

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. IPR2023-00035  
U.S. Patent No. 10,062,377

**DECLARATION OF CHRISTOPHER SCHMANDT  
IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF  
U.S. PATENT NO. 10,062,377**

**TABLE OF CONTENTS**

|                     |                                                                       |                  |
|---------------------|-----------------------------------------------------------------------|------------------|
| <b><i>I.</i></b>    | <b><i>INTRODUCTION AND QUALIFICATIONS.....</i></b>                    | <b><i>9</i></b>  |
| <b><i>A.</i></b>    | <b><i>Educational Background and Professional Experience.....</i></b> | <b><i>9</i></b>  |
| <b><i>II.</i></b>   | <b><i>METHODOLOGY; MATERIALS CONSIDERED.....</i></b>                  | <b><i>12</i></b> |
| <b><i>III.</i></b>  | <b><i>OVERVIEW AND LEGAL STANDARDS.....</i></b>                       | <b><i>15</i></b> |
| <b><i>A.</i></b>    | <b><i>Person of Ordinary Skill in the Art.....</i></b>                | <b><i>15</i></b> |
| <b><i>B.</i></b>    | <b><i>Obviousness .....</i></b>                                       | <b><i>16</i></b> |
| <b><i>C.</i></b>    | <b><i>Analogous Art.....</i></b>                                      | <b><i>21</i></b> |
| <b><i>D.</i></b>    | <b><i>Claim Construction.....</i></b>                                 | <b><i>22</i></b> |
| <b><i>IV.</i></b>   | <b><i>LEVEL OF A PERSON OF ORDINARY SKILL.....</i></b>                | <b><i>22</i></b> |
| <b><i>V.</i></b>    | <b><i>OVERVIEW OF THE TECHNOLOGY.....</i></b>                         | <b><i>24</i></b> |
| <b><i>A.</i></b>    | <b><i>Speech Recognition .....</i></b>                                | <b><i>24</i></b> |
| <b><i>B.</i></b>    | <b><i>Feature Vectors .....</i></b>                                   | <b><i>32</i></b> |
| <b><i>C.</i></b>    | <b><i>Acoustic Models.....</i></b>                                    | <b><i>41</i></b> |
| <b><i>D.</i></b>    | <b><i>Hidden Markov Models.....</i></b>                               | <b><i>42</i></b> |
| <b><i>E.</i></b>    | <b><i>Distance Calculations.....</i></b>                              | <b><i>51</i></b> |
| <b><i>F.</i></b>    | <b><i>Gaussian Distribution and Probability .....</i></b>             | <b><i>54</i></b> |
| <b><i>G.</i></b>    | <b><i>Speech Recognition System Hardware.....</i></b>                 | <b><i>57</i></b> |
| <b><i>H.</i></b>    | <b><i>Pipelining .....</i></b>                                        | <b><i>65</i></b> |
| <b><i>I.</i></b>    | <b><i>Interrupts.....</i></b>                                         | <b><i>69</i></b> |
| <b><i>J.</i></b>    | <b><i>Prior Art Speech Recognition Systems .....</i></b>              | <b><i>71</i></b> |
| <b><i>VI.</i></b>   | <b><i>OVERVIEW OF THE '377 PATENT.....</i></b>                        | <b><i>72</i></b> |
| <b><i>VII.</i></b>  | <b><i>THE PROSECUTION HISTORY OF THE '377 PATENT.....</i></b>         | <b><i>73</i></b> |
| <b><i>A.</i></b>    | <b><i>Prosecution History of the '377 Patent.....</i></b>             | <b><i>73</i></b> |
| <b><i>VIII.</i></b> | <b><i>SUMMARY OF UNPATENTABILITY .....</i></b>                        | <b><i>74</i></b> |
| <b><i>IX.</i></b>   | <b><i>OVERVIEW OF THE PRIOR ART.....</i></b>                          | <b><i>75</i></b> |
| <b><i>A.</i></b>    | <b><i>Overview of Jiang.....</i></b>                                  | <b><i>75</i></b> |
| <b><i>B.</i></b>    | <b><i>Overview of Smyth.....</i></b>                                  | <b><i>76</i></b> |

|    |                                                                                                                                                                                                                  |           |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| C. | Overview of Nguyen .....                                                                                                                                                                                         | 77        |
| D. | Overview of Boike .....                                                                                                                                                                                          | 78        |
| E. | Overview of Baumgartner .....                                                                                                                                                                                    | 78        |
| X. | <b><i>OPINIONS REGARDING GROUND 1: CLAIM 1 IS OBVIOUS<br/>OVER JIANG IN VIEW OF SMYTH.....</i></b>                                                                                                               | <b>79</b> |
| A. | <b>Claim 1(a): “a first programmable device programmed to<br/>calculate a feature vector from a digital audio stream,” .....</b>                                                                                 | <b>79</b> |
| 1. | Jiang Alone Teaches Claim 1(a) .....                                                                                                                                                                             | 79        |
| 2. | Jiang in Combination with Smyth Teaches Claim 1(a) .....                                                                                                                                                         | 86        |
| 3. | Motivation to Modify Jiang to Include Smyth’s First<br>Programmable Device.....                                                                                                                                  | 90        |
| B. | <b>Claim 1(b): “wherein the feature vector comprises a<br/>plurality of extracted and/or derived quantities from said<br/>digital audio stream during a defined audio time frame;” .....</b>                     | <b>91</b> |
| 1. | ’377 Patent’s Discussion of “extracted” or “derived<br>quantities” .....                                                                                                                                         | 91        |
| 2. | Jiang’s Teachings.....                                                                                                                                                                                           | 92        |
| C. | <b>Claim 1(c): “a second programmable device programmed to<br/>calculate distances indicating the similarity between a<br/>feature vector and a plurality of acoustic states of an<br/>acoustic model” .....</b> | <b>95</b> |
| 1. | Overview of Jiang’s Teachings Regarding Utilizing<br>HMMs and Prefix Trees for Speech Recognition as<br>Applicable to Claims 1(c)-1(h) .....                                                                     | 95        |
| 2. | ’377 Patent’s Discussion of “distance” .....                                                                                                                                                                     | 101       |
| 3. | ’377 Patent’s Teachings of “a plurality of acoustic states<br>of an acoustic model” .....                                                                                                                        | 103       |
| 4. | Jiang’s Teachings.....                                                                                                                                                                                           | 103       |
| 5. | Smyth’s Teachings.....                                                                                                                                                                                           | 113       |
| 6. | Motivation to Modify Jiang to Include Smyth’s Second<br>Programmable Device.....                                                                                                                                 | 133       |
| 7. | Motivation to Modify Jiang to Include Smyth’s Second<br>Programmable Device Performing Distance Calculations ....                                                                                                | 143       |

|              |                                                                                                                                                                           |            |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>D.</b>    | <b>Claim 1(d): “wherein said feature vector is received by the second programmable device after it is calculated by the first programmable device; and”</b> .....         | <b>145</b> |
| 1.           | Jiang’s Teachings.....                                                                                                                                                    | 145        |
| 2.           | Smyth’s Teachings.....                                                                                                                                                    | 146        |
| <b>E.</b>    | <b>Claim 1(e): “a third programmable device programmed to identify spoken words in said digital audio stream using Hidden Markov Models and/or Neural Networks”</b> ..... | <b>146</b> |
| 1.           | Jiang’s Teachings.....                                                                                                                                                    | 147        |
| 2.           | Smyth’s Teachings.....                                                                                                                                                    | 154        |
| <b>F.</b>    | <b>Claim 1(f): “wherein said word identification uses one or more distances that were calculated by the second programmable device,”</b> .....                            | <b>167</b> |
| <b>G.</b>    | <b>Claim 1(g): “wherein said identification of spoken words uses one or more distances calculated from a first feature vector; and”</b> .....                             | <b>172</b> |
| <b>H.</b>    | <b>Claim 1(h): “a search stage for using the calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words.”</b> .....       | <b>173</b> |
| <b>XI.</b>   | <b><i>OPINIONS REGARDING GROUND 2: CLAIMS 2-3 ARE OBVIOUS OVER JIANG, SMYTH, AND NGUYEN</i></b> .....                                                                     | <b>177</b> |
| <b>A.</b>    | <b>Dependent Claim 2</b> .....                                                                                                                                            | <b>177</b> |
| <b>B.</b>    | <b>Dependent Claim 3</b> .....                                                                                                                                            | <b>185</b> |
| <b>XII.</b>  | <b><i>OPINIONS REGARDING GROUND 3: CLAIM 4 IS OBVIOUS OVER JIANG, SMYTH, NGUYEN, AND BOIKE</i></b> .....                                                                  | <b>187</b> |
| <b>A.</b>    | <b>Dependent Claim 4</b> .....                                                                                                                                            | <b>187</b> |
| <b>XIII.</b> | <b><i>OPINIONS REGARDING GROUND 4: CLAIM 5 IS OBVIOUS OVER JIANG, SMYTH, NGUYEN, AND BAUMGARTNER</i></b> .....                                                            | <b>193</b> |
| <b>A.</b>    | <b>Dependent Claim 5</b> .....                                                                                                                                            | <b>193</b> |
| <b>XIV.</b>  | <b><i>OPINIONS REGARDING GROUND 5: CLAIM 6 IS OBVIOUS OVER JIANG, SMYTH, NGUYEN, BAUMGARTNER, AND BOIKE</i></b> .....                                                     | <b>203</b> |
| <b>A.</b>    | <b>Dependent Claim 6</b> .....                                                                                                                                            | <b>203</b> |

***XV. CONCLUSION..... 206***

**CLAIM LISTING**

**Claim 1:**

**Claim 1[Pre]** A speech recognition system comprising:

**1(a)** a first programmable device programmed to calculate a feature vector from a digital audio stream, wherein the feature vector comprises a plurality of extracted and/or derived quantities from said digital audio stream during a defined audio time frame;

**1(b)** a second programmable device programmed to calculate distances indicating the similarity between a feature vector and a plurality of acoustic states of an acoustic model wherein said feature vector is received by the second programmable device after it is calculated by the first programmable device; and

**1(c)** a third programmable device programmed to identify spoken words in said digital audio stream using Hidden Markov Models and/or Neural Networks wherein said word identification uses one or more distances that were calculated by the second programmable device, wherein said identification of spoken words uses one or more distances calculated from a first feature vector; and

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

**Claim 2:**

The speech recognition system of claim 1 wherein the time period taken for said identification of spoken words overlaps the time period taken for said second programmable device to calculate one or more distances from a second feature vector.

**Claim 3:**

The speech recognition system of claim 2 wherein the first programmable device, the second programmable device, and third programmable device are physically distinct processing circuits.

**Claim 4:**

The speech recognition system of claim 3 wherein said physically distinct processing circuits are all formed on a single integrated circuit.

**Claim 5:**

The speech recognition system of claim 2 wherein the first programmable device and the second programmable device comprise one physical uni-processor or multi-processor, and the second programmable device is a physically distinct processing circuit.

**Claim 6:**

The speech recognition system of claim 5 wherein said one physical uni-processor or multi-processor and said second programmable device are all formed on a single integrated circuit.

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. 10,062,377 (“’377 Patent”) (Ex. 1001) including the specification and claims, and the ’377 Patent’s prosecution history (“’377 File History”) (Ex. 1002). In developing my opinions relating to the ’377 Patent, I have considered the materials cited or discussed herein, including those itemized in the Exhibit Table below.

| <b><u>Exhibit</u></b> | <b><u>Description</u></b>                                                                                                            |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>Exhibit 1001</b>   | U.S. Patent No. 10,062,377 (“’377 Patent”)                                                                                           |
| <b>Exhibit 1002</b>   | Prosecution History for the ’377 Patent (“’377 File History”)                                                                        |
| <b>Exhibit 1004</b>   | U.S. Patent No. 6,374,219 to Jiang (“Jiang”)                                                                                         |
| <b>Exhibit 1005</b>   | U.S. Patent No. 5,819,222 to Smyth et al. (“Smyth”)                                                                                  |
| <b>Exhibit 1006</b>   | U.S. Patent No. 6,959,376 to Boike et al. (“Boike”)                                                                                  |
| <b>Exhibit 1007</b>   | U.S. Patent Application Publication No. 2002/0049582 to Baumgartner et al. (“Baumgartner”)                                           |
| <b>Exhibit 1008</b>   | Harry Newton. <i>Newton’s Telecom Dictionary</i> . 19 <sup>th</sup> Edition CMP Books March 2003 (“Newton”)                          |
| <b>Exhibit 1009</b>   | Kevin L. Kloker <i>The Motorola DSP56000 Digital Signal Processor</i> IEEE Micro Vol. 6, Issue 6, pp. 29-48 December 1986 (“Kloker”) |
| <b>Exhibit 1010</b>   | Christopher Schmandt <i>Voice Communication with Computers</i> Van Nostrand Reinhold 1994 (“Schmandt”)                               |

