (explaining *Vensko*’s template processors compare “each character distance vector received from the front end processor”).

256. A POSITA would have understood that even though *Vensko* contemplates intervening steps between the interrupt and the next distance vector being sent from the front end processor—for example, such as “syntax analysis” or “reallocat[ion] among the template processors” (*Vensko*, 6:41-48)—that the interrupt still informs the front end that the claimed “second processor” is ready to

receive a next feature vector from the front end. For example, *Vensko* further teaches “[t]o continue, an interrupt of the master processor by a slave processor was described above. The interrupt is used to tell the master that the slave processor which is interrupting is finished a processing task.” *Vensko*, 10:10-13. *Vensko*’s master processor then “decides on what data transfers are to take place” and sets up a data transfer based on the interrupt. *Vensko*, 10:13-19.

257. Accordingly, a POSITA would have understood that, despite any potential intervening tasks or steps, *Vensko*’s interrupt controls when the next feature vector is to be sent to the claimed “calculating circuit” implemented using the claimed “second processor.” In pipelined speech recognition systems, feature vectors are moved to the distance (or probability) determination stage after the previous feature vector has been used to determine distances (or probabilities) per system instructions, such as interrupts. *See, e.g., Luick*, 1:20-37 (“a pipeline may include separate stages for fetching the instruction from memory, executing the instruction, and writing the results of the instruction back into memory.”); *Brown*, 3:32-48 (describing pipelining through the well-known stages of speech recognition). Thus, a POSITA would have understood that regardless of potential intervening steps, *Vensko*’s interrupt is the component which “inform[s] the front end that the [calculating circuit implemented on the second processor] is ready to receive a next feature vector from the front end,” as claimed.

258. A POSITA would have understood that *Vensko*'s teachings of template processors generating an interrupt to the master processor that results are ready for transfer to mean the distance calculation functionality (i.e., the claimed "calculating circuit" implemented using the claimed "second processor") generating an interrupt to the processor executing the front end functionally (i.e., the claimed "first processor") that the distance calculation processor (i.e., the claimed "second processor") is ready for the next frame of data in a pipelined system. Therefore, a POSITA would have understood that the *Jiang-Baumgartner-Brown* system in further view of *Vensko* teaches, or renders obvious, Claim 9.

259. Accordingly, *Vensko* teaches or renders obvious the use of interrupt signals between different processors in speech recognition systems. As explained above, *Vensko* teaches both that the master processor sending an interrupt to the template processors to command processing character distance vectors and that the template processors send interrupts to the master processor to indicate to the master processor when distance results are complete and ready to transfer. *Vensko*, 6:22-33, 10:10-19. In my opinion, a POSITA would have understood that *Vensko* suggests utilizing interrupt commands to instruct processing and also to instruct transfer of data once processing is complete.

260. I additionally note that it was well-known for interrupts to signal to different stages (or processors) in systems when data transfers are ready. *See* Section

V.I (citing *Atkas 1989*, 1; *Atkas 1991*, 2; *Wu*, Abstract, 2:50–3:21). A POSITA would have been well-aware of such interrupt functionality. As such, a POSITA would have understood that interrupts in speech recognition systems were capable of signaling between the basic, well-known stages—i.e., when one stage was ready to process data from another stage. Accordingly, a POSITA would have understood that *Vensko*'s interrupt is cable of signaling (and informing) the front end that the distance calculation is ready to receive another feature vector from the front end to process. *Vensko*, 6:22-33, 10:10-19.

261. A POSITA would have been motivated and found it obvious to further modify the *Jiang-Baumgartner-Brown* system to include *Vensko*'s interrupt functionality. For example, *Baumgartner* already contemplates the use of an interrupt in *Baumgartner*'s SLA. *Baumgartner*, [0179]-[0180]. A POSITA would have further understood that pipelined systems process system stages concurrently—for example the claimed “audio front end,” “calculating circuit,” and “search stage,” as explained in *Brown*. Specifically, *Brown* explains how the functionalities of each speech recognition stage are pipelined and synchronized:

Attention is first directed to FIG. 1 which is a block diagram of a speech recognition system 100 embodying the principles of the invention. The recognition system 100 includes a number of functional blocks including, a linear-predictive-coding-(LPC)-based feature extractor 105, a

vector quantizer 110, a mixture probability processor 115, a phone probability processor 120, a word probability processor 125, a grammar processor 130, and a traceback processor 135. The functionality of each block of the recognition system may be performed by a respective different processor, or the functionality of several or all blocks may be performed by the same processor. Furthermore, each stage can include multiple processing elements. The stages are pipelined and their operations are synchronized with the rate at which frames of the incoming speech signal are generated.

Brown, 3:32-48. Therefore, modifying *Jiang's* system such that the calculating circuit implemented using the second processor is ready to receive a next feature vector for processing would have been a routine and obvious modification for obtaining efficient processing.

262. A POSITA would have also been motivated to modify the *Jiang-Baumgartner-Brown* system to include *Vensko's* interrupt functionality because a POSITA would have understood that interrupts keep processors working efficiently. Specifically, triggering processor action on an interrupt signal is computationally more efficient than having the processor wait (what I would call “busy waiting”) and continually checking on the status of other processors. Processor efficiency is even more important when a system is processing data via multiple processors concurrently—such as in the case of the *Jiang-Baumgartner-Brown* system which

uses pipelining. Interrupts in such a system keep the processors working efficiently as new data and results become available, thus providing more effective and efficient computing and leading to better recognition results.

263. A POSITA would have had a reasonable expectation of success without undue experimentation in modifying the *Jiang-Baumgartner-Brown* system to utilize the interrupt functionality of *Vensko* at least because interrupt signals were well-known at the time and easy to implement. *See* Section, V.I (citing *Atkas 1989*, *Atkas 1991*, and *Wu*).

B. Dependent Claim 10

1. ***Claim 10: “The speech recognition circuit of claim 1, wherein the accelerator signals to the search stage when the distances for a new frame are available in a result memory.”***

264. As I explained above in Section IX.A.7 regarding Claim 1(f), data is pipelined from the calculating circuit (which computes distances) to the search stage. A POSITA would have understood that in such a pipelined system, the search stage uses the calculated distances output from the calculating circuit. As also explained in Section XI.A.1 regarding Claim 9, *Vensko*’s interrupt functionality is from the template processor (i.e., the processor for calculating distances or a similarity indication) to the master processor (i.e., the processor performing the functionality of the front end—which in the *Jiang-Baumgartner-Brown* system is the same processor which performs the search stage functionality). Further, *Jiang* teaches tree

search engine 74 accesses information stored in memory 72. *Jiang*, 7:31-34. A POSTIA would have understood that in a pipelined system, the template processor would provide an interrupt signal to the search stage processors indicating results are ready. Accordingly, a POSITA would have understood that the *Jiang-Baumgartner-Brown-Vensko* system teaches or at least renders obvious Claim 10.