Declaration of Christopher Schmandt  
Patent No. 10,062,377

|                     |                                                                                                                                                                                        |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Exhibit 1011</b> | U.S. Patent No. 6,499,012 to Peters et al. (“ <i>Peters</i> ”)                                                                                                                         |
| <b>Exhibit 1012</b> | Ph.D. Thesis of Mosur Ravishankar (“ <i>Ravishankar</i> ”)                                                                                                                             |
| <b>Exhibit 1013</b> | J. Mariani <i>Recent Advances in Speech Processing</i> IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 429-440 May 1989 (“ <i>Mariani</i> ”)            |
| <b>Exhibit 1015</b> | Lawrence Rabiner and Biing-Hwang Juang <i>Fundamentals of Speech Recognition</i> Prentice Hall PTR 1993 (“ <i>Rabiner</i> ”)                                                           |
| <b>Exhibit 1016</b> | Richard Klevans and Robert Rodman <i>Voice Recognition</i> Artech House 1997 (“ <i>Klevans</i> ”)                                                                                      |
| <b>Exhibit 1017</b> | U.S. Patent No. 7,120,582 to Young et al. (“ <i>Young</i> ”)                                                                                                                           |
| <b>Exhibit 1018</b> | John Holmes and Wendy Holmes, <i>Speech Synthesis and Recognition</i> 2 <sup>nd</sup> Edition, Taylor & Francis, 2001 (“ <i>Holmes</i> ”)                                              |
| <b>Exhibit 1019</b> | U.S. Patent No. 4,926,488 to Nadas et al. (“ <i>Nadas</i> ”)                                                                                                                           |
| <b>Exhibit 1020</b> | Frederick Jelinek <i>Statistical Methods for Speech Recognition</i> Third Printing Massachusetts Institute of Technology 2001 (“ <i>Jelinek</i> ”)                                     |
| <b>Exhibit 1021</b> | Shigeru Katagiri <i>Handbook of Neural Networks for Speech Processing</i> Artech House, Inc. 2000 (“ <i>Katagiri</i> ”)                                                                |
| <b>Exhibit 1022</b> | Lawrence Rabiner <i>A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition</i> Proceedings of the IEEE Vol. 77, No. 2 February 1989 (“ <i>Rabiner89</i> ”) |
| <b>Exhibit 1023</b> | U.S. Patent No. 6,182,037 (“ <i>Maes</i> ”)                                                                                                                                            |
| <b>Exhibit 1024</b> | U.S. Patent Application Publication No. 2003/02207 to Aoyama et al. (“ <i>Aoyama</i> ”)                                                                                                |
| <b>Exhibit 1025</b> | U.S. Patent No. 5,895, 447 to Ittycheriah et al. (“ <i>Ittycheriah '447</i> ”)                                                                                                         |
| <b>Exhibit 1026</b> | U.S. Patent No. 6,092,045 to Stubbley et al. (“ <i>Stubbley</i> ”)                                                                                                                     |
| <b>Exhibit 1027</b> | U.S. Patent No. 6,151,574 to Lee et al. (“ <i>Lee</i> ”)                                                                                                                               |
| <b>Exhibit 1028</b> | U.S. Patent No. 6,580,814 to Ittycheriah et al. (“ <i>Ittycheriah</i> ”)                                                                                                               |
| <b>Exhibit 1029</b> | U.S. Patent No. 6,452,348 to Toyoda (“ <i>Toyoda</i> ”)                                                                                                                                |
| <b>Exhibit 1030</b> | U.S. Patent No. 6,195,634 to Dudemaine et al. (“ <i>Dudemaine</i> ”)                                                                                                                   |
| <b>Exhibit 1031</b> | U.S. Patent No. 5,893,058 to Kosaka (“ <i>Kosaka</i> ”)                                                                                                                                |
| <b>Exhibit 1032</b> | U.S. Patent Application Publication No. 2001/0010039 to Yang (“ <i>Yang</i> ”)                                                                                                         |
| <b>Exhibit 1033</b> | U.S. Patent No. 5,754,878 to Asghar et al. (“ <i>Asghar</i> ”)                                                                                                                         |
| <b>Exhibit 1034</b> | U.S. Patent No. 6,041,400 to Ozcelik et al. (“ <i>Ozcelik</i> ”)                                                                                                                       |
| <b>Exhibit 1035</b> | Declaration of June Ann Munford (“ <i>Munford</i> ”)                                                                                                                                   |
| <b>Exhibit 1036</b> | U.S. Patent No. 5,699,456 to Brown et al. (“ <i>Brown</i> ”)                                                                                                                           |
| <b>Exhibit 1038</b> | U.S. Patent No. 4,567,606 to Vensko et al. (“ <i>Vensko</i> ”)                                                                                                                         |
| <b>Exhibit 1039</b> | U.S. Patent No. 6,182,203 to Simar, Jr. et al. (“ <i>Simar</i> ”)                                                                                                                      |
| <b>Exhibit 1047</b> | U.S. Patent No. 6,879,954 to Nguyen et al. (“ <i>Nguyen</i> ”)                                                                                                                         |

Declaration of Christopher Schmandt  
Patent No. 10,062,377

|                     |                                                                                                                                                                                                                                                                        |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Exhibit 1052</b> | U.S. Patent No. 4,718,094 to Bahl et al. (“ <i>Bahl</i> ”)                                                                                                                                                                                                             |
| <b>Exhibit 1053</b> | 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> 1996 (“ <i>Ney</i> ”)                                                      |
| <b>Exhibit 1054</b> | U.S. Patent No. 7,058,678 to Luick et al. (“ <i>Luick</i> ”)                                                                                                                                                                                                           |
| <b>Exhibit 1057</b> | U.S. Patent No. 5,983,180 to Robinson (“ <i>Robinson</i> ”)                                                                                                                                                                                                            |
| <b>Exhibit 1058</b> | Abdulmesih Aktas et al. “Multi-DSP and VQ-ASIC Based Acoustic Front-End for Real-Time Speech Processing Tasks” First European Conference on Speech Communication and Technology EUROSPEECH ’89. Paris, France. September 1989 ( <i>Atkas 1989</i> )                    |
| <b>Exhibit 1059</b> | Abdulmesih Aktas et al. “Speaker Independent Continuous Hmm-Based Recognition of Isolated Words on A Real-Time Multi-Dsp System” Second European Conference on Speech Communication and Technology EUROSPEECH ’91. Genova, Italy. September 1991 ( <i>Atkas 1991</i> ) |
| <b>Exhibit 1060</b> | U.S. Patent No. 6,029,221 to Wu et al. (“ <i>Wu</i> ”)                                                                                                                                                                                                                 |
| <b>Exhibit 1067</b> | U.S. Patent No. 7,697,641 to Haddadin et al. (“ <i>Haddadin</i> ”)                                                                                                                                                                                                     |
| <b>Exhibit 1068</b> | U.S. Patent No. 4,745,544 to Renner et al. (“ <i>Renner</i> ”)                                                                                                                                                                                                         |
| <b>Exhibit 1069</b> | U.S. Patent No. 4,908,781 to Levinthal et al. (“ <i>Levinthal</i> ”)                                                                                                                                                                                                   |
| <b>Exhibit 1070</b> | U.S. Patent No. 6,603,990 to Zhang et al. (“ <i>Zhang</i> ”)                                                                                                                                                                                                           |

11. I have considered these materials from the viewpoint of a POSITA as of the priority date of the ’377 Patent. For the purposes of this declaration, I have been asked to assume that the priority date of the ’377 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 ’377 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 ’377

Patent's priority date (again, unless expressly stated otherwise). I merely use the present verb tense for ease of reading.

### **III. OVERVIEW AND 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 review 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 '377 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:

- 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;
- 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.

**IV. LEVEL OF A PERSON OF ORDINARY SKILL**

24. Based on my review and analysis of the ’377 Patent, the prior art cited herein, and the ordinary skill factors described in this section, a POSITA in the field

of the '377 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. *See, e.g., '377 Patent* (Ex. 1001), 1:18-2:20 (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 '377 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.

25. 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).

26. 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

27. 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

28. Speech recognition using computer systems was known and widely used well before the priority date of the '377 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.*

29. 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.*

30. 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.

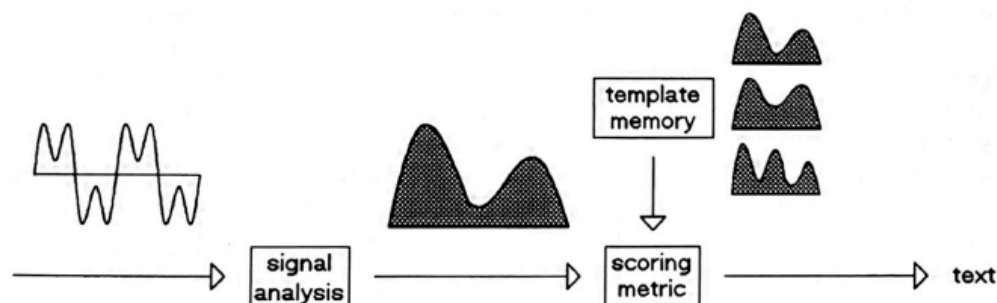


**Figure 7.1.** The functional elements of a speech recognizer. The user's speech is digitized and converted to the recognizer's internal representation. The captured speech is then compared with the words stored in the recognizer's template memory. The pattern matching algorithm then determines which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

*Schmandt*, 46 ("Figure 7.1").

31. 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* 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.

32. 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.*

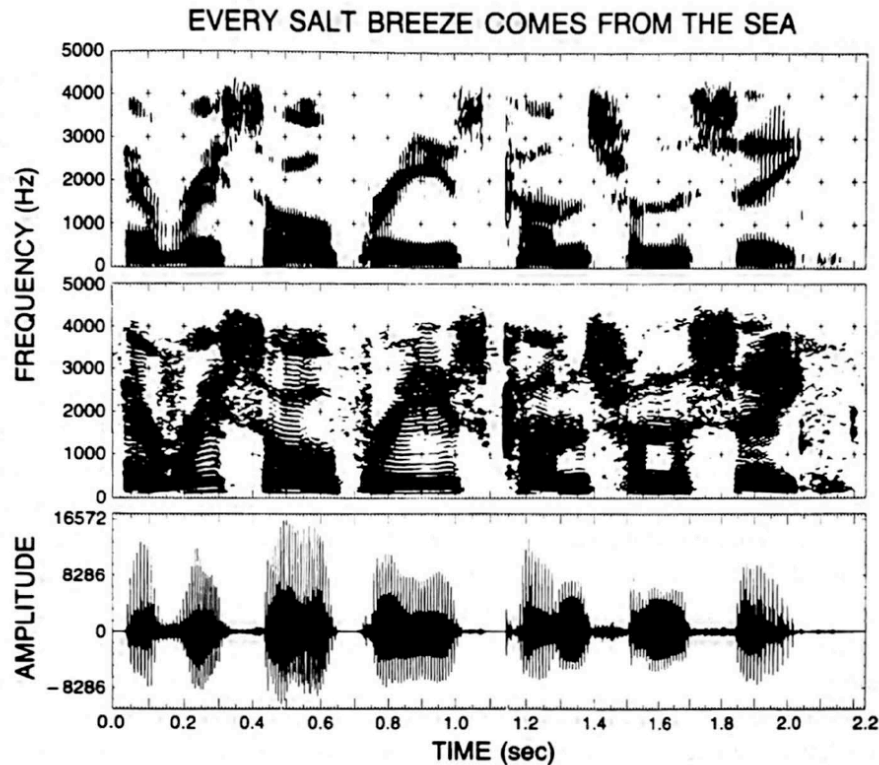


**Figure 2.8** Wideband and narrowband spectrograms and speech amplitude for the utterance “Every salt breeze comes from the sea.”

*Rabiner*, 50 (“Figure 2.8”).

33. One speech recognition technique known well before the priority date of the '377 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.*

34. It was also known to use lexical trees to organize vocabulary before the priority date of the '377 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.

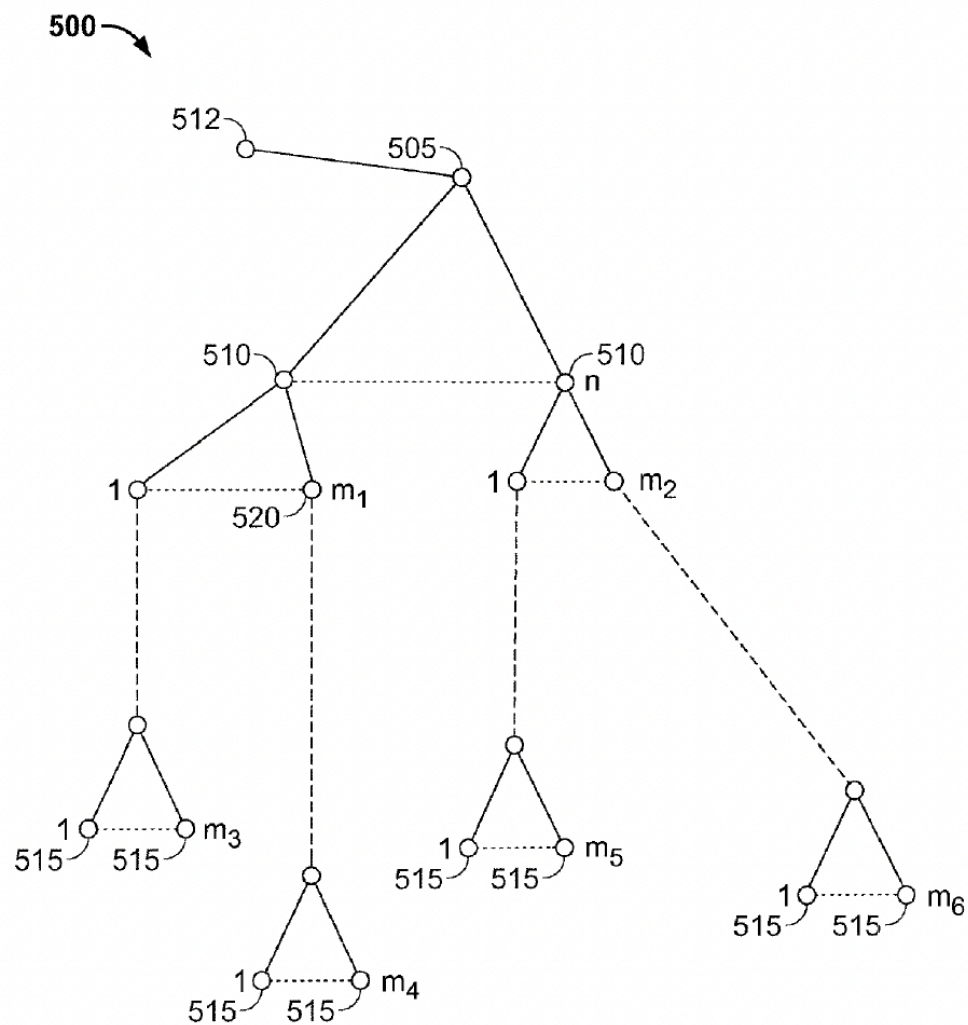


**FIG. 5**  
**(Prior Art)**

*Young, Fig. 5.*

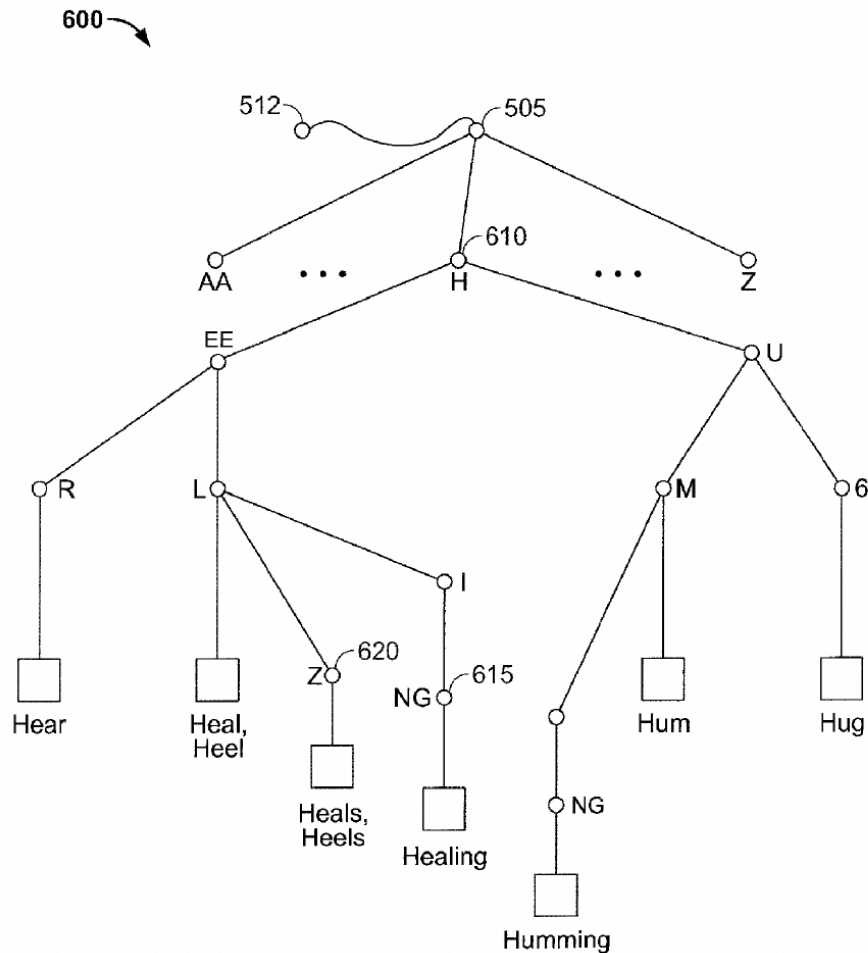


**FIG. 6**  
(Prior Art)

*Young*, Fig. 6.

## **B. Feature Vectors**

35. The '377 Patent describes feature vectors as comprising “a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame[.]” ’377 Patent, 5:66-6:1; *see also id.*, 6:23-25, 6:38-41, 6:51-54, 6:62-64. Using feature vectors in speech recognition systems was well known before the priority date of the '377 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.*

36. 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.

37. 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.

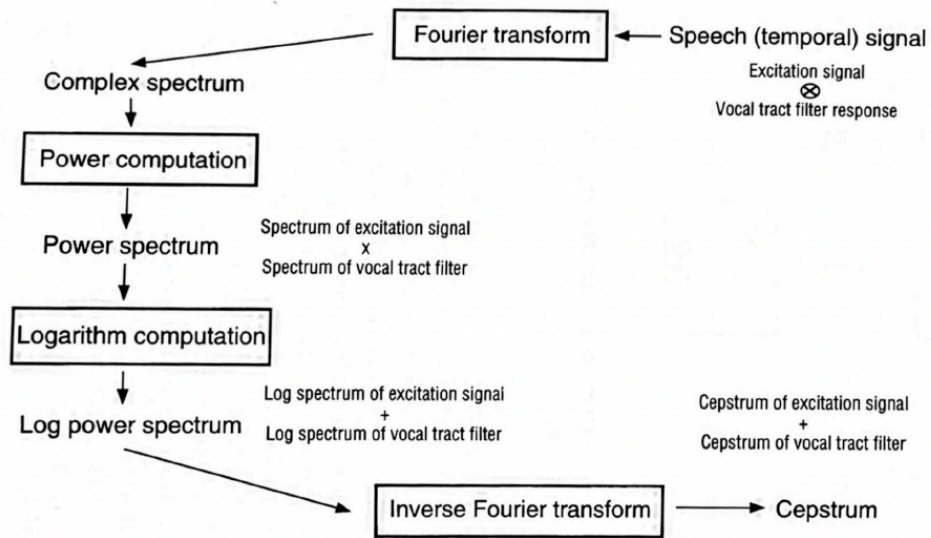


**Figure 3.21** Relationships among a speech signal, a vocal tract filter, and a vocal source excitation signal.

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

38. 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. *Rabiner*<sup>89</sup> 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.” *Rabiner*<sup>89</sup> (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  $O_t$  is computed.” *Id.*

39. 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.”).

40. 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 Frequency<sup>1</sup> 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.**

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

---

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

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].

41. 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 '377 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.” ’377 Patent, 13:19-23.

42. *Stubley* 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.

*Stubley* (Ex. 1026), 11:30-36.

43. *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**

44. It was also known before the priority date of the '377 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.*

45. 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.

46. 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.

47. 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.”).

48. 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**

49. Before the priority date of the '377 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 of 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.*

50. *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.

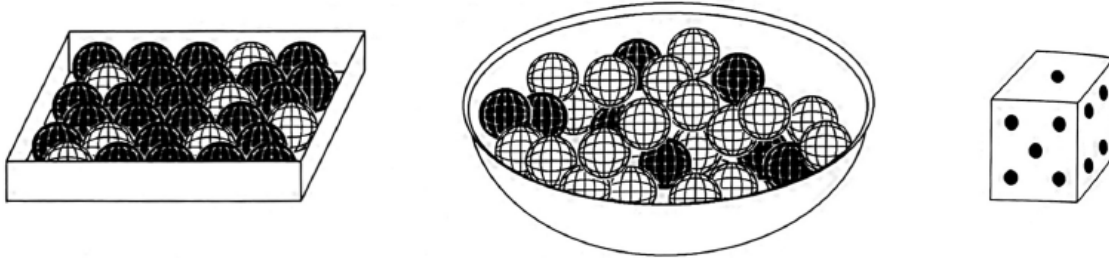


**Figure 7.10.** A box and a bowl full of colored balls illustrate a Hidden Markov Model.

*Schmandt*, 59 (“Figure 7.10”).

51. 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.

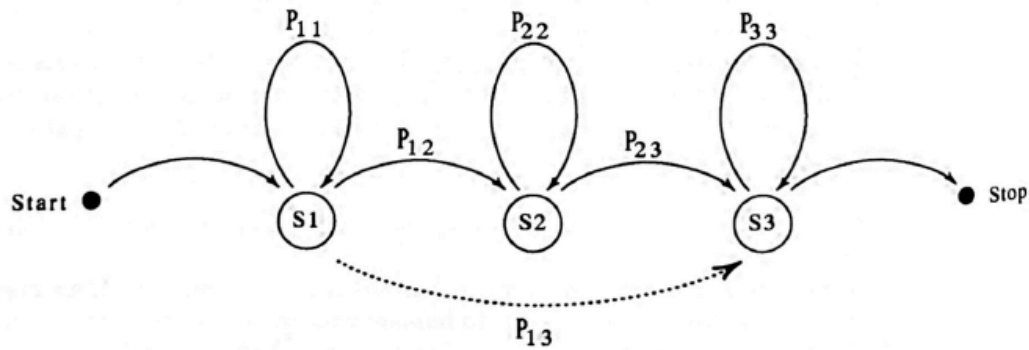


Figure 7.11. A Markov Model.

*Schmandt*, 59 (“Figure 7.11”).

52. *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).

53. *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.*

54. *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).

55. Another example of using an HMM for speech recognition well before the priority date of the '377 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*, 9:68-10:4.

56. 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].

57. In yet another example, *Stubbley* 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.

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

58. 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).

59. 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**

60. 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 '377 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.

61. 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).

62. 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.

63. 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 ’377 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 “[t]he ‘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.

64. 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).

65. 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[.]”).

66. 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).

67. It was also known to use neural networks as distance calculations before the priority date of the '377 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**

68. It was known to use Gaussian distributions and probabilities before the priority date of the '377 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).

69. 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).**

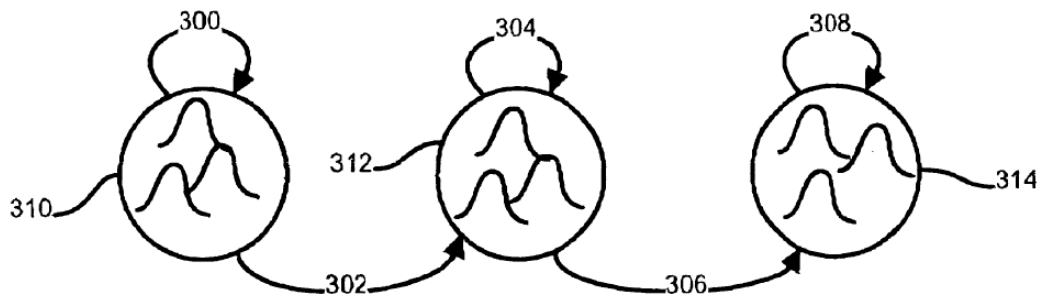
*Peters* (Ex. 1011), 1:35-48. *Peters* further teaches states of Hidden Markov Models associated with parameters that define Gaussian mixture of probability density functions.

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**

70. It was known in the art before the priority date of the '377 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.

71. It was known in the art before the priority date of the '377 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.").

72. 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.

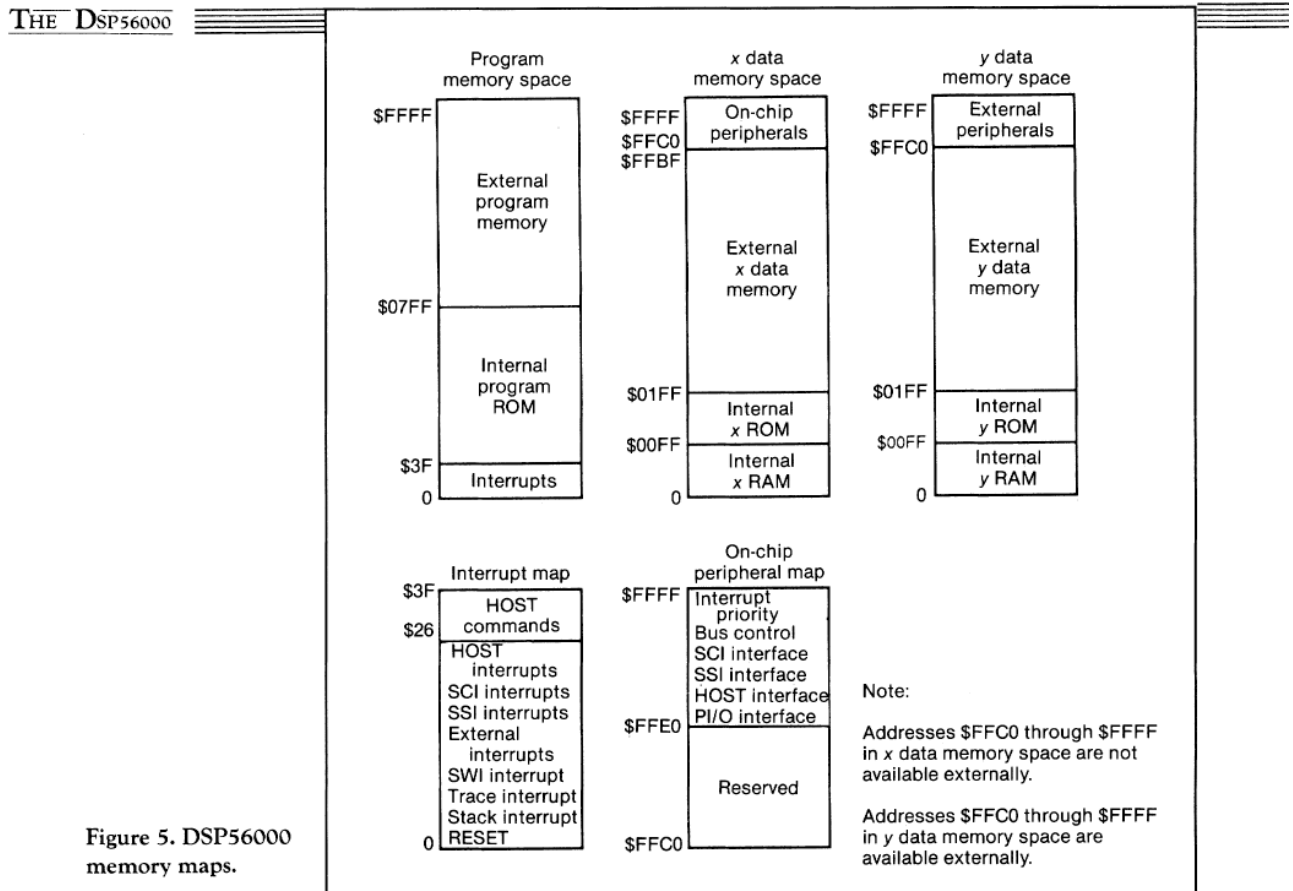


Figure 5. DSP56000 memory maps.

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

73. It was also known to have a calculating apparatus and acoustic model memory on the same integrated circuit before the priority date of the '377 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.

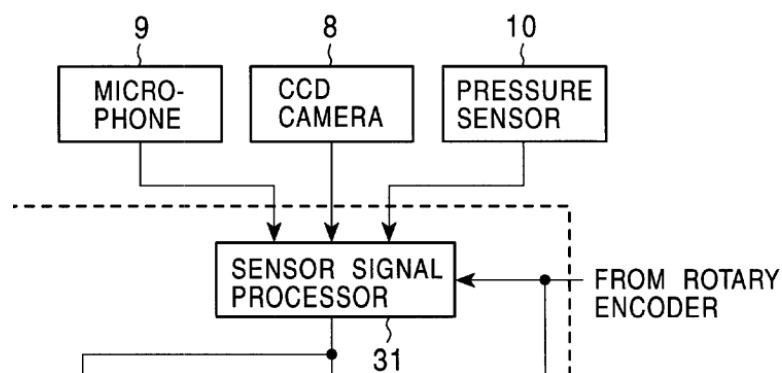
74. *Toyoda* teaches a voice recognition module (sensor signal processor 31) that uses an HMM. *Toyoda*, 7:48-50. Furthermore, *Toyoda* teaches referencing an acoustic model memory that stores an acoustic model representing phonemes while a matching unit 42 recognizes voice input.

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.

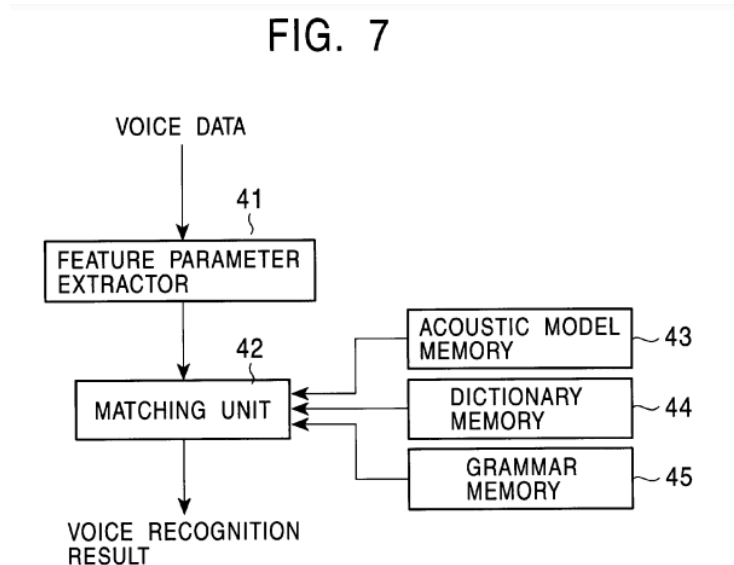
The acoustic model memory 43 stores an acoustic model that represents an acoustic feature such as phonemes and syllables in a voice to be recognized. Since voice recognition is here carried out using the continuous HMM, an HMM is employed.