XII. OPINIONS REGARDING GROUND 4: CLAIMS 1, 5, 7, 12, AND 14-16 ARE OBVIOUS OVER *JIANG*, *BAUMGARNTER*, *BROWN*, AND *SMYTH*

A. Independent Claim 1

1. Claim 1(Pre): “A speech recognition circuit, comprising:”

265. See Section IX.A.1 regarding Claim 1(Pre).

2. Claim 1(a): “an audio front end for calculating a feature vector from an audio signal,”

266. See Section IX.A.2 regarding Claim 1(a).

3. Claim 1(b) “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”

267. See Section IX.A.3 regarding Claim 1(b).

4. Claim 1(c): “a calculating circuit for calculating distances indicating the similarity between a feature vector and a plurality of predetermined acoustic states of an acoustic model; and”

268. As I explained above in Section IX.A.4 regarding Claim 1(c), *Jiang* alone or in combination with *Baumgartner* teaches or at least renders obvious Claim 1(c).

269. In my opinion, *Smyth* also teaches Claim 1(c). Specifically, *Smyth* teaches classifier (also referred to as classification or classifying) processor 341 which as explained below, calculates state probabilities including the similarity that a feature vector corresponds to each acoustic state in state memory 342. *Smyth*, 5:55–6:4. Below I explain how *Smyth* teaches both claimed “calculating circuit” as well as the functionality for the claimed “calculating distances.”

a) “a calculating circuit”

270. A POSITA would have understood that *Smyth*’s classifier processor 341 (also referred to as “classification processor 34” and “classifying processor 341”) teaches the claimed “calculating circuit for calculating distances.” As I explain below, *Smyth*’s classifier processor teaches the claimed “calculating circuit” for which the “calculating distances” functionality is performed on. *Smyth*, 5:55–6:4.

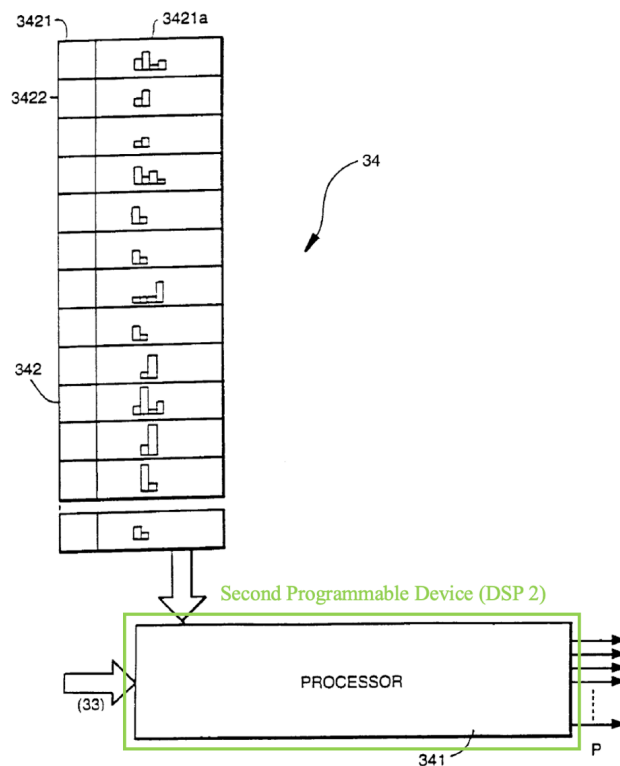
271. For the same reasons I explained above, a POSITA would have understood that a processor comprises a circuit and therefore teaches a circuit, or at least renders such obvious. See Section IX.A.4.b.2 (explaining how *Jiang*’s processor similarly teaches or renders obvious a circuit). *Smyth*’s teaches “classification processor 34” and “classifying processor 341.” I note that both refer to classifier processor 341 and that both terms would have been interchangeable with no difference in meaning. Specifically, a POSITA would have understood that

classifier processor 341 would have performed the calculating functions of classifier 34. *Smyth*, 5:12-13, 5:55-57, 5:63, 6:1-8.

272. Additionally, as I explained above in Section IX.A.4.b.1, the '277 Patent does not appear to ascribe any structural features or requirements to the claimed "circuit," and a POSITA would have understood a "circuit" is met by a processor.

273. *Smyth*'s Figure 3 labels classifier processor 341 as a processor:

Fig. 3



Smyth, Fig. 3 (annotated).

b) “a plurality of predetermined acoustic states”

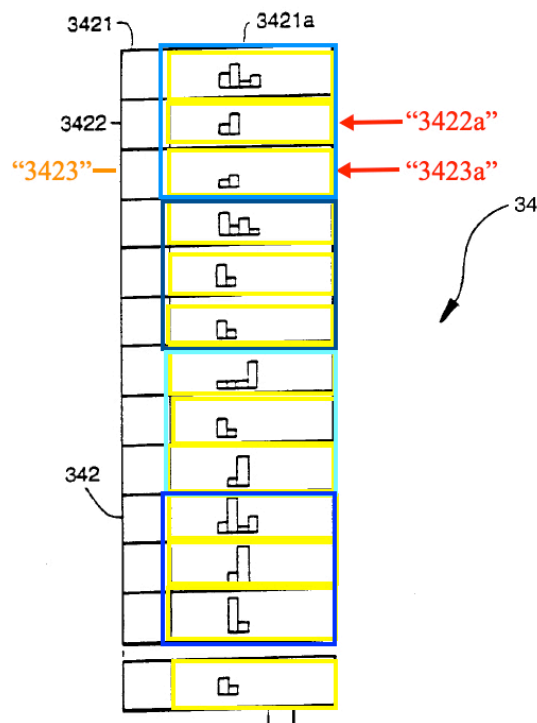
274. In my opinion, *Smyth* teaches the claimed “plurality of predetermined acoustic states,” namely speech states stored in state fields of state memory.

275. I note that *Smyth* teaches classifier 34 comprises classifier processor 341 and state memory 342. *Smyth*, 5:55-57, Fig. 3. “The state memory 342 comprises a state field 3421, 3422, ...” *Smyth*, 5:58-59. Each state field stores a speech state. *Smyth*, 5:58-59 (“a state field 3421, 3422 ..., **for each of the plurality of speech states.**”). I note that *Smyth*’s original Fig. 3 only expressly identifies two of *Smyth*’s state fields by reference number—3421 and 3422. *Smyth*, Fig. 3. *Smyth*’s figure 3 also only identifies one speech state by reference number—3421a. *Smyth*, Fig. 3.

276. A POSITA would have understood that in *Smyth*’s Figure 3, the cells following 3421, 3422 are also state fields, and that each state field stores a speech state similar to speech state 3421a. *Smyth*, 5:58-59 (“The state memory 342 comprises a state field 3421, 3422 . . . , for each of the plurality of speech states.”). The corresponding disclosures in *Smyth* further support that *Smyth*’s Figure 3 contained additional state fields and speech states. For example, *Smyth*’s use of ellipses indicate similar state fields with respective speech states, specifically indicating to a POSITA that state fields 3421 and 3422 were not exhaustive, but rather that *Smyth* only chose to give a few express examples while leaving open the possibility of more state fields. Annotated Fig. 3 below depicts state memory 342,

state fields 3421-3422 (with state field 3423 added and annotated orange), speech state 3421a (with speech states 3422a-3423a added and annotated red), and groups of three speech states corresponding to an allophone (annotations added in shades of blue):

Fig. 3



Smyth, Fig. 3 (annotated). I note that *Smyth* does not identify element 3421a in its written description (although 3421a is identified in Fig. 3). However, a POSITA would have understood that *Smyth* teaches a “plurality of speech states” with each state field having a speech state. *Smyth*, 5:58-62. For example, in the annotated Figure 3 above, a POSITA would have understood that the right column represents speech states corresponding to state fields. *Smyth*’s Figure 3 only shows one

example of a corresponding pair—3421 and 3421a. A POSITA would have understood that this trend continues in *Smyth*'s Figure 3 as seen with the exemplary reference numbers added in orange and red to the annotated Figure 3. Accordingly, 3421a, 3422a, and 3423a are speech states of state fields 3421, 3422, and 3423, respectively. A POSITA would have understood that these three pairs of speech states and corresponding state fields are merely exemplary and not exhaustive.). *Smyth*, 5:58-62. Therefore, in my opinion *Smyth* teaches a plurality of states.