*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.



*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).

75. Additionally, it was known to use a single integrated circuit containing multiple digital signal processors (DSPs) before the priority date of the '377 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).

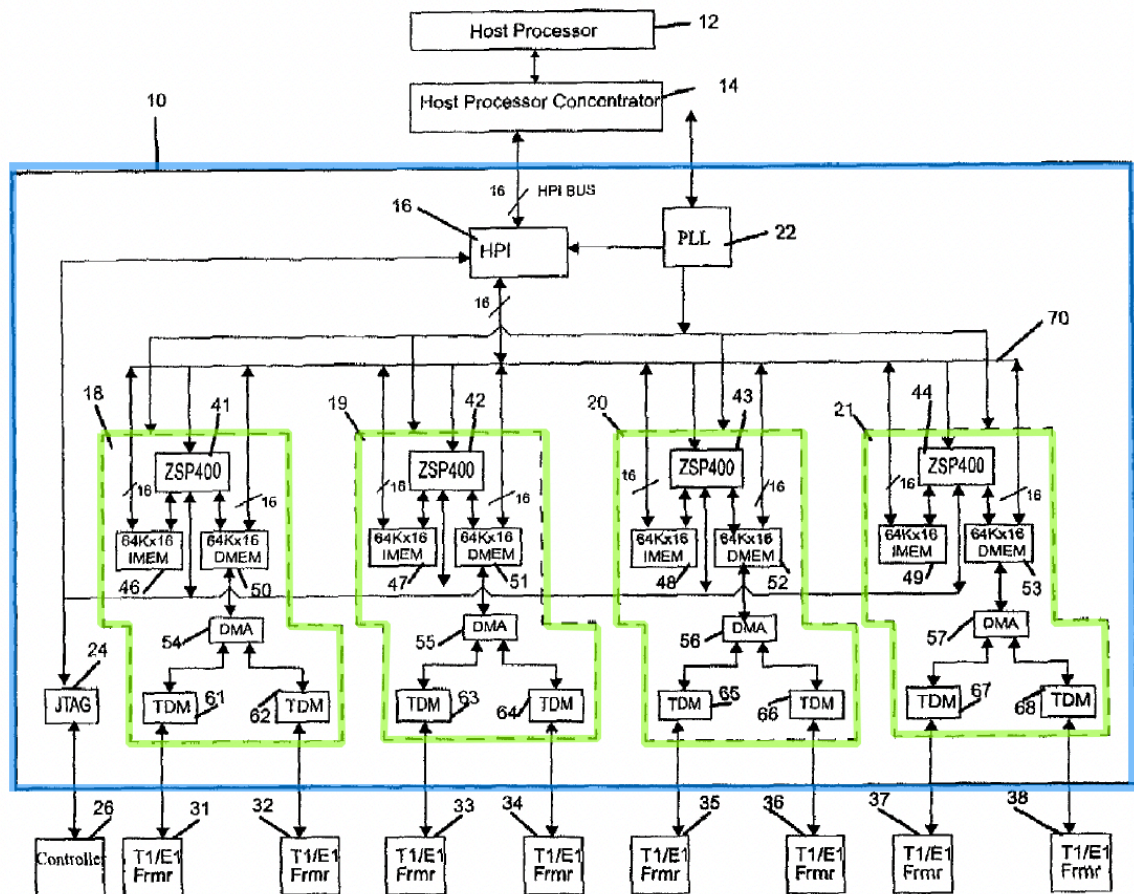


FIGURE 1

*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*, 2:3-9, 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.

76. 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**

77. It was well known prior to the '377 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.

78. 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.

79. 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:

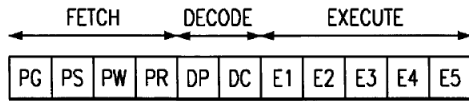


FIG. 11

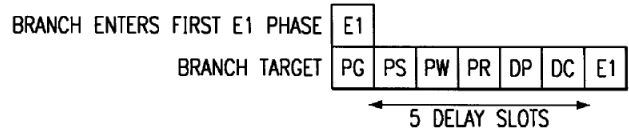


FIG. 12

| FETCH<br>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).

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

81. As an example of pipelining in speech recognition, Ravishankar<sup>2</sup> 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.

82. 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.

83. 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 vector quantizer 110, a mixture probability processor 115, a phone probability processor 120, a word probability processor 125, a

---

<sup>2</sup> 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.

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.

84. 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**

85. Interrupt signals in speech recognition systems to inform different processors when data was finished or ready was well-known in general prior to the '789 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.

86. 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.

87. 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.

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

**J. Prior Art Speech Recognition Systems**

88. Numerous speech recognition systems were well known before the priority date of the '377 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.

89. 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.

90. 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 '377 PATENT**

91. I have reviewed the '377 Patent (Ex. 1001). Descriptions of the '377 Patent that may be pertinent to my analysis are provided herein, and my descriptions of the '377 Patent are intended only as a summary of the '377 Patent and/or intended as being pertinent to the Challenged Claims of the '377 Patent. My opinions provided herein are not intended as a discussion of my entire understanding of the '377 Patent or any other claims in any related patent to the '377 Patent. The '377 Patent describes a speech recognition system comprising several programmable devices for performing steps of the speech recognition process. '377 Patent, 5:63-6:6, 6:20-33, Fig. 16, Claim 1. A first programmable device calculates a feature vector from a digital audio stream; a second programmable device calculates distances indicating a similarity between a feature vector and a plurality of acoustic vectors; a third programmable device identifies spoken words in said digital audio stream using Hidden Markov Models and/or Neural Networks. '377 Patent, 12:56-63, 13:1-35, 24:5-11, Claim 1.

92. The first programmable device provides the functionality of front end 103, which calculates a feature vector representative of the spectral components of

an audio signal during a given timeframe. *'377 Patent*, 12:56-63. The second programmable device provides the functionality of the distance engine and takes the calculated feature vector and calculates the distances between the feature vector and each state to determine the similarity of the feature vector to the predefined states. *'377 Patent*, 13:1-35. The third programmable device provides the functionality of the search stage, and identifies words from the input audio signal using the distances calculated in the second programmable device. *'377 Patent*, 24:5-11.

## **VII. THE PROSECUTION HISTORY OF THE '377 PATENT**

93. I have reviewed the prosecution history for the '377 Patent, (Ex. 1002). Pertinent aspects of the prosecution history of the '377 Patent are described below.

### **A. Prosecution History of the '377 Patent**

94. In a Non-Final Rejection dated May 9, 2017, I see that the Examiner objected to then-pending Claims 2-17 because “‘acoustic coprocessor’, ‘first, second, and third programmable device’ are not found in the Specification. The Applicant was advised to show where in the Specification support for the new claims are. Appropriate correction is required.” *'377 File History*, 301.

95. In response, I note that Applicant argued that the specification and drawings of the application support the terms “a first programmable device,” “a second programmable device,” and “a third programmable device.” *'377 File History*, 308.

Paragraph [0035] of the published application states: "In particular, the other stages may be implemented on a CPU, or on a DSP and CPU. The "DSP" and "CPU" may each be implemented as software programmable devices."

*'377 File History, 307.*

See also the headings of Fig. 22 which include "Front end on DSP", "Distance Calculation Accelerator", and "Search stage on CPU". Given that a DSP is a well-known programmable device, and a CPU is a well-known programmable device, and the quotes from the specification given above indicate that the "distance calculation engine may be included within an accelerator" and "the accelerator may be implemented in software or in hardware, or in a combination," it is clear that

each of "Front end on DSP", "Distance Calculation Accelerator", and "Search stage on CPU" can be (in some embodiments) three different programmable devices.

*'377 File History, 307-308.*

96. In my opinion, Applicant admitted that digital signal processors (DSPs) are a form of programmable device.

## **VIII. SUMMARY OF UNPATENTABILITY**

97. In my opinion, it was known that speech recognition systems that utilized Hidden Markov Models (HMMs) generally included three main steps: 1) extracting feature vectors from an audio signal (i.e., speech utterances); 2) computing a probability/likelihood that a particular feature vector is modeled by a particular subword unit (e.g., phoneme, allophone) that is associated with an HMM; and then 3) performing a word identification process using a prefix tree (i.e., lexical tree) that concatenates the phonemes/allophones to determine the word that was

spoken. I note that the '377 Patent's claims recite these three steps that were all well known in the prior art. Performing said speech recognition steps using distinct programmable devices is the purported point of novelty of the '377 Patent.

98. As I opine below, *Jiang* teaches nearly every limitation of Claim 1 including performing the three aforementioned speech recognition steps. Specifically, *Jiang* teaches feature extraction performed on a first processor, distance/likelihood calculations and word identification that are both performed by a tree search engine 74, which *Jiang* teaches may be a processor. Thus, *Jiang* teaches at least two processors for performing said three steps and *Jiang* contemplates various computer system configurations including implementing the invention on a multi-processor system. *Smyth*, another prior art reference, also teaches a speech recognition system that performs the three aforementioned speech recognition steps on three distinct programmable devices (e.g., processors, DSPs).

## **IX. OVERVIEW OF THE PRIOR ART**

### **A. Overview of Jiang**

99. *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.

100. 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 '377 Patent. Compare, *Jiang*, Abstract, 4:8-27 with '377 Patent, Abstract, 1:18-21. 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 '377 Patent. Compare *Jiang*, 4:51–5:3, 6:46-7:10, 7:30-47 with '377 Patent, 1:18-29, 2:21-24.

#### **B. Overview of Smyth**

101. *Smyth* discloses a token-based speech recognition system with a recognition processor 3. *Smyth* (Ex. 1005), Abstract, 4:38-45, Fig. 1. *Smyth* discloses recognition processor 3 comprises several programmable devices including a feature extractor 33, classifier 34 with classifier processor 341, and sequencer 35 with parsing processor 351. *Smyth* at generally 5:1-6:63, Fig. 2. *Smyth* also teaches embodiments utilizing three DSPs or two DSPs and a processor for components such as feature extractor 33, classifier processor 341, and parsing processor 351. *Smyth*, 5:49-53, 6:5-8, 6:47-51.

102. In my opinion, *Smyth* teaches a speech recognition system using multiple programmable devices 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 ’377 Patent. For example, *compare Smyth*, 5:20-46, 5:58-6:4, 6:36-63 *with the ’377 Patent*, 1:18-19, 12:56-63, 13:1-35, 24:5-11. It is also my opinion that *Smyth* is reasonably pertinent to a problem faced by the inventors of the ’377 Patent because both recognize challenges with performing speech recognition tasks while using a limited set of computing resources.

### C. Overview of Nguyen

103. *Nguyen* teaches a speech recognition system having a plurality of acoustic models that performs feature vector extraction, distance/probability calculations, and word identification. *Nguyen* (Ex. 1047), Abstract, 1:8-50, 2:8-4:36.

104. In my opinion, *Nguyen* teaches a “large vocabulary continuous speech recognition system” and “a method for improving pattern matching in a large vocabulary continuous speech recognition system” and therefore *Nguyen* is in the same field of endeavor as the ’377 Patent. *Compare, Nguyen*, Abstract, 1:1-26 *with ’377 Patent*, Abstract, 1:18-21. Further, because *Nguyen* is directed to a problem of performing speech recognition, *Nguyen* is reasonably pertinent to a problem faced by the inventors of the ’377 Patent. *Compare Nguyen*, 1:8-50, 2:8-4:36 *with ’377 Patent*, 1:18-29, 2:21-24.

**D. Overview of Boike**

105. *Boike* teaches multiple digital signal processors (DSPs) on a single integrated circuit (IC). *Compare* '377 Patent, 37:60-62 (referring to a System on Chip (SoC) design), Fig. 1A, *with Boike*, Abstract, 1:6-8 (“The invention relates to digital signal processing, and more particularly to integrated circuits containing multiple digital signal processing cores.”), 2:3-35, 2:52-63, Figs. 1-2. Therefore, in my opinion, *Boike* is in the same field of endeavor as the '377 Patent. Further, because *Boike* is directed to a problem of dividing processing between distinct processors, *Boike* is reasonably pertinent to a problem faced by the inventors of the '377 Patent. *Compare* '377 Patent, 24:28-33 *with Boike*, 2:21-35, 2:52-63.

**E. Overview of Baumgartner**

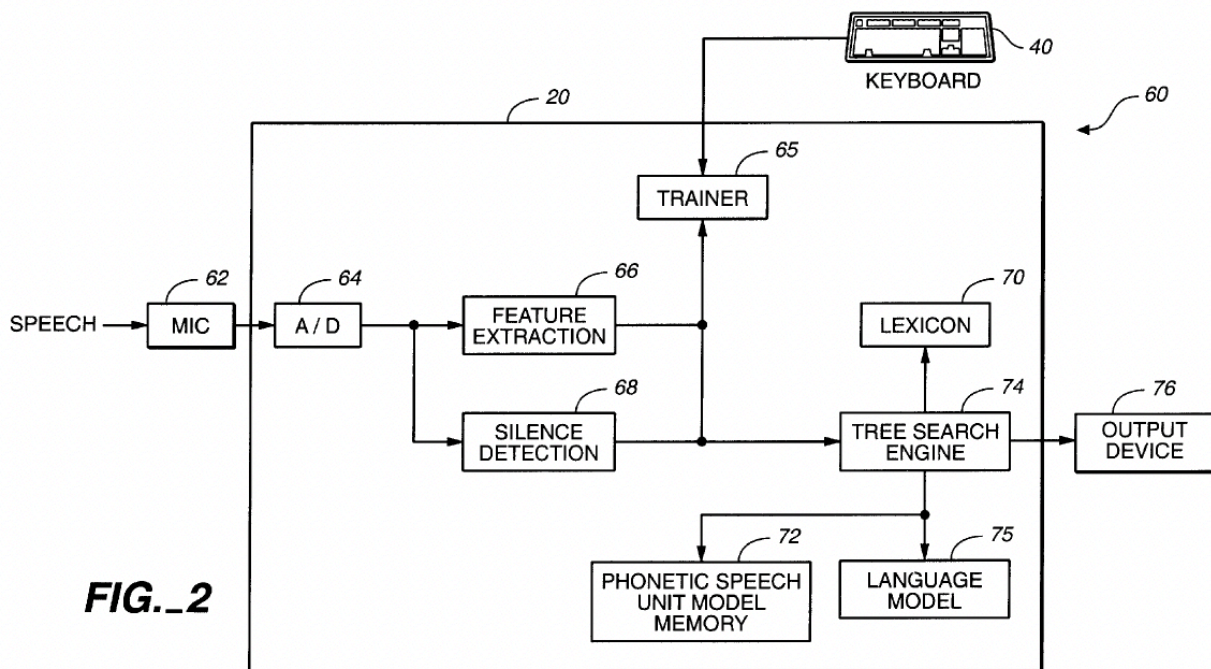
106. *Baumgartner* teaches accelerators for speech recognition systems. *Compare* '377 Patent, Abstract, Fig. 21, 30:60–31:5 *with Baumgartner*, Abstract, [0002], [0007]-[0010], Fig. 19. Therefore, in my opinion, *Baumgartner* is in the same field of endeavor as the '377 Patent. Further, because *Baumgartner* 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 '377 Patent. *Compare* '377 Patent, 1:30-47, 2:21-24 *with Baumgartner*, [0005].

**X. OPINIONS REGARDING GROUND 1: CLAIM 1 IS OBVIOUS OVER  
JIANG IN VIEW OF SMYTH**

**A. Claim 1(a): “a first programmable device programmed to calculate  
a feature vector from a digital audio stream,”**

**1. Jiang Alone Teaches Claim 1(a)**

107. Regarding the claimed “a digital audio stream,” *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 Fig. 2 below.



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

108. *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 digital audio stream.

109. Regarding feature extraction module 66, *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 “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. 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.

110. 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 "to calculate a feature vector," as claimed. In particular, *Jiang* teaches a feature vector is encoded, which a POSITA would have understood to mean *calculated*, as claimed. I note that the '377 Patent does not provide express descriptions of steps for how feature vectors are calculated nor do the claims recite anything more than merely "calculating" a feature vector. See '377 Patent, 15:60-61 ("At the start of each time frame, the audio input is converted into a feature vector[.]"), 23:26-67 ("At the start of the time frame, the feature vector is generated[.]"), 12:56-58 ("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 calculate (i.e., generating, encoding, converting) a feature vector

from a frame of a digital audio sample, and *Jiang*'s teachings of encoding a feature vector from digital audio samples teaches or at least renders obvious Claim 1(a).

111. It is also my opinion that *Jiang* teaches the claimed "first programmable device." Specifically, *Jiang* teaches "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." *Jiang*, 6:53-57. A POSITA would have understood that feature extraction module 66 implemented as a conventional array processor would have been a dedicated, programmed processor for performing disclosed tasks, including computing a magnitude value from inputted data received from A/D converter 64, for example. Indeed, it was known in the art that array processors were programmable devices. *Haddadin* (Ex. 1067), 2:65-3:2 ("programmable devices such as ... array processor"); *Renner* (Ex. 1068), 3:13-15 ("programmable array processor"); *Levinthal* (Ex. 1069), 5:53 ("programmable array processor"); *Zhang* (Ex. 1070), 8:13-14 ("programmable computer array processor"). *Jiang*'s teaching of implementing feature extraction module 66 as a conventional array processor would have indicated to a POSITA that *Jiang* envisions utilizing a dedicated processor for feature extraction module 66.

112. In another example, *Jiang* teaches implementing feature extraction module 66 on a processor. *Jiang*, 7:16-22 ("As feature extraction module 66 is

processing the digital samples from A/D converter 64, silence detection module 68 is also processing the samples. 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.”). In my opinion, *Jiang* teaches or at least renders obvious feature extraction module 66 is a processor, whether the same processor or a different processor from silence detection module 68’s processor.

113. Further, I note that *Jiang* teaches feature extraction module 66 along with other components may be either hardware modules or software modules.

Training module 65, feature extraction module 66 and silence detection module 68 may be either hardware modules in computer 20, or software modules stored in any of the information storage devices disclosed in FIG. 1 and accessible by CPU 21 or another suitable processor. In addition, lexicon storage module 70 and phonetic speech unit storage module 72 are also preferably stored in any suitable memory devices shown in FIG. 1. 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.

*Jiang*, 6:31-42. *Jiang* also teaches numerous implementations of the invention including computer-executable instructions and program modules executed by a computer and said program modules include routine programs performing particular tasks.

[T]he invention will be described, at least in part, in the general context of computer-executable instructions, such as program modules, being executed by a personal computer. Generally, program modules include routine programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

*Jiang*, 4:54-5:3.

114. In my opinion, a POSITA would have understood that disclosed modules including feature extraction module 66 are programmed modules for performing disclosed tasks. As I discussed above, *Jiang* discloses feature extraction module 66 as a conventional array processor and also discloses CPU 21 of computer 20 programmed to perform the functionality of feature extraction module 66. I further note that during prosecution of the application that resulted in the '377 Patent that Applicant admitted a CPU was a programmable device. *See* Section VII. Therefore, in addition to *Jiang's* disclosures that feature extraction module 66

implemented as a processor, a POSITA would have understood and found obvious that in the embodiment where CPU 21 is programmed to perform the functions of feature extraction module 66, *Jiang's* feature extraction module 66 is a “first programmable device,” as claimed.

**2. *Jiang in Combination with Smyth Teaches Claim 1(a)***

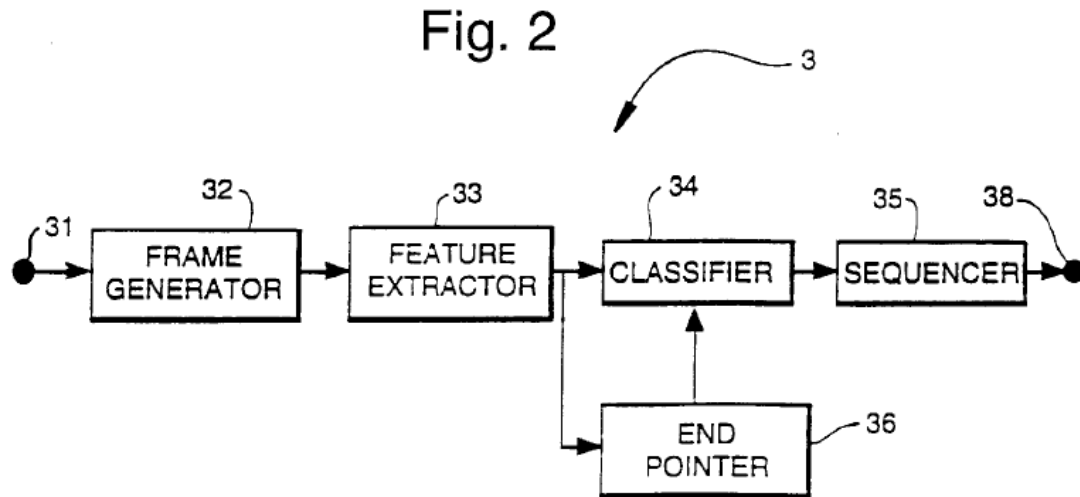
115. It is also my opinion that *Smyth* teaches a digital signal processor (DSP), which is a first programmable device, as claimed. Specifically, *Smyth* teaches a digital signal processor (DSP) that is “suitably programmed.” *Smyth*, 5:49-53 (“The frame generator 32 and feature extractor 33 are, in this embodiment, provided by a suitably programmed digital signal processor (DSP) device (such as the Motorola DSP56000, or the Texas Instruments TMCC320) or similar device.”). I also note that a POSITA would have understood that a DSP is a programmable device, as I discussed above regarding *Newton* and *Kloker*. Indeed, as I discussed above, *Newton's Telecom Dictionary* teaches one of the main advantages of a DSP is its programmability. *See* Section V.G.

116. I also note that during prosecution of the application that later issued as the '377 Patent, the Applicant responded to an objection from the Examiner by stating “a DSP is a well-known programmable device, and a CPU is a well-known programmable device.” '377 *File History*, 301, 307. Therefore, in addition to the reasons I discussed above, a POSITA would have understood that a DSP would have

met the claimed “first programmable device” because such was admitted by the Applicant during prosecution. *See* Section VII (discussing the prosecution history of the ’377 Patent).

117. It is also my opinion that *Smyth* teaches a DSP (i.e., a first programmable device) is “programmed to calculate a feature vector from a digital audio stream” as claimed. Specifically, *Smyth* teaches receiving speech in digital form and generating frames of contiguous samples. I note that *Smyth* teaches a frame generator 32 and frame extractor 33 that are “provided by a single suitably programmed digital signal processor (DSP) device (such as the Motorola DSP 56000, or the Texas Instruments TMC C 320) or similar device.” *Smyth*, 5:49-53. *Smyth* teaches that the frame generator 32 “is arranged to receive speech samples at a rate of, for example, 8,000 samples per second, and to form frames comprising 256 contiguous samples, at a frame rate of 1 frame every 16 ms.” *Smyth*, 5:20-23; *see also Smyth*, 1:24-39 (*Smyth* disclosing it was “common in speech recognition processing” to “input [speech] via a microphone, sample[], digitise[], [and] segment [the speech] into frames of length 10-20 ms (e.g. sampled at 8 kHz).”), 5:19-29. After frame generator 32 generates frames, *Smyth* teaches that feature extractor 33 “receives frames from the frame generator 32 and generates, in each case, a set or vector of features. The features may, for example, comprise cepstral coefficients (for example, LPC cepstral coefficients or mel frequency cepstral coefficients ...) or

differential values of such coefficients[.]” *Smyth*, 5:31-38. Specifically, *Smyth* teaches that feature extractor 33 “generat[es] from a frame of samples a corresponding feature vector[.]” *Smyth*, 5:1-7.



*Smyth*, Fig. 2.

118. Additionally, *Smyth* teaches frame generator 32 receives speech samples and forms frames. *Smyth*, 5:20-29. Furthermore, I note that *Smyth* expressly teaches digitizing speech data was common in speech recognition processing.

It is common in speech recognition processing to input speech data, typically in digital form, to a so-called front-end processor, which derives from the stream of input speech data a more compact, perceptually significant set of data referred to as a front-end feature set or vector. For example, speech is typically input via a microphone, sampled, digitised, segmented into frames of length 10-20 ms (e.g. sampled at 8 kHz) and, for each frame, a set of coefficients is calculated. In speech recognition, the speaker is normally assumed to be speaking one of a set of words or phrases. 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*, 1:24-39. Also, as I discussed above, it was a well-known and common practice to transduce analog speech into a series of digital signal samples. *See* Section V.A (discussing *Klevans* teaches “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.). Therefore, a POSITA would have understood that *Smyth* discloses that it was well known to “digitise” input speech into a digital audio stream for further processing.

119. Additionally, *Smyth* teaches a DSP that is programmed to calculate a feature vector. As I discussed above, *Smyth* teaches the digital audio stream passes through frame generator 32. *Smyth*, 5:1-5, 5:20-29. Then, *Smyth* teaches “feature extractor 33 receives frames from the frame generator 32 and generates, in each case, a set or vector of features.” *Smyth*, 5:31-33, 1:24-39 (in particular, “a front-end processor” “derives from the stream of input speech data a more compact, perceptually significant set of data referred to as a front-end feature set or vector,” and “for each frame, a set of coefficients is calculated.”), 5:6-7 (“a feature extractor

33 for generating from a frame of samples a corresponding feature vector.”). In my opinion, because *Smyth*’s DSP is a “suitably programmed” DSP that generates a feature vector using feature extractor 33, a skilled artisan would have understood that *Smyth*’s DSP is programmed to calculate a feature vector.

120. As I opine below, in my opinion, *Smyth* teaches utilizing a first DSP for the frame generator 32 and feature extractor 33, a second DSP for the classifier processor 341 of classifier 34, and a third DSP (or microprocessor) for the parsing processor 351 of sequencer 35. See Section X.C.5.

### ***3. Motivation to Modify Jiang to Include Smyth’s First Programmable Device***

121. In my opinion, a skilled artisan would have been motivated and would have found it obvious to modify *Jiang* to include *Smyth*’s first programmable device for calculating a feature vector. I note that *Jiang* itself discloses modifications to its hardware architecture and that the invention may be “practiced with other computer system configurations.” *Jiang*, 4:54-65. Specifically, *Jiang* discloses “multi-processor systems,” such that a POSITA would have understood and found obvious to implement *Jiang* as a multi-processor system utilizing multiple processors. *Jiang* already teaches performing the claimed calculating a feature vector utilizing feature extraction module 66. Modifying *Jiang* to use *Smyth*’s DSP for calculating feature vectors would have merely required hardware or software changes well known in the art. In my opinion, because *Jiang* expressly contemplates various computer

system configurations for performing speech recognition processing tasks, modifying *Jiang* according to *Smyth*'s teachings would have had a reasonable expectation of success and further would have yielded predictable results.

122. I also note that *Jiang* is concerned with identifying silences. The '377 Patent also describes identifying silences. *See* '377 Patent, 38:19-20 (describing dummy states for start/end of utterance). In my opinion, modifying *Jiang* to include certain of *Smyth*'s teachings does not frustrate *Jiang*'s purpose related to identifying silences, as *Jiang*'s silences are still incorporated in the modified system. Thus, in my opinion, modifying *Jiang* with *Smyth*'s teachings does not change or alter *Jiang*'s principle of operation or render the modified *Jiang* unsatisfactory for its intended purpose.

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

**1. '377 Patent's Discussion of “extracted” or “derived quantities”**

123. In my opinion, a POSITA would have understood that the claimed a plurality of extracted and/or derived quantities from a digital audio stream during a defined audio time frame would have included spectral components of the feature vector and the component derivatives. For example, the '377 Patent describes calculating 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.

*'377 Patent*, 12:56-63.

124. Further, a POSITA would have understood that Claim 1(b) references the *'377 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 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 *'377 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; *'377 Patent*, 13:19-23). 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).

## **2. Jiang's Teachings**

125. In my opinion, *Jiang* teaches a calculated feature vector "reflect[s] the spectral characteristics for a plurality of frequency bands." *Jiang*, 6:62-7:10, 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 that the ’377 Patent also describes “the audio signal is converted into a feature vector. This may be done by splitting the audio signal into spectral components.” ’377 Patent, 12:56-58, 12:58-62. Therefore, in my opinion, *Jiang* teaches a feature extraction module 66 that extracts quantities from an audio signal where the quantities are indicative of the spectral characteristics of said audio signal. Further, *Jiang* teaches such quantities are encoded into a feature vector. In my opinion, it was also 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[.]”). 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 a metric (i.e., the claimed quantity) 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.”).

126. In my opinion, *Jiang* teaches feature vectors that comprise extracted and/or derived quantities of a digital audio stream “during a defined audio time frame.” For example, *Jiang* teaches feature extraction module 66 divides the digital 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 POSITA would have understood that 10 milliseconds in duration was a “defined audio time frame,” as claimed. Feature extraction module 66 (i.e., first programmable device) then 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 speech input, as was well known in the art as I discussed above.

**C. Claim 1(c): “a second programmable device programmed to calculate distances indicating the similarity between a feature vector and a plurality of acoustic states of an acoustic model”**

**1. *Overview of Jiang’s Teachings Regarding Utilizing HMMs and Prefix Trees for Speech Recognition as Applicable to Claims 1(c)-1(h)***

127. The mapping for Claims 1(c)-1(h) can best be understood with an understanding of basic speech recognition principles, which are taught in *Jiang* and about which I discuss herein. This section provides a background discussion of utilizing HMMs and lexical trees to transcribe spoken speech into words, with references to *Jiang’s* teachings that are relied on for the express mapping in Claims 1(c)-1(h).