277. A POSITA would have understood that *Smyth*'s speech states are “acoustic states,” claimed. As I explained above, it was well-known that allophones represented speech segments. *See* Section V.D. (quoting *Schmandt*, 59-60). *Smyth* expressly teaches “each allophone to be recognised by the recognition processor comprises 3 states, and accordingly 3 state fields are provided in the state memory 342 for each allophone.” *Smyth*, 5:60-62 (“each allophone to be recognised by the recognition processor comprises 3 states, and accordingly 3 state fields are provided in the state memory 342 for each allophone.”).

278. An allophone is well-known as an “acoustic” subword unit, namely alternative pronunciations of phonemes. In my book, *Schmandt*, I explained that an “allophone is one of a number of different ways of pronouncing the same phoneme.” *Schmandt*, 24. *Schmandt* also teaches that “[f]or most practical purposes, phone and phoneme may be considered to be synonyms.” Because allophones correspond to

phonemes, and phonemes are synonymous with phones, a POSITA would have understood that *Smyth*'s speech states meet the claimed "acoustic states" because such speech states correspond to a "phone or group of phones."

279. A POSITA would have also understood that *Smyth*'s speech states are predetermined. As I explained above in Section IX.A.4.b.2, it was well-known that speech states were predetermined because they existed in system memory and were created prior to distance computations (or the three-well known speech recognition steps), for example during a training phase. *Smyth*'s speech states are also "predetermined" for the same reasons—because the states exist and are stored in system memory prior to the distance calculations. *Smyth* expressly teaches storing a template or model of a word or phrase containing a reference feature matrix. *Smyth*, 1:34-39. It was also well-known that speech states were predetermined in speech recognition. *Ravishankar*, 19-20 (explaining HMM models are trained and stored in system memory in *Ravishankar*'s section titled "Acoustic Models"), 32 (referencing the state-time matrix of HMM states in *Ravishankar*'s Figure 2.1), 27-28. *Smyth*'s state fields are specifically stored in state memory. *Smyth*, 5:58-67. In sum, *Smyth* teaches a model or template of speech words are created and such model contains speech states. Therefore, a POSITA would have understood that *Smyth* teaches its states are "predetermined."

c) “[a plurality of predetermined acoustic states] of an acoustic model”

280. In my opinion, *Smyth* teaches or renders obvious its predetermined acoustic states are “of an acoustic model,” as claimed. Specifically, teaches its acoustic states are created “as a template or model”: :

A stored representation of the word or phrase, known as a template or model, comprises a **reference feature matrix** of that word as previously derived from, in the case of speaker independent recognition, multiple speakers. The **input feature vector is matched with the model** and a measure of similarity between the two is produced.

Smyth, 1:34-39, 5:7-9. Such templates or models are used to determine the distance (or probability or similarity) that a given feature vector is one of the speech states in the model or template. *Smyth*, 1:34-39, 5:7-9. *Smyth* further teaches “classifier 34 receiv[es] the succession of feature vectors and operat[es] on each with a **plurality of model states**, to generate recognitions results.” *Smyth*, 5:6-9.

281. A POSITA would have understood that it was well-known to train the speech recognition system to generate models or templates of spoken input, as otherwise taught by *Smyth*, 1:34-39. For example, *Ney* teaches estimating probabilities of acoustic vectors “during the training phase of the recognition system.” *Ney*, 4. *Jiang* teaches “[i]n order to train the system shown in FIG. 2, training words are spoken into the system 60 by the user. The training words

comprise a set of training data which is converted to digital samples by A/D converter 64 and then to codewords by feature extraction module 66.” *Jiang*, 11:25-29. *Ravishankar* discusses training in order to build word models by creating HMMs corresponding to phonemes. *Ravishankar*, 19-20, 27-28 (further discussing training acoustic models in subword units, such as phonemes)). A POSITA would have understood *Smyth*’s “plurality of model states” is an *acoustic model*. *Smyth*’s speech states in the model are used to determine distances—where such distances indicate the likelihood that a given frame of spoken speech (in the form of a feature vector) corresponds to one of *Smyth*’s speech states. *Smyth*, 5:58-62. For example, my book explains a “**phoneme** is a unit of speech, the set of which defines all the sounds from which words can be constructed in a particular language.” *Schmandt*, 23. My book also explains an “**allophone** is one of a number of different ways of pronouncing the same phoneme” (*Id.*, 24) and further teaches “[e]ach language employs a characteristic set of phonemes or allophones to convey words.” *Id.*, 33. In another example, *Peters* teaches a “reference speaker independent model set database comprises 800 allophone speech models[.]” *Peters*, 9:39-40. *Peters* teaches that such speech models were “available as off-the-shelf components and are well known in the art[.]” *Peters*, 9:45-48.

282. *Smyth*’s acoustic model is used to calculate probabilities for each state in state memory 342. *Smyth*, 5:63–6:4. This probability represents the likelihood that

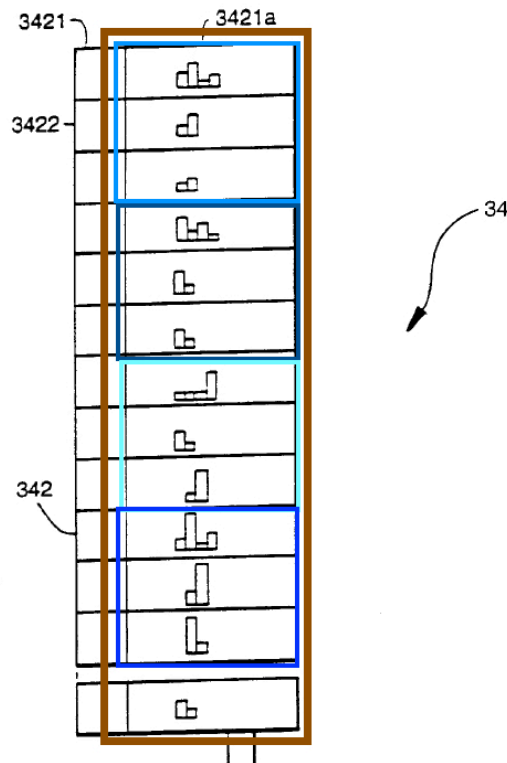
a feature vector corresponds to a particular speech state. *Smyth* also teaches its recognition system “operat[es] on each [feature vector] with a plurality of model states,” thus further indicating the plurality of model states are an acoustic model used to determine probabilities. *Smyth*, 5:7-9. In other words, *Smyth* teaches a “plurality of model states” used to determine the likelihood that a feature vector corresponds to an allophone. *Smyth*, 5:7-9.

283. *Smyth* teaches words are determined by stringing together identified allophones. *Smyth*, 6:15-19. Regarding *Smyth*’s sequencer 35, I note the sequencer determines a word or phrase “consisting of a string of allophones.” *Smyth*, 6:9-19. *Smyth* teaches “[t]he states of the sequence are a plurality of groups of three states each relating to a single allophone.” *Smyth*, 6:24-26. *Smyth* then teaches that the parsing processor 351 reads for each frame the state probability and calculates the most likely state path. *Smyth*, 6:36-42. Therefore, a **sequence** of states corresponds to a word, phrase, or other speech sequence. *Smyth*, 6:52-55. In my opinion, a POSITA would have understood from these teachings and other collective teachings in *Smyth* that a word is determined by stringing together strings of allophones. Because words are made up of phonemes, and allophones are just different ways of saying a phoneme, *Smyth* teaches word identification by stringing together allophones. *Smyth* describes the word or phrase sequence consisting of a string of

allophones (*Smyth*, 6:17-19), which is also frequently described as *concatenation* of phonemes (or allophones). *Ravishankar*, 41.

284. *Smyth*'s "string of allophones" represents a stored word or phrase in the template or model. *Smyth*, 1:34-39. When the "string of allophones" are put together, a word can be determined. *Smyth*, 6:15-19. *Smyth*'s allophones (annotated shades of blue) are depicted in annotated Figure 3 below as a collection of acoustic states (annotated brown):

Fig. 3



Smyth, Fig. 3 (excerpt, annotated). Three speech states are provided for each allophone. Therefore, *Smyth*'s groups of speech states teach or render obvious the claimed "acoustic model."

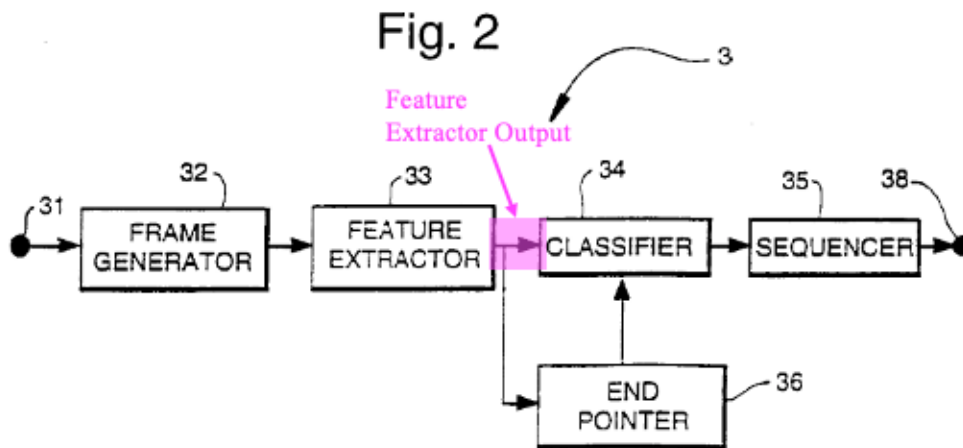
d) "for calculating distances ..."

285. *Smyth* teaches calculating distances for indicating the similarity, as claimed. *Smyth* teaches:

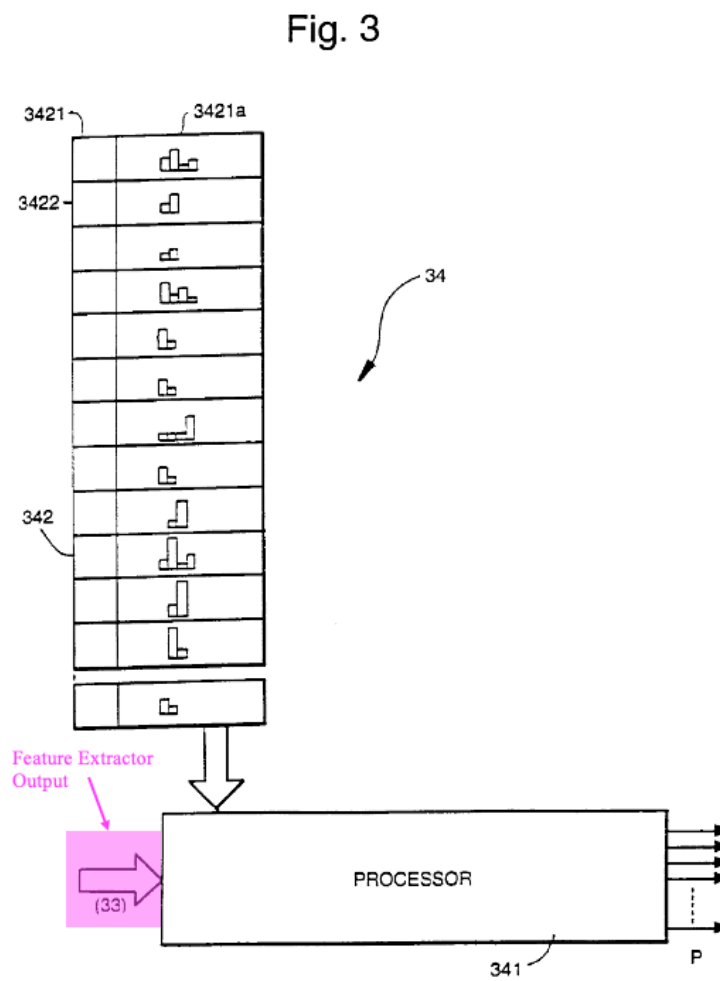
The **classification processor 34[1]** is arranged to **read each state field** within the memory 342 in turn, and **calculate for each**, using the current input **feature coefficient set**, the *probability* that the **input feature set or vector corresponds to the corresponding state**. [¶]
Accordingly, the **output** of the classification processor is **a plurality of state probabilities P, one for each state** in the state memory 342, **indicating the likelihood that the input feature vector corresponds to each state**.

Smyth, 5:63–6:4. A POSITA would have understood that *Smyth*'s classifier 34 compares feature vectors to states in *Smyth*'s predetermined acoustic model. *Smyth*, 5:7-9 ("a classifier 34 receiving the succession of feature vectors and operating on each with a plurality of model states."), 5:55-57 (disclosing "the classifier 34 comprises a classifying processor 341 and a state memory 342").

286. Additionally, a POSITA would have understood that *Smyth*'s Figures illustrate that classifier 34 receives such feature vectors. Specifically, *Smyth*'s Figs. 2-3 depict the output of feature extractor 33 as the input to classifier processor 341.



Smyth, Fig. 2 (annotated).



Smyth, Fig. 3 (annotated).

287. A POSITA would have understood *Smyth* teaches or renders obvious the claimed “calculating distances.” I note that the ’277 Patent describes a distance as “an indication of the probability or likelihood that a feature vector corresponds to a particular state.” ’277 *Patent*, 4:5-15. I additionally note the ’277 Patent describes “[t]he distance calculator [(i.e., the claimed calculating circuit)]...computes a probability or likelihood that a feature vector corresponds to a particular state.” ’277 *Patent*, 12:38-40. *Smyth* accordingly teaches a “distance” because *Smyth*’s calculated probabilities P indicates the likelihood (or similarity) between the feature vector and the stored acoustic state. *Smyth*, 5:63–6:4. A POSITA would have understood that *Smyth*’s disclosures parallel the ’277 Patent’s description regarding distance calculations.