128. *Jiang’s* Background of the Invention provides an overview of how HMMs and lexical trees are utilized in speech recognition. *Jiang*, 1:15–3:21. *Jiang* teaches that an HMM includes 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*, 1:21-24; *Ravishankar*, 28 (“An HMM is a set of states connected by transitions” that “model the emission of one frame of speech”). “Consequently, 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.” *Jiang*, 1:43-48. I also note that, per *Jiang*:

In order to find the most likely sequence of states corresponding to a sequence of acoustic vectors, the Viterbi algorithm is employed. The Viterbi algorithm performs a computation which starts at the first frame and proceeds one frame at a time, in a time-synchronous manner. **A probability score is computed for each state in the state sequences (i.e., the HMMs) being considered.** Therefore, **a cumulative probability score is successively computed for each of the possible state sequences** as the Viterbi algorithm analyzes the acoustic signal frame by frame. By the end of an utterance, the state sequence (or HMM or series of HMMs) having the **highest probability score** computed by the Viterbi algorithm provides the most likely state sequence for the entire utterance. The most likely state sequence is then converted into a corresponding spoken subword unit, word, or word sequence.

*Jiang*, 2:1-15, 1:34-40; *Ravishankar*, 32 (discussing determining the “best state sequence”), 19 (discussing “[i]t is necessary to represent words in terms of sub-word units,” such as phonemes, to sufficiently train the speech recognition system, and to model the phonemes using HMMs).

129. In my opinion, *Jiang* teaches that speech is commonly modeled by HMMs, where each HMM comprises a plurality of states. Further, *Ravishankar* teaches the same. *Ravishankar*, 28 (“Most systems use hidden Markov models (HMMs) to represent the basic units of speech.”)). For a spoken speech input having

specific spectra, the spectra are converted into one or more feature vectors. *Jiang*, 1:58-67, 6:62-7:10. Each state of the HMM has an associated probability corresponding to a frame of spoken speech, where each frame may represent a plurality of feature vectors. *Jiang*, 1:21-24, 1:58-66, 6:62-7:10; *Ravishankar*, 32-33, 40-41, 67-68. *Jiang* teaches the spoken utterance (i.e., speech) is encoded as one or more feature vectors reflecting the spectral characteristics of the speech. *Jiang*, 1:58-66, 6:62-7:10. To determine the entirety of the uttered speech unit (whether it be a subword unit, a word, or a sequence of words), a most likely *state sequence* is computed. In general, discussing utilizing HMMs, *Jiang* refers to a “probability score” being “computed for each state” (of the HMM), and a “cumulative probability score” successively computed, utilizing the Viterbi algorithm, for each possible state sequence. *Jiang*, 2:10-15.

130. Thus, the HMM includes a plurality of states, where *each state* includes a *probability* that the feature vector(s) are modeled accurately. *Jiang* teaches the system can be trained by providing speech input that is converted to codewords. HMMs are created by training module for each phoneme. The HMMs thus represent modeled speech associated with the phonemes. Therefore, a POSITA would have understood that *Jiang* contemplates training the system to generate HMMs that are representative of one or more phonemes. I note *Jiang* teaches “training” the speech recognition system to generate model HMMs. *Jiang*, 11:29-48. Some states will

have a low probability because the HMM state is unlikely to represent the feature vector(s) representative of the spoken speech, whereas other states will have a higher probability. *Jiang*, 2:15-46 (discussing pruning state sequences that are unlikely to model the spoken speech). The higher the probability, the more likely the state is an accurate model of the feature vector(s). As the HMM is traversed, a *cumulative probability* is calculated for each successive state in the HMM. *Jiang*, 2:7-15. At the end of the utterance (and therefore, at the end of the HMM), the sequence of states (i.e., the “state sequence”) with the highest cumulative probability is the “most likely state sequence for the entire utterance.” *Jiang*, 2:10-15. This “most likely sequence” is then converted into the utterance, whether it be a spoken subword unit, word, or word sequence. *Id.*

131. A feature vector commonly corresponds to a frame of speech, where a frame of speech is commonly 10 ms in length. *Jiang*, 1:58-67, 6:62–7:1; *Ravishankar*, 28 (“Frames are typically spaced at 10msec intervals”); *Section X.B.2* (§ 125). I note that a 10 ms length of speech is short, and therefore, most phonemes (and, by default, words) in the English language will comprise a plurality of frames. As is known, there are approximately 40-50 phonemes in the English language. *Jiang*, 3:4-6; *Ravishankar*, 42 (Section 3.1.2). Because a spoken 10 ms utterance will not usually include the entirety of a phoneme, the speech recognition process frequently divides the speech recognition process into multiple stages. First, a most

likely phoneme of the approximate 40-50 phonemes is determined, and then, a most likely word formed by one or more phonemes is determined.

132. To recognize *phonemes*, the speech recognition system is trained, which creates acoustic or phonetic models, e.g., HMMs, representing the phonemes.

Per *Jiang*:

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; *Ravishankar*, 28. Once the acoustic models, such as the HMMs, are created, the system can then use the HMMs to recognize phonemes. These HMMs are then used in *Jiang's* prefix tree to identify words. Further, *Jiang* teaches:

Another **conventional technique** for further **reducing the magnitude of computation** required for speech recognition includes the use of a **prefix tree**. 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.

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.

*Jiang*, 2:47-60, 7:30–8:51; *Ravishankar*, 33 (discussing that organizing HMMs as a phonetic tree (referred to as a lexical tree) to perform word searching significantly reduces the computation).

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*. If the pronunciations of two or more words contain the same  $n$  initial phonemes, they share a single sequence of  $n$  HMM models representing that initial portion of their pronunciation. (In practice, most systems use triphones instead of just basephones, so we should really consider triphone pronunciation sequences. But the basic argument is the same.) Since the word-initial models in a non-tree structured Viterbi search are typically the majority of the total number of active models, the reduction in computation is significant.

*Ravishankar*, 33. See *Ravishankar*, 67 (Section 4.2) (same), 68-69, 22 (discussing that a linear sequence of HMM models for each word in the vocabulary is “one to two orders of magnitude beyond the capabilities of modern workstations”)).

133. Thus, a subword unit, such as a phoneme, is represented as a branch in the prefix tree. Each branch is alone associated with a phonetic model, such as one or more HMMs. Therefore, the prefix tree employs a plurality of HMMs, where each branch of the prefix tree is associated with an HMM that was used to determine the particular phoneme branch.

## 2. '377 Patent's Discussion of "distance"

134. The '377 Patent describes "distances" as "probabilities" or "likelihoods," and "probability distributions," as can be seen in Fig. 16 below:

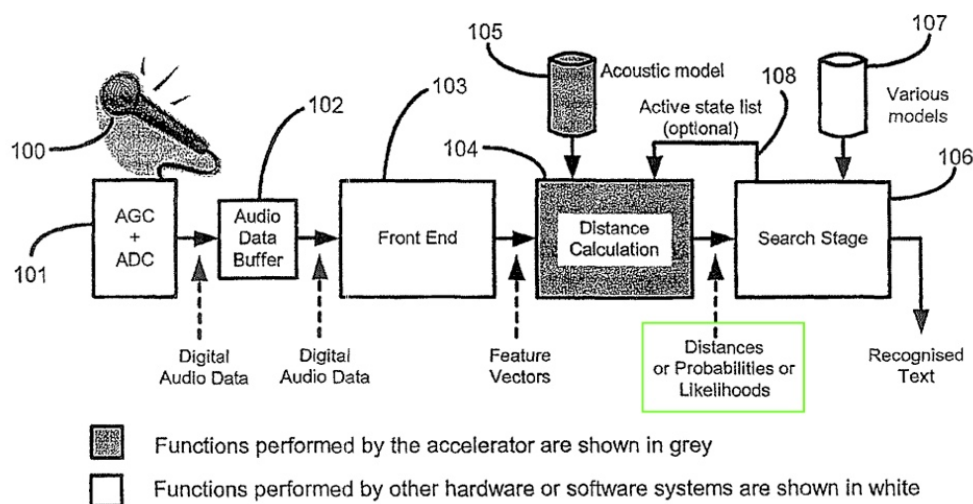


Figure 16

'377 Patent, Fig. 16 (annotated), 4:9-19 (also describing distances as representing "the likely correspondence of feature vectors to states in an acoustic model"). The '377 Patent describes "[i]n 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 in speech recognition is the calculation of

**output state probabilities in recognisers using Hidden Markov Models.” ’377**

*Patent*, 4:38-42, 4:15-17, 13:24-26 (“The distance calculator stage of the recognition process computes a probability or likelihood that a feature vector corresponds to a particular state.”). The ’377 Patent describes “a wide variety of distance metrics used” and are “widely documented in the literature.” ’377 *Patent*, 4:15-19.

135. In my opinion, the ’377 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. That is, a distance (e.g., likelihood, probability) is 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 because a low probability indicates a larger distance, and a high probability indicates a smaller distance between an expected HMM output and the input speech feature vector.

136. I note that the ’377 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.” ’377 *Patent*, 13:55-58. Additionally, the ’377 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.” *'377 Patent*, 14:14-18. In my opinion, the *'377 Patent* draws a mathematical distinction between a probability and a likelihood. However, I understand from these passages and other collective discussions in the *'377 Patent* that regardless of any mathematical distinction between a probability and a likelihood, both measurements are a similarity measurement.

**3. *'377 Patent's Teachings of “a plurality of acoustic states of an acoustic model”***

137. The *'377 Patent* describes examples of acoustic states, “including states corresponding to [a] phone or group of phones.” *'377 Patent*, 7:27-28, 8:23-24. The *'377 Patent* further describes acoustic models include HMM models and “alternative types of acoustic models, rather than HMM models.” *'377 Patent*, 38:29-31.

**4. *Jiang's Teachings***

**a) “a plurality of acoustic states of an acoustic model”**

138. In my opinion, *Jiang* teaches Hidden Markov Models (HMMs), which are the claimed “acoustic models.” *Jiang*, 7:34-38 (“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.”), 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).”).

In my opinion, *Jiang's* use of HMMs for determining the most likely phoneme (7:30-44) is consistent with the well-known use of HMMs, which are in turn well known to include states. I also note that HMMs were well known, as I discussed above and as taught by *Ravishankar*. See Section V.D; *Ravishankar*, 18-36, 41-53, 58-60. A POSITA would also have understood from *Jiang's* teachings that HMMs are phonetic speech unit models that said phonetic speech is acoustic. Indeed, such speech relates to sounds, which a POSITA would have understood as related to acoustics. Further, the '377 Patent compares HMM models and "alternative types of acoustic models," which indicates to a POSITA that HMMs are one example of an acoustic model. '377 Patent, 38:29-31.

139. I also note that *Jiang* teaches that HMMs have the claimed "plurality of states." *Jiang*, 1:21-24 ("A hidden Markov model includes 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."), 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). Further, *Ravishankar* teaches HMMs include a plurality of

states. “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.” *Ravishankar*, 28. 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.

**b) “a feature vector”**

140. It is also my opinion that *Jiang* teaches encoding the frames of spoken speech into a feature vector. 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. I also note that *Jiang* teaches an optional additional step involving vector quantization techniques. *Jiang*, 7:1-10. In my opinion, vector quantization is a well-known technique that clusters feature vectors into similar codewords and represents cepstral features in an efficient manner to reduce the size of the search space and reduce computation requirements. 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.

141. *Ravishankar* also discusses vector quantization, further informing my understanding of *Jiang. Ravishankar*, 48-49 (“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 the ’377 Patent is consistent with the known understanding of codewords as an index into a codebook, and where a codebook stores a value representative of a cluster of data in the original form such as a representative feature vector. *See* ’377 Patent, 35:36-37, 5:6-8; *see also* *Maes* (discussed at ¶ 66, above). 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.

**c) “calculat[ing] distances ...”**

142. Regarding the claimed calculating distances, *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

acoustic states of at least one HMM. *Jiang*, 8:29-33, 7:30-47. 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 acoustic states of an acoustic model, as I discuss in ¶ 145, below. 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 is 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 “organizing 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. *Jiang*, 7:30-47. 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. *Jiang*, 7:49-54, 8:5-43 (including disclosing lexicon 70).

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

146. 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*, 6:64-7:11, 1:58-67, 7:38-43, 8:29-36. I note that because *Jiang* teaches determining the likelihood a given phoneme is indicated by a codeword also determines the likelihood the given

phoneme is indicated by at least one feature vector. *See* 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.

147. 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*, 8:28-31. 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 comprises a distance calculation.

148. It is also my opinion that *Jiang* teaches or at least renders obvious calculating a plurality of distances, including calculating a score for each respective node. *Jiang*, 2:5-7. *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, 8:5-43. Thus, a score is computed for each node where the score indicates the likelihood that that phoneme for each node is the phoneme encoded by a codeword. *Jiang*, 8:16-32. *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. *Jiang* discloses the Fig. 3 prefix tree begins with the phonemes represented by the letters AO, AE, and T. *Id.* Because *Jiang* calculates at least three scores for respective phonemes AO, AE, and T, *Jiang* teaches calculating distances (plural). Thus, 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 is calculated. Indeed, a POSITA would have understood that because the prefix tree has a plurality of nodes, multiple scores or distances are calculated.



“classification processor 34” refers to the processor of classifier 34, which is “classifying processor 341” or “classifier processor.” That is, classifier processor 341 performs the classification functions of classifier 34 and is the processor of classifier 34. Thus, in my opinion, a POSITA would have understood that “classification processor 34” and classifier processor 341 are used interchangeably and have no difference in meaning. Merely for ease of reading, I refer exclusively to “classifier processor 341” in my analysis below when discussing *Smyth*’s interchangeable terms “classifying processor 341,” “classification processor 34,” and “classification processor.”

**a) “a second programmable device programmed to”**

150. Regarding *Smyth*’s disclosure of DSPs, in my opinion, *Smyth* teaches utilizing a first DSP for the frame generator 32 and feature extractor 33, a second DSP for the classifier processor 341 of classifier 34, and a third DSP (or microprocessor) for the parsing processor 351 of sequencer 35. I note the following teachings from *Smyth*:

(1) “The frame generator 32 and feature extractor 33 are, in this embodiment, provided by a single suitably programmed digital signal processor (DSP) device (such as the Motorola DSP 56000, or the Texas Instruments TMC C 320) or similar device.” *Smyth*, 5:49-53.

(2) “The classifier processor 341 may be a suitably programmed digital signal processing (DSP) device, may in particular be the same digital signal processing device as the feature extractor 33.” *Smyth*, 6:5-8.