288. *Smyth* also teaches calculating “distances” (plural) because such distance computation is done for each state, and as I explained above *Smyth*’s acoustic model contains a plurality of states. *Smyth*, 1:34-37, 5:9, 5:59, 6:1-4 (teaching calculating a plurality of probabilities “one for each state in the state memory”). *Smyth*’s classification processor 341 (i.e., the claimed “calculating circuit”) calculates distances because *Smyth*’s processor 341 receives the feature vector from feature extractor 33 and calculates the probability indicating the likelihood (or similarity) the feature vector corresponds to a given predetermined acoustic state of the acoustic model. *Smyth*, 5:7-9 (discussing classifier receiving

feature vectors and generating recognitions results by operating on a plurality of model states).

e) Motivation to Modify *Jiang-Baumgartner-Brown* with *Smyth*

289. A POSITA would have found it obvious and been motivated to modify the *Jiang-Baumgartner-Brown* system to include *Smyth*'s "calculating circuit," as claimed.

290. I note that although *Jiang* expressly teaches calculating a probability or likelihood that an observed feature vector corresponds to a an HMM state, *Smyth* provides additional disclosures regarding how the distance calculation is performed, for example by expressly teaching the computation of probabilities "indicating the likelihood that the input feature vector corresponds to each state." *Smyth*, 6:1-4.

291. The *Jiang-Baumgartner-Brown* system utilizes *Baumgartner*'s SLA as the claimed "second processor" where the claimed "calculating circuit" is implemented on *Baumgartner*'s SLA, per Claim 1(e).

292. A POSITA would have been motivated to incorporate the functionality of *Smyth*'s classifier processor 341 into the *Jiang-Baumgartner-Brown* system by specifically implementing *Smyth*'s classifier processor 341 on *Baumgartner*'s SLA, where *Smyth*'s classifier processor calculates the probability that the input feature vectors correspond to given states in the acoustic model. *Smyth*, 5:63-6:4.

293. A POSITA would have been motivated to make such modification for multiple reasons, including using a known technique to improve a similar device and performing a simple substitution of one known element for another to obtain predictable results, particularly given the well-known nature of distance calculation.

294. I described the well-known nature of distance calculations for determining the similarity between a feature vector and states of acoustic models above in Section V.E. Specific examples of references evidencing the well-known nature of distance calculations include, although certainly are not limited to, *Rabiner, Nadas, Holmes, Dudemaine, Ittycheriah, Kosaka, Yang, Maes, Jelinek*. See Section V.E. A POSITA would have understood that determining the similarity (or likelihood or probability) between input feature vectors and pre-stored states of acoustic models was one of the three main steps in speech recognition.

295. A POSITA would have been motivated to incorporate *Smyth's* calculating circuit on the *Jiang-Baumgartner-Brown* system's SLA because such would have been a known technique to improve a similar device. As I explained in the preceding paragraph, it was well-known to perform the claimed "distance" computation. Additionally, *Jiang, Baumgartner, Brown*, and *Smyth* all teach speech recognition systems for performing the different stages of speech recognition. See Sections VIII.A-C, F. *Jiang* provides the background architecture for performing probability computations by using HMMs to determining "most likely phoneme[s]"

and therefore identify words. *Jiang*, 7:36-42. Therefore, a POSITA would have been able to use *Smyth's* well-known technique—*Smyth's* calculated probabilities *P* indicates the likelihood (or similarity) between the feature vector and the stored acoustic state (*Smyth*, 5:63–6:4)—to improve the similar *Jiang-Baumgartner-Brown* system.

296. A POSITA would have also been motivated to incorporate *Smyth's* calculating circuit on the *Jiang-Baumgartner-Brown* system's SLA because such would have been a simple substitution of one known element for another, providing predictable results. Determining a similarity (or likelihood or probability) was so well-known, a POSITA would have been motivated to replace *Jiang's* probability scores for each phoneme branch (where the “score represents the likelihood that the particular phoneme encoded by the codeword corresponds to the phoneme for the branch under consideration”), with *Smyth's* probability calculation that an input feature vector corresponds to a given state. *Jiang*, 8:29-33; *Smyth*, 5:63-67. As explained in this Section and above, such distance or probability computation was well-known. Section V.E; *see also*, *Ravishankar*, 21, 28-29. Therefore, such a substitution would have provided the predicable result of “distance” computations to allow the *Jiang-Baumgartner-Brown* system to perform a “search” and identify spoken words. A POSITA would have been able to readily implement *Smyth's* calculating circuit into the *Jiang-Baumgartner-Brown* system.

297. A POSITA would have had a reasonable expectation of success without undue experimentation at least because determining the similarity (or distance or probability) of feature vectors to states was so well-known, and further because *Jiang's* hardware architecture already provides for such. Because the similarity (or distance or probability) determination was so well-known and commonly used, a POSITA would have had a reasonable expectation of success in making the proposed modification. *See Ravishankar*, 21, 28-29 (further evidencing the well-known nature of the similarity/distance/probability determination). Additionally, a POSITA would have had a reasonable expectation of success implementing *Smyth's* "calculating circuit," as claimed, on *Baumgartner's* SLA, as *Baumgartner* expressly teaches the "heart" of the SLA is a "circuit." *Baumgartner*, [0043].

5. ***Claim 1(d): "a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words;"***

298. *See* Section IX.A.5 regarding Claim 1(d).

6. ***Claim 1(e): "wherein said audio front end and said search stage are implemented using a first processor, and said calculating circuit is implemented using a second processor, and"***

299. *See* Section IX.A.6 regarding Claim 1(e).

7. ***Claim 1(f): "wherein data is pipelined from the front end to the calculating circuit to the search stage."***

300. *See* Section IX.A.7 regarding Claim 1(f).

B. Dependent Claim 5

1. *Claim 5: “A speech recognition circuit as claimed in claim 1, wherein the said calculating circuit is configured to autonomously calculate distances for every acoustic state defined by the acoustic model.”*

301. See Section IX.B regarding Claim 5.

C. Dependent Claim 7

1. *Claim 7: “The speech recognition circuit of claim 1, wherein the feature vector comprises a plurality of spectral components of an audio signal for a predetermined time frame.”*

302. See Section IX.C regarding Claim 7.

D. Dependent Claim 12

1. *Claim 12: “The speech recognition circuit of claim 1, wherein the audio front end is configured to input a digital audio signal.”*

303. See Section IX.D regarding Claim 12.

E. Independent Claim 14

1. *Claim 14(Pre): “a speech recognition circuit, comprising:”*

304. See Section IX.E.1.a regarding Claim 14(Pre).

2. *Claim 14(a): “an audio front end for calculating a feature vector from an audio signal,”*

305. See Section IX.E.2 regarding Claim 14(a).

3. *Claim 14(b) “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”*

306. See Section IX.E.3 regarding Claim 14(b).

4. ***Claim 14(c): “calculating means for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and”***

307. See Section IX.E.4 regarding Claim 14(c).

308. Additionally, see Section XII.A.4 regarding Claim 1(c) as further taught by *Smyth*.

309. Additionally, *Jiang*, *Baumgartner*, and *Smyth* each teach a “calculating means” comprising a processor, as properly constructed. See Section III.D.1 (mapping the construed structure (processor) and function).

5. ***Claim 14(d): “a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words;”***

310. See Section IX.E.5 regarding Claim 14(d).

6. ***Claim 14(e): “wherein said audio front end, said calculating means, and said search stage are connected to each other to enable pipelined data flow.”***

311. See Section IX.E.6 regarding Claim 14(e).

F. Independent Claim 15

1. ***Claim 15(Pre): “A speech recognition method, comprising:”***

312. See Section IX.F.1 regarding Claim 15(Pre).

2. ***Claim 15(a): “calculating a feature vector from an audio signal using an audio front end,”***

313. See Section IX.F.2 regarding Claim 15(a).

3. ***Claim 15(b): “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”***

314. See Section IX.F.3 regarding Claim 15(b).

4. ***Claim 15(c): “calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model using a calculating circuit; and”***

315. See Section IX.F.4 regarding Claim 15(c).

316. Additionally, see Section XII.A.4 regarding Claim 1(c) as further taught by Smyth.

5. ***Claim 15(d): “using a search stage to identify words within a lexical tree using said calculated distances, the lexical tree comprising a model of words;”***

317. See Section IX.F.5 regarding Claim 15(d).

6. ***Claim 15(e): “wherein data is pipelined from the front end, to the calculating circuit, and to the search stage.”***

318. See Section IX.F.6 regarding Claim 15(e).

G. Independent Claim 16

1. ***Claim 16(Pre): “A non-transitory storage medium storing processor implementable code for controlling at least one processor to implement a speech recognition method, the code comprising:”***

319. See Section IX.G.1 regarding Claim 16(Pre).

2. ***Claim 16(a): “code for controlling the processor to calculate a feature vector from an audio signal,”***

320. See Section IX.G.2 regarding Claim 16(a).

3. *Claim 16(b): “wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;”*

321. See Section IX.G.3 regarding Claim 16(b).

4. *Claim 16(c): “code for controlling the processor to calculate a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and”*

322. See Section IX.G.4 regarding Claim 16(c).

323. Additionally, see Section XII.A.4 regarding Claim 1(c) as further taught by Smyth.

5. *Claim 16(d): “code for controlling the processor to identify words within a lexical tree using said calculated distances, the lexical tree comprising a model of words,”*

324. See Section IX.G.5 regarding Claim 16(d).

6. *Claim 16(e): “wherein data is pipelined by the processor pursuant to the code from the feature calculation, to the distance calculation, and to the word identification.”*

325. See Section IX.G.6 regarding Claim 16(e).

XIII. OPINIONS REGARDING GROUND 5: CLAIM 4 IS OBVIOUS OVER JIANG, BAUMGARTNER, BROWN, SMYTH, AND KAZEROONIAN

A. Dependent Claim 4

1. *Claim 4: “A speech recognition circuit as claimed in claim 1, wherein the first processor supports multi-threaded operation, and runs the search stage and front ends as separate threads”*

326. See Section X.A.1 regarding Claim 4.

327. Additionally, as explained above in Section XII.A.4 regarding Claim 1(c), *Smyth* teaches, or renders obvious, the “calculating circuit for calculating distances ...” limitation.

XIV. OPINIONS REGARDING GROUND 6: CLAIMS 9-10 ARE OBVIOUS OVER *JIANG, BAUMGARTNER, BROWN, SMYTH, AND VENSKO*

A. Dependent Claim 9

1. *Claim 9: “The speech recognition circuit of claim 1, wherein the speech accelerator has an interrupt signal to inform the front end that the accelerator is ready to receive a next feature vector from the front end.”*

328. See Section XI.A.1 regarding Claim 9.

329. Additionally, as explained above in Section XII.A.4 regarding Claim 1(c), *Smyth* teaches, or renders obvious, the “calculating circuit for calculating distances ...” limitation.

B. Dependent Claim 10

1. *Claim 10: “The speech recognition circuit of claim 1, wherein the accelerator signals to the search stage when the distances for a new frame are available in a result memory.”*

330. See Section XI.B.1 regarding Claim 10.

331. Additionally, as explained above in Section XII.A.4 regarding Claim 1(c), *Smyth* teaches, or renders obvious, the “calculating circuit for calculating distances ...” limitation.

Declaration of Christopher Schmandt
Patent No. 7,979,277

XV. CONCLUSION

332. For the aforementioned reasons, in my opinion, to a POSITA, the claims of the '277 Patent are rendered obvious by the prior art.

333. I hereby declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct, and that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true. I understand that willful false statements are punishable by fine or imprisonment or both. See 18 U.S.C. § 1001.

Date: November 15, 2022

Respectfully submitted:

By: 

Chris Schmandt

Christopher Schmandt

4 Longfellow
Winchester, MA 01890
+1-617-230-4257

Education

M.I.T., Master of Science, Visual Studies (Computer Graphics), 1980

M.I.T., Bachelor of Science, Computer Science, 1978

Professional Experience -- MIT

Media Laboratory, Principal Research Scientist, 1985-2018 (retired)
Director, Living Mobile Research Group (formerly Speech + Mobility)

Architecture Machine Group, Research Associate, 1980-1984

Architecture Machine Group, Research Assistant, 1979-1980

Architecture Machine Group, Graphics Programmer, 1977-1979

Departmental Undergraduate Research Opportunities Program Coordinator, 1984-2018

Laboratory Intellectual Property Committee 2001-2016, chair 2002-2009

Departmental Committee on Graduate Studies, 1996-2001, 2007-2018

Sponsored Research Activities

Alerting and Mobile Messaging, Digital Life Consortium, MIT Media Lab, 1997-2018

Acoustical Cues to Discourse Structure, National Science Foundation, Principal Investigator, 1995-1998

Parsing Radio, News in the Future Consortium, MIT Media Lab, 1993-1999

Desktop Audio, SUN Microsystems, Inc., Principal Investigator, 1989-1996

Voice Interaction in Hand Held Computers, Apple Computer, Principal Investigator, 1991-1993

Voice Interfaces for Network Services, AT&T, Principal Investigator, 1989-1991

Back Seat Driver, NEC, Principal Investigator, 1988-1991

Acoustic and Visual Cues for Speech Recognition, DARPA, co-Principal Investigator, 1986-1988

Personal Computers and Telephony, NTT Public Corporation, Principal Investigator, 1984-1989

Home Telecomputing, Atari, Inc., Principal Investigator, 1983

Testifying legal engagements within the last five years, representing party in italics

- *Facebook v. Windy City*, 2017
- *Facebook v. EVERYMD.COM LLC*, 2017
- *Comcast v. Promptu*, 2017
- *Express Mobile, Inc v. Big Commerce*, 2017
- *Facebook v. Hypermedia*, 2017
- *Microsoft v. Uniloc 2017 LLC*, 2019
- *Microsoft v. SpeakWare*, 2019
- *Bumble Trading LLC v. Match Group LLC*, 2019
- *Tile Inc. v. Cellwitch Inc*, 2019
- *Blackberry Ltd v. Facebook, Inc* 2020
- *EROAD Ltd v. PerDiem Co*, 2020
- *Snap Inc. v. SRK Technology LLC*, 2020
- *Shopify Inc Ltd v. Express Mobile Inc*, 2020
- *Express Mobile Inc, v Wix Ltd* 2020
- *Motorola Mobility LLC v. Ironworks patents, LLC*. 2021
- *Express Mobile Inc, v GoDaddy Inc* 2021
- *Bumble Trading LLC v. Kinectus, LLC*. 2021
- *Tile, Inc v. Linquet Technologies, Inc*. 2021
- *Facebook, Inc. v. Palo Alto Research Center, Inc*. 2021
- *Unified Patents, LLC v. Gesture Technology Partners, LLC*. 2021
- *Quantum Metric, Inc v. Content Square Isreal, Ltd.* 2021
- *Meta Platforms Inc. v. AlmondNet, Inc* 2022