(3) “The parsing processor 351 may, for example, be a microprocessor such as the Intel<sup>(TM)</sup> i-486<sup>(TM)</sup> microprocessor or the Motorola<sup>(TM)</sup> 68000 microprocessor, or may alternatively be a DSP device (for example, the same DSP device as is employed for any of the preceding processors).” *Smyth*, 6:47-51.

151. In my opinion, *Smyth*’s teaching at 5:49-53 is informing the reader of suitable exemplary DSP models, such as the Motorola DSP56000, for the DSP for the frame generator and extractor. *Smyth* then discusses the DSP for the classifier, where the DSP is also a suitably programmed DSP and may be the same DSP device as the feature extractor’s DSP. Here, rather than repeating suitable model types (e.g., Motorola DSP 56000), *Smyth* is simply informing the reader the DSP for the classifier may be the same model of DSP device as for the frame generator and extractor.

152. This understanding is further elucidated by *Smyth*’s teaching at 6:47-50, discussing the processing element for the parsing processor. The parsing processor is included in the sequencer 35, per *Smyth*, 6:10-12, 4:14-18 (disclosing “the processor of the sequence parser of FIG. 4[.]”). At *Smyth*, 6:47-50, *Smyth*

discusses the parsing processor may either be a microprocessor or alternatively a DSP device. In disclosing the parsing processor may be a microprocessor, *Smyth* again lists exemplary models (e.g., Intel™ i-486™ and Motorola™ 68000). However, *Smyth* does not provide exemplary models for the DSP device, instead simply saying the parsing processor “may alternatively be a DSP device (for example, the same DSP device as is employed for any of the preceding processors).” *Smyth*, 6:49-51. *Smyth* chooses not to restate the exemplary types of DSP devices, such as the Motorola DSP 56000 and TI TMCC320, but does state the exemplary type of microprocessor (because it is the first time in the written description a microprocessor is referenced). Therefore, it is my understanding *Smyth* is teaching that three distinct DSP devices of the same exemplary model types may be used, one DSP for *each* of the feature extractor (and frame generator), the classifier processor, and the parsing processor.

153. I also note *Smyth*, 6:50-51 states “the same DSP device as is employed for any of the preceding processors.” Here, the reference to “any of the preceding processors” is referring to any of the previously-discussed processors, which is (1) the “single suitably programmed digital signal processor (DSP) device for the frame generator and feature extractor (*Smyth*, 5:49-53); and (2) the classifier processor 341 of classifier 34, which also “may be a suitably programmed digital signal processing (DSP) device” (*Smyth*, 6:5-8). If the same physical DSP device was used for all three

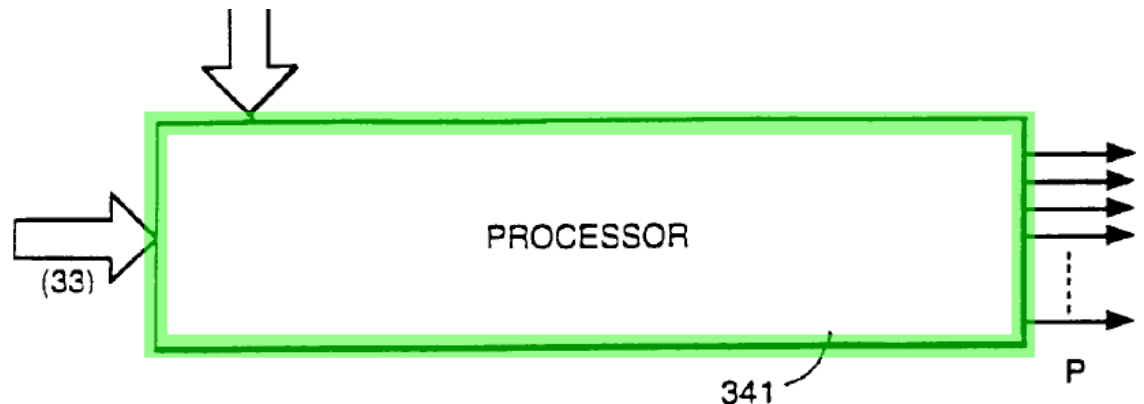
processors—rather than three distinct processors each of the same DSP model type—there would have been no reason to refer to “any of the preceding processors” (plural). That is, because each processor is a suitably programmed DSP (*Smyth*, 5:50-51, 6:5-6), and because *Smyth* references plural preceding processors, then this indicates to me (in addition to the other reasons I provide herein), that *Smyth* is teaching that each processor is a distinct DSP. Moreover, the reference to “preceding processors” would have been understood to refer to a digital signal *processor* (DSP) device, per *Smyth*, 5:49-53.

154. As further support for my understanding, *Smyth* expressly teaches that the frame generator 32 and feature extractor 33 are “provided by a single suitably programmed” DSP. Indeed, *Smyth* discloses a “front-end processor” that was “common in speech recognition processing” that performs the same functions as frame generator 32 and feature extractor 33 in the background disclosures. *Smyth*, 1:24-34. Therefore, in an embodiment where *Smyth* does want multiple processing elements on a single DSP, it specifically references providing the elements on a “single” DSP. However, for the DSPs for the classifier processor 341 and parsing processor 351, *Smyth* does not reference providing the processing components on a single DSP and instead refers to “preceding processors.”

155. Further, I also note that a skilled person would have understood that in at least some embodiments, the functionalities of frame generator 32 and feature

extractor 33 are implemented by a single DSP. *Smyth* teaches that frame generator 32 and feature extractor 33 are provided by a “**single**” DSP “in this embodiment,” which a POSITA would have understood *Smyth* teaching a single processor for performing the functions of frame generator 32 and feature extractor 33. *See also Smyth*, 1:24-32 (disclosing common to have a “front-end processor” performing the functions of frame generator 32 and feature extractor 33, further indicating a single processor for frame generation and feature extraction is contemplated); Fig. 2 (“a block diagram showing schematically the functional elements of a recognition processor according to an embodiment of the invention,” *Smyth*, 4:4-6). In my opinion, *Smyth* thus differentiates between embodiments utilizing a single DSP for certain functions and embodiments utilizing distinct DSPs for the same functions. When *Smyth* contemplates functions performed by a single DSP, it says such explicitly by referencing a *single* DSP. Therefore, in my opinion, *Smyth* teaches an embodiment in which a first DSP is utilized for the frame generator 32 and feature extractor 33, a second DSP is utilized for the classifier processor 341 of classifier 34, and a third DSP (or microprocessor) is utilized for the parsing processor 351 of sequencer 35. *See also Smyth*, 6:36-42 (teaching parsing processor 351 reads probabilities output by classifier processor 341) and Figs. 3-4 (illustrating both processors 341 and 351 having separate element numbers and having their own respective memories 342 and 352 (and state probability store 353), respectively).

156. *Smyth* discloses classifier processor 341 “may be a suitably **programmed** digital signal processing (DSP) device[.]” *Id.* As I discussed above, it was well known in the art before the priority date of the ’377 Patent that DSPs were **programmable** devices and that the Motorola DSP56000 specifically mentioned by *Smyth* was a well-known, widely available programmable DSP. *Smyth*, 5:49-53; see V.G (discussing *Kloker* and *Newton*). Therefore, in my opinion, *Smyth* teaches classifier processor 341, which is implemented as a second DSP, is a second “programmable device.” Classifier processor 341 (annotated green) is depicted in the excerpt of Fig. 3 below.



*Smyth*, Fig. 3 (annotated, excerpt).

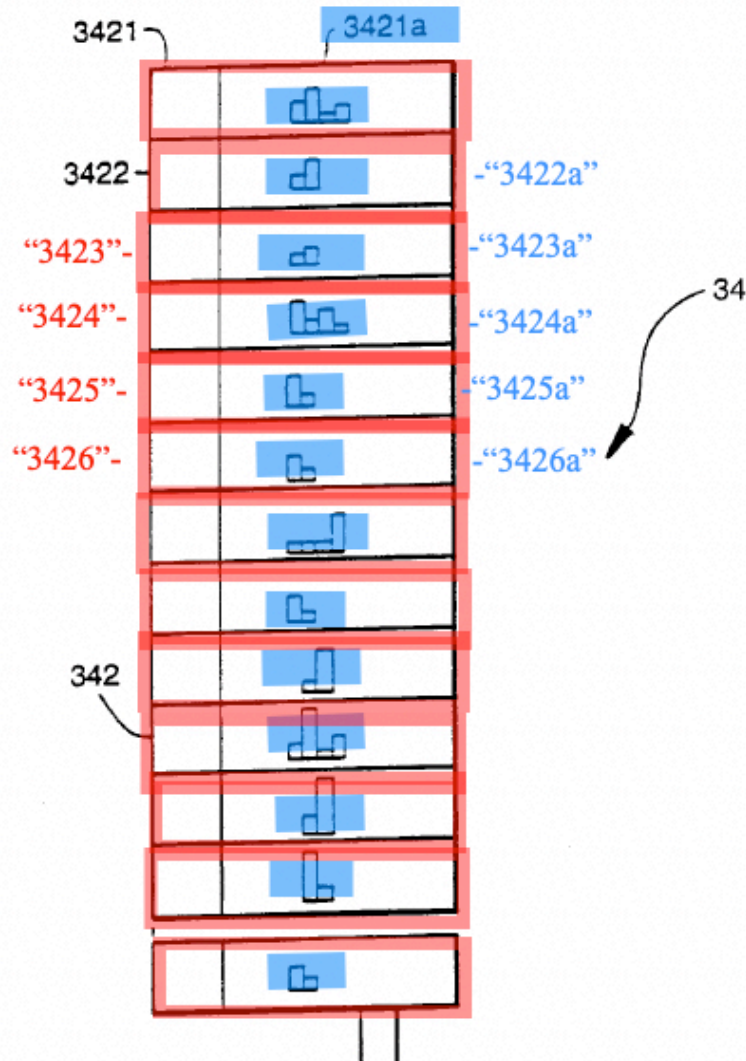
**b) “a plurality of acoustic states”**

157. In my opinion, *Smyth* teaches a state memory 342 that includes state fields (e.g., 3421, 3422, “3423,” ...) each with a corresponding “speech state” (e.g., 3421a, “3422a,” “3423a,” ...) as depicted in annotated Fig. 3 below. Specifically, state fields 3421, 3422, “3423,” “3424,” “3425,” “3426,” ... (annotated red) and

each respective speech state 3421a, “3422a,” “3423a,” “3424a,” “3425a,” “3426a,”

... (annotated blue) of said state fields are depicted in Fig. 3 below.<sup>3</sup>

Fig. 3



<sup>3</sup> State fields and speech states listed in quotation marks are annotations added for clarity.

*Smyth*, Fig. 3 (excerpt, annotated). I note that although *Smyth* does not individually label every state field and speech state, a POSITA would have understood that each state field has a corresponding speech state, such as state field 3421 having a corresponding speech state 3421a.

158. I note that *Smyth* does not expressly refer to element 3421a, “3422a,” etc., as “speech states.” However, a POSITA would have understood such elements to be speech states for several reasons. *Smyth* teaches the “state memory 342 comprises a state field 3421, 3422 ..., for each of the plurality of speech states. For example, 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*, 5:58-62, 6:1-4 (“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.”). In my opinion, *Smyth*’s use of ellipses indicates to a skilled artisan that *Smyth* teaches similar state fields each with respective speech states. Thus, in my opinion, a POSITA would have understood that element 3421a refers to speech state 3421a and the annotated elements “3422a” through “3426a” are also speech states of their respective state fields (e.g., state fields 3422 through “3426”).

159. In my opinion, *Smyth*’s speech states are **acoustic** states, as claimed. First, a POSITA would have understood that speech relates to the sounds of

utterances of speakers, and acoustics refers to sounds. *Schmandt* teaches “acoustics” of speech processing is “[t]he realization of the string of phonemes in the sentence as vibrations of air molecules to produce pressure waves, i.e., sound.” *Schmandt*, 18. Therefore, because *Smyth*’s speech states relate to sounds of speech, a POSITA would have understood that such speech states are “acoustic states.”

160. Additionally, a POSITA would have understood that *Smyth*’s speech states meet the claimed “acoustic states” according to the ’377 Patent’s description. As I discuss in detail below, in my opinion, because allophones correspond to phonemes and “phoneme” and “phone” are synonyms, *Smyth*’s speech states, which relate to allophones, thus correspond to a “phone or group of phones” and are therefore “acoustic states,” as claimed. The ’377 Patent describes acoustic states as a “phone or group of phones.” ’377 Patent, 7:25-28 (“a content addressable memory for storing content addressable data for each phone or group of phones, including states corresponding to said phone or group of phones[.]”), ’377 Patent, 8:23-24.

161. *Smyth* expressly teaches that its speech states are related to allophones, which a POSITA would have understood are units of speech. *Smyth* teaches “[t]he state memory 342 comprises a state field 3421, 3422 ..., for each of the plurality of speech states. For example, 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*, 5:58-62, 6:10-26, 6:52-60.

162. In more detail regarding allophones, *Schmandt* teaches that “[e]ach language employs a characteristic set of phonemes or allophones to convey words. These speech sounds are generated by manipulating the components of the vocal tract into specific physical configurations. The set of sounds comprising the phonemes of a language must be distinctive because the listener perceives only the sound, not the vocal tract configuration that produces it.” *Schmandt*, 33-34. *Schmandt* also teaches that allophones are related to phonemes: “An **allophone** is one of a number of different ways of pronouncing the same phoneme. Replacing one allophone of a phoneme with another does not change the meaning of a sentence, although the speaker will sound unnatural, stilted, or like a non-native.” *Schmandt*, 24. *Schmandt* further teaches that any set of phonemes will have a set of allophones and that phone and phonemes are synonymous for most practical purposes: “An individual speech sound is a **phone**, or **segment**. For any particular language, a given phoneme will have a set of allophones, each of which is a segment. Segments are properties of the human vocal mechanism, and phonemes are properties of languages. For most practical purposes, phone and phoneme may be considered to be synonyms.” *Id.* Therefore, in my opinion, because allophones correspond to phonemes and “phoneme” and “phone” are synonyms in this case, *Smyth’s* speech states are “acoustic states” as claimed according to the description of the ’377 Patent because *Smyth’s* speech states correspond to a “phone or group of phones.”