Professional Experience -- Intellectual Property

Co-inventor on U.S. Patents

- 5,177,685 for "Automobile navigation system using real time spoken driving instructions"
- 6,728,348 for "System for storing voice recognizable identifiers using a limited input device such as a telephone key pad"
- 6,937,986 for "Automatic dynamic speech recognition vocabulary based on external sources of information"
- 7,098,776 for "Methods and apparatus for vibrotactile communication"
- 7,392,280 for "Method for summarization of threads in electronic mail"
- 7,738,637 "Interactive voice message retrieval"
- 7,443,283 for "Methods and apparatus for connecting an intimate group by exchanging awareness cues and text, voice instant messages, and two-way voice communications"
- 7,865,560 for "System for summarization of threads in electronic mail"
- 8,121,653 for "Methods and apparatus for autonomously managing communications using an intelligent intermediary"
- 8,135,128 for "Animatronic creatures that act as intermediaries between human users and a telephone system"

Co-inventor on U.S. Patent Applications

- 20020076009 "International dialing using spoken commands"
- 20030023688 "Voice-based message sorting and retrieval method"
- 20030081738 "Method and apparatus for improving access to numerical information in voice messages"
- 20030144846 "Method and system for modifying the behavior of an application based upon the application's grammar"

- 20030158903 "Calendar bar interface for electronic mail interaction"

Publications

Most publications can be found at <http://www.media.mit.edu/speech/publications> or <http://living.media.mit.edu/publications>

Field Study of a Tactile Sound Awareness Device for Deaf and Hard of Hearing Users ISWC 2020. (with D. Jain, B. Chiu, S. Goodman, L. Findlater, J. Froehlich)

SkinMorph: Texture-Tunable On-Skin Interface Through Thin, Programmable Gel ISWC 2018. (with Cindy Kao, M. Banforth, D. Kim)

Technical Interventions to Detect, Communicate, and Deter Sexual Assault. ISWC 2017. (with Manisha Mohan)

Exploring Interactions and Perceptions of Kinetic Wearables. DIS 2017. (with Cindy Hsin-Liu Kao, D. Ajilo, O. Anilionyte, A. Dementyev, I. Choi and S. Follmer)

Leveraging User-made Predictions to Help Understand Personal Behavior. MobileHCI 2017. (with Miriam Greis, Tilman Dangler and Albrecht Schmidt)

Rovables: On-Body Robots as Mobile Wearables. UIST 2016. (with Cindy Hsin-Liu Kao, A. Dementyev, I. Choi, D. Ajilo, M. Xu, and S. Follmer)

DuoSkin: Rapidly Prototyping On-Skin User Interfaces Using Skin-Friendly Materials. ISWC 2016. (with Cindy Hsin-Liu Kao, Christian Holz, Asta Roseway, and Andres Calvo)

Immersive Terrestrial Scuba Diving Using Virtual Reality (with Dhruv Jain, Misha Sra, Jingru Go, Rodrigo Margues, Raymond Wu and Justin Chiu) Proceedings, UIST 2016

Expanding social mobile games beyond the device screen (with Misha Sra) Journal of Personal and Ubiquitous Computing, 2015

NailO: Fingernails as an input surface (with Cindy Hsin-Liu Kao, Artem Dementyev, Joseph Paradiso) CHI 2015

Mugshots: A mug display for front and back stage social interaction in the workplace (with Cindy Hsin-Liu Kao) TEI (Tangible and Embedded Interfaces) 2015

Mime: compact, low power 3D gesture sensing for interaction with head mounted displays (with Andrea Colaco, Ahmend Kirmani, Hye Soo Yang, Nan-Wei Gong, and Vivek Goyal) Proceedings of UIST 2013.

Spotz: A location-based approach to self-awareness (with Misha Sra) Proceedings of Persuasive 2013.

Setting the stage for interaction: A tablet application to augment group discussion in a seminar class (with Drew Harry and Eric Gordon) Proceedings of CSCW 2012.

Indoor Location Sensing using Geo-Magnetism (with Jaewoo Chung, Matt Donahoe, Ig-Jae Kim, Pedram Razavai and Micaela Wiseman) Proceedings of International Conference on Mobile Systems, Applications, and Services (Mobisys) 2011.

My second bike: a TV-enabled social and interactive riding experience (with Jaewoo Chung, Kuang Xu, Andrea Colaco, and Victor Li) Proceedings of IEEE Communications and Networking Conference, Jan 2010.

Going my way?: User-aware route planner (with Jaewoo Chung) Proceedings CHI 2009.

Globetoddler: Designing for remote interaction between preschoolers and their traveling parents (with Paulina Modlitba) CHI 2008 Extended Abstracts

Are we there yet? - a temporally aware media player (with Matt Adcock and Jaewoo Chung), Australian User Interface Conference (AUIC) 2008

Physical embodiments for mobile communication agents (with Stefan Marti), UIST 2005

Giving the caller the finger: collaborative responsibility for cellphone interruptions (with Stefan Marti) Extended Abstracts, CHI 2005

Active Messenger: email filtering and delivery in a heterogeneous network (with Stefan Marti) Human-Computer Interaction Journal (HCI) Volume 20 (2005)

WatchMe: communication and awareness between members of a closely-knit group (with Natalia Marmasse) Proceedings of Ubicomp 2004

An audio-based personal memory aid (with S. Vemuri, W. Bender, S. Tellex and B. Lassey) Proceedings of Ubicomp 2004

Improving speech playback using time-compression and speech recognition (with Sunil Vemuri, Philip DeCamp, and Walter Bender) Proceedings of CHI 2004

Impromptu: managing networked audio applications for mobile users (with Kwan Lee, Jang Kim, and Mark Ackerman), Proceedings of MobiSys 2004

TalkBack: a conversational answering machine (with Vidya Lakshminpathy and Natalia Marmasse) Proceedings of UIST 2003

"ListenIn" to domestic environments from remote locations (with Gerardo Vallejo) Proceedings of the 2003 International Conference on Auditory Display (ICAD)

Safe & Sound: a wireless leash (with Natalia Marmasse) Extended Abstracts, Proceedings of CHI 2003

Mediated voice communication via mobile IP (with Jang Kim, Kwan Lee, Gerardo Vallejo, and Mark Ackerman), Proceedings of UIST 2002.

The Audio Notebook: Paper and pen interaction with structured speech (with Lisa Stifelman and Barry Arons), Proceedings of CHI 2001

Synthetic News Radio (with Keith Emnett) IBM Systems Journal, Vol. 39 Nos. 3-4, pp. 646-659, 2000.

Everywhere messaging (with Natalia Marmasse, Stefan Marti, Nitin Sawhney, and Sean Wheeler) IBM Systems Journal, Vol. 39 Nos. 3-4, pp. 660-677, 2000.

Location-aware information delivery with comMotion (with Natalia Marmasse), Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing, pp. 157-171, Springer, 2000.

Nomadic Radio: Scalable and contextual notification for wearable audio messaging (with Nitin Sawhney), Proceedings of CHI 1999.

Speaking and listening on the run: Design for wearable audio computing (with Nitin Sawhney), Proceedings of International Symposium on Wearable Computing, 1998.

Audio Hallway: A virtual acoustic environment for browsing, Proceedings of UIST 1998.

Dynamic Soundscape: Mapping time to space for audio browsing (with Minoru Kobayashi), Proceedings of CHI 1997.

CLUES: Dynamic personalized message filtering (with Matt Marx), Proceedings of CSCW 1996.

Using acoustic structure in a hand-held audio playback device (with Deb Roy), IBM Systems Journal, Vol 35, Nos. 3 and 4, 1996.

Mailcall: Message presentation and navigation in a nonvisual environment (with Matt Marx), Proceedings of CHI 1996

AudioStreamer: Exploiting simultaneity for listening (with Atty Mullins), short paper, CHI 1995.

Multimedia momadic services on today's hardware, IEEE Network, September/October 1994.

Putting people first: Specifying proper names in speech interfaces (with Matt Marx), proceedings of UIST 1994.

Chatter: A conversational learning speech interface (with E. Ly) AAAI Workshop on Intelligent Multi-Media Multi-Modal Systems, 1994.

Voice Communication with Computers: Conversational Systems. New York: Van Nostrand Reinhold. 1994.

Capturing, structuring, and representing ubiquitous audio (with D. Hindus and C. Horner), ACM Transactions on Information Systems, Vol. 11, No. 4, October 1993.

Speech Recognition Architectures for Multimedia Environments, (with E. Ly and B. Arons), Proceedings of the 1993 AVIOS Conference, September 1993.

Phoneshell: the Telephone as Computer Terminal, Proceedings of the ACM Multimedia Conference, August 1993.

Voicenotes: A Speech Interface for a Hand-Held Voice Notetaker (with L. Stifelman, B. Arons, and E. Hulteen), Proceedings of INTERCHI'93, April 1993.

From Desktop Audio to Mobile Access: Opportunities for Voice in Computing, book chapter in Advances in Human-Computer Interaction, Vol. 4, H.R. Hartson and D. Hix editors. 1992.

Ubiquitous Audio: Capturing Spontaneous Collaboration (with D. Hindus), Proceedings of CSCW'92, November 1992.

Integrating Audio and Telephony in a Distributed Workstation Environment (with S. Angebrannt, R. Hyde, D. Luong, and N. Siravara), Proceedings of the Summer 1991 USENIX Conference, June 1991.

Augmenting a Window System with Speech Input (with M. Ackerman and D. Hindus), Computer, IEEE Computer Society, Vol. 23, No. 8, August 1990.

Observations on Using Speech Input for Window Navigation (with D. Hindus, M. Ackerman, and S. Manandhar), Proceedings, Human-Computer Interaction, Interact '90, IFIP, August 1990.

Phonetool: Integrating Telephones and Workstations (with S. Casner), Proceedings, GLOBECOM '89, IEEE Communications Society, November 1989.

Desktop Audio (with B. Arons), UNIX Review, October 1989.

Synthetic Speech for Real Time Direction-Giving (with J. Davis), IEEE Transactions on Consumer Electronics, IEEE, September 1989.

The Back Seat Driver: Real Time Spoken Driving Instructions (with J. Davis), Proceedings, IEEE Vehicle Navigation and Information Systems Conference, IEEE, Toronto, Canada, September 1989.

An Audio and Telephone Server for Multi-media Workstations (with M. McKenna), Proceedings, Second IEEE Conference on Workstations, IEEE, Palo Alto, CA., 1988.

Employing Voice Back Channels to Facilitate Audio Document Retrieval, Proceedings, ACM Conference on Office Information Systems (COIS), Santa Clara, CA, 1988.

Conversational Telecommunications Environments, Proceedings, Second International Conference on Human-Computer Interaction, 1987.

Understanding Speech Without Recognizing Words, Proceedings, American Voice Input/Output Society Conference, AVIOS, 1987.

A Robust Parser and Dialog Generator for a Conversational Office System (with B. Arons and C. Simmons), Proceedings, American Voice Input/Output Society Conference, AVIOS, Palo Alto, CA, 1987.

Integrated Messages and Network Services for a Personal Workstation, IEEE Workshop on Telematics and Message Handling Systems, IEEE, 1986.

Voice Interaction in an Integrated Office and Telecommunications Environment, Proceedings, American Voice Input/Output Society Conference, AVIOS, San Francisco, CA, 1985.

Voice Communication with Computers, book chapter in *Advances in Human-Computer Interaction*, H. R. Hartson ed., 1985.

Speech Synthesis Gives Voiced Access to an Electronic Mail System, *Speech Technology*, Vol. 2, No. 3, Aug/Sept 1984.

A Conversational Telephone Messaging System (with B. Arons), *IEEE Transactions on Consumer Electronics*, IEEE, Vol CE-30, August 1984.

Phone Slave: A Graphical Telecommunications Interface (with B. Arons), Proceedings, Society for Information Display International Symposium, SID, San Francisco, CA, June 1984.

Input/Display Registration in a Stereoscopic Workstation, *Displays*, April 1984.

Remote Access to Voice and Text Messages, Proceedings, American Voice Input/Output Society Conference, AVIOS, Washington D.C., 1984.

Fuzzy Fonts: Analog Models Improve Digital Text Quality, Proceedings, National Computer Graphics Association Conference, Chicago, IL, 1983.

Greyscale Fonts Designed From Video Signal Analysis, Society of Applied Learning Technology, Houston, TX, 1983.

Spatial Input/Display Correspondence in a Stereoscopic Computer Graphic Work Station, Proceedings, ACM/SIGGRAPH, Detroit, MI, 1983.

A Programmable Virtual Vocabulary Speech Processing Peripheral (with W. Bender), Proceedings, American Voice Input/Output Society Conference on Voice Data Entry Systems Applications, AVIOS, 1983.

The Intelligent Voice Interactive Interface (with E. A. Hulteen), Proceedings, Human Factors in Computer Systems, National Bureau of Standards/ACM, Gaithersburg, MD, 1982.

Interactive Three-Dimensional Computer Space, Proceedings, SPIE Conference on Processing and Display of Three-Dimensional Data, SPIE, San Diego, CA, 1982, Vol. 367.

Speech Communications, a Systems' Approach, Proceedings, American Voice Input/Output Society Conference on Entry Systems Applications, 1982.

Voice Interaction: Putting Intelligence into the Interface, Proceedings, IEEE International Conference on Cybernetics and Society, IEEE, Seattle, WA, 1982.

The Intelligent Ear: A Graphical Interface to Digital Audio, Proceedings, IEEE International Conference on Cybernetics and Society, IEEE, Atlanta, GA, 1981.

Soft Typography, Information Processing 1980, IFIPS, S. Lavington ed., North-Holland Publishing Co., 1980.