**c) “[plurality of speech states] of an acoustic model”**

163. It is also my opinion that *Smyth* teaches an acoustic model, as claimed. I note that a POSITA would have understood a set of speech states such as those taught by *Smyth* would have been an acoustic model. *Smyth* teaches 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, 5:6-9 (“[C]lassifier 34 receiv[es] the succession of feature vectors and operat[es] on each with a plurality of model states, to generate recognitions results[.]”). A POSITA would have understood that it is 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

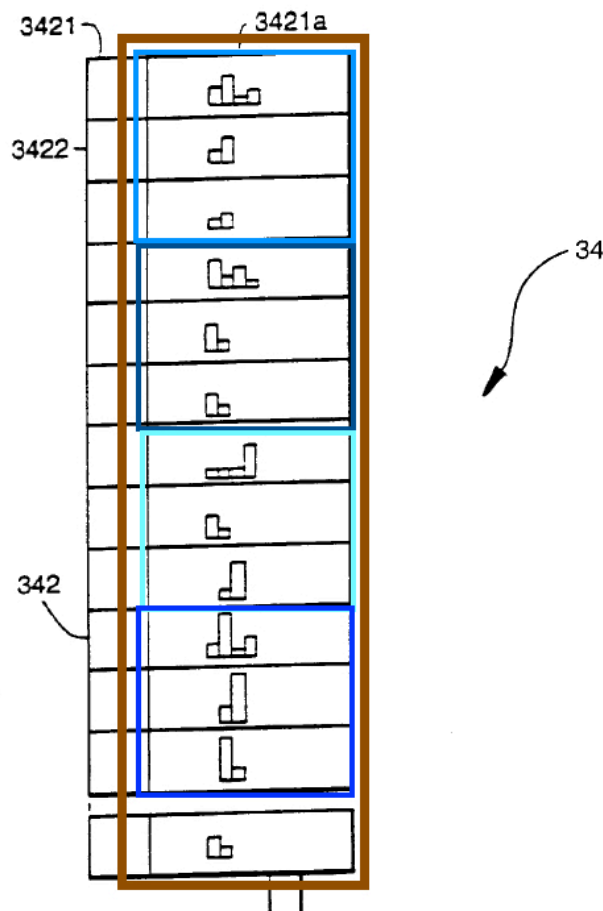
acoustic models in subword units, such as phonemes)). A POSITA would have understood *Smyth*'s "plurality of model states" is an *acoustic model*, where reference to the states of the model (i.e., *Smyth*'s speech states at 5:58-62) is used to determine the likelihood that a feature vector corresponds to one of the speech states. For example, *Schmandt* teaches 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. *Schmandt* also teaches 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.

164. As I discussed above, a POSITA would have understood that *Smyth*'s speech states are acoustic states, as claimed. Indeed, *Smyth*'s speech states relate to allophones, which correspond to phonemes and which convey words. *Schmandt*, 33 ("Each language employs a characteristic set of phonemes or allophones to convey words."). *Smyth* teaches that state memory 342, which stores a plurality of speech states, has 3 state fields provided for each allophone. *Smyth*, 5:58-62 ("each allophone to be recognized by the recognition processor comprises 3 states, and accordingly 3 state fields are provided in the state memory 342 for each allophone."). In my opinion, *Smyth*'s collection of speech states in state memory 342 (wherein groups of three speech states are provided for each allophone) is an acoustic model,

as claimed. *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. *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. *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)<sup>4</sup>. Three speech states are provided for each allophone. Therefore, *Smyth*'s groups of speech states teach or render obvious the claimed "acoustic model."

165. Additionally, I note that prior art speech models utilizing allophones were well known, with *Peters* as one example. *Peters* teaches a "reference speaker independent model set database comprises 800 allophone speech models[.]" *Peters*, 9:39-40. Further, *Peters* teaches that speech models utilizing allophone speech models were "available as off-the-shelf components and are well known in the art[.]" *Peters*, 9:45-48.

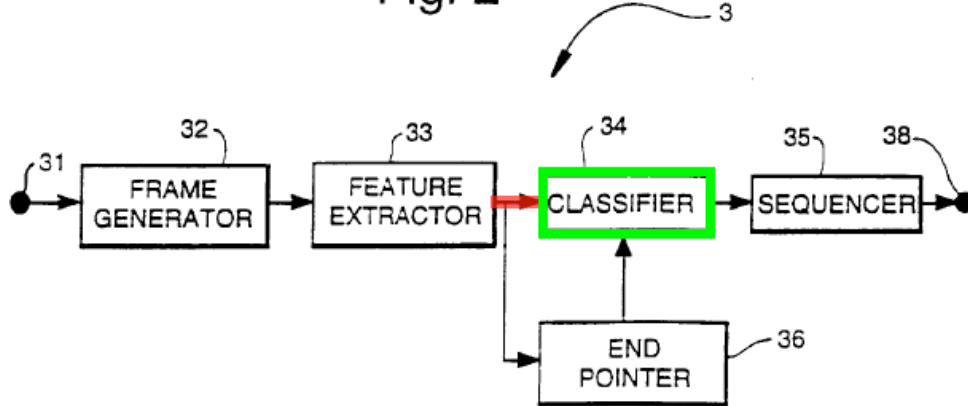
**d) "[programmed to] calculate distances indicating the similarity ..."**

166. As I discussed above, *Smyth* teaches the claimed "a first programmable device programmed to calculate a feature vector from a digital audio stream." See Claim 1(a). It is also my opinion that *Smyth* teaches a feature vector (annotated red) is output from feature extractor 33 and is received at classifier 34 (annotated green).

---

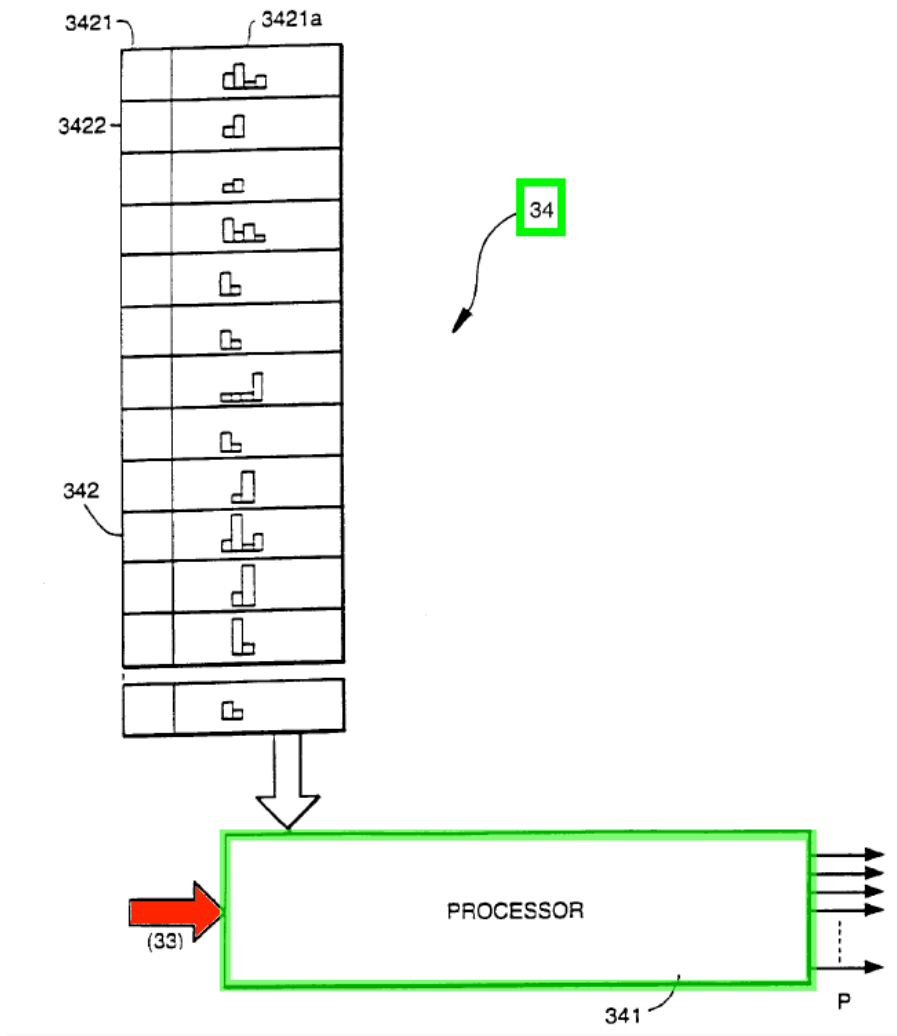
<sup>4</sup> I reproduced this annotated Fig. 3 from the Petition.

Fig. 2



*Smyth*, Fig. 2 (annotated). In particular, as depicted in annotated Fig. 3 below, processor 341 (annotated green) of classifier 34 receives the output of feature extractor 33 (annotated red).

Fig. 3



*Smyth*, Fig. 3 (annotated).

167. Therefore, in my opinion, *Smyth* teaches classifier processor 341 receives a feature vector. It is also my opinion that *Smyth* teaches classifier processor 341 performs the claimed distance calculation “indicating the similarity between a feature vector and a plurality of acoustic states of an acoustic model.” In particular,

*Smyth's* classifier processor 341 outputs a probability (i.e., distance) that the input feature vector corresponds to each state.

The classification processor 34 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”). Compare *Smyth*, 5:63-6:4 with *'377 Patent*, 4:29-40 (“[A] distance is an indication of the probability or likelihood that a feature vector corresponds to a particular state.”), 13:24-26 (“The distance calculator stage of the recognition process computes a probability or likelihood that a feature vector corresponds to a particular state.”), 13:26-30 (“Each state has an associated probability distribution in the N-dimensional space. The distance calculator computes the distance in the N-dimensional space from the Feature Vector to the probability distribution for each state.”).

168. In my opinion, because the '377 Patent describes distances as an indication of probability and because *Smyth* teaches outputting a probability for each state, a POSITA would have understood that *Smyth* teaches the claimed “calculate distances.” Indeed, *Smyth* teaches that the plurality of state probabilities that are outputted indicates a likelihood (similarity) that the inputted feature vector corresponds to a particular state. *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”). Thus, *Smyth* teaches the claimed “indicating the similarity between a feature vector and a plurality of states.” I note that *Smyth*’s teachings are similar to the '377 Patent’s descriptions. In my opinion, *Smyth* teaches calculating the claimed distances (i.e., probabilities) between a feature vector and individual speech states of state memory 342. Further, in my opinion, *Smyth* teaches calculated the claimed “distances indicating the similarity between a feature vector and a plurality of acoustic states of an acoustic model.” For example, *Smyth* teaches a “classifier 34 receiving the succession of feature vectors and operating on each with a plurality of model states.” *Smyth*, 5:7-9, *Smyth*, 5:63-6:4 (reproduced above). See Claim 1(c)(2).

169. Also, as I discussed above, it was well known in the art that DSPs were programmable devices. See Section V.G (discussing *Kloker* and *Newton*); Claim

1(c)(1). In my opinion, because classifier processor 341 is “a suitably programmed digital signal processing (DSP) device,” a POSITA would have understood that *Smyth*’s classifier processor 341 is programmed to calculate the claimed distances (i.e., probabilities). *Smyth*, 6:5-6.

**6. *Motivation to Modify Jiang to Include Smyth’s Second Programmable Device***

170. In my opinion, a skilled artisan would have been motivated and found it obvious to modify *Jiang*’s tree search engine 74 to either include a second processor/DSP in view of *Smyth*’s teachings or to perform *Jiang*’s distance calculation using a processor/DSP that is distinct from tree search engine 74. *Jiang* teaches the tree search engine 74 is programmed to perform the claimed calculating functionality recited in Claim 1(c), i.e., calculating the distance, as discussed in Section X.C.4, above. *Jiang*, 7:38-43. Per *Jiang*, the tree search engine 74 may be a dedicated speech recognition processor. *Jiang*, 6:39-42. As established below in Section X.E, *Jiang*’s tree search engine 74 also is programmed to identify spoken words, as recited in Claim 1(e). A POSITA would have found it obvious and been motivated to modify *Jiang* such that the distance calculation was performed with a processor distinct from the tree search engine’s processor or to otherwise modify tree search engine 74 to comprise two programmable devices instead of the processor already taught in *Jiang*.

171. I note that *Jiang* itself discloses various system configurations and expressly discloses CPU 21 “may include one or more processors.” *Jiang*, 6:39-42. *Jiang* teaches tree search engine 74 “is preferably implemented in CPU 21.” *Id.* Further, *Jiang* teaches several distinct processors that perform various speech recognition tasks. For example, *Jiang* teaches “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.” *Jiang*, 6:53-57. Additionally, as I noted above, *Jiang* teaches a silence detection module 68 that “can either be implemented on the same, or a different processor as that used to implement the feature extraction module 66. Silence detection module 68 operates in a well known manner. Briefly, silence detection module processes the digital samples[.]” *Jiang*, 7:19-24. In another example, *Jiang* teaches a “training module 65 can be implemented on either the same processor, or a separate processor, from the remainder of the speech recognition system.” *Jiang*, 11:34-37. Further, *Jiang* teaches numerous system configurations. *Jiang*, 4:60-65 (including multiprocessor systems, microprocessor-based electronics, PCs, computers).

172. I also note that it was well known in the art that calculating probabilities/distances was a computationally-intensive processing operation. For example, *Jiang* teaches that “for a large vocabulary, the number of states and transitions becomes large and the computation required to update the probability

score at each state in each frame for all possible state sequences takes many times longer than the duration of one frame, which is typically approximately 10 milliseconds in duration.” *Jiang*, 2:19-24.

173. In another example, *Baumgartner* also teaches that such calculation is computationally intensive. “Speech recognition, in which speech is translated to text, is currently a computationally intensive task.” *Baumgartner*, [0003], [0005] (“While it is possible to purchase larger and faster systems in order to help translate this amount of speech, it would be beneficial to reduce the load on the processors of the system when the system performs speech recognition. The latter would allow the system to use the extra processing power on other tasks or to accommodate additional simultaneous speakers.”), [0007] (“The present invention provides Speech Label Accelerators (SLAs) and techniques for using such SLAs. The present invention speeds processing during the labeling stage of a speech recognizer and frees one or more processors from performing labeling.”).

174. I note that *Jiang* teaches performing a pruning process that “eliminates the need to compute the probability score for state sequences that are very unlikely.” *Jiang*, 2:25-29. However, while pruning decreases the computational load for speech recognition, it also has the potential downside of potentially lowering the accuracy of the speech recognition results. Per *Jiang*:

Thus, a technique called pruning, or beam searching, has been developed to greatly reduce computation needed to determine the most likely state sequence. This type of technique eliminates the need to compute the probability score for state sequences that are very unlikely. This is typically accomplished by comparing, at each frame, the probability score for each remaining state sequence (or potential sequence) under consideration with the largest score associated with that frame. If the probability score of a state for a particular potential sequence is sufficiently low (when compared to the maximum computed probability score for the other potential sequences at that point in time) the pruning algorithm assumes that it will be unlikely that such a low scoring state sequence will be part of the completed, most likely state sequence. The comparison is typically accomplished using a minimum threshold value. Potential state sequences having a score that falls below the minimum threshold value are removed from the searching process. The threshold value can be set at any desired level, based primarily on desired memory and computational savings, and a desired error rate increase caused by memory and computational savings.

*Jiang*, 2:25-46.

175. *Jiang* utilizes a prefix tree, which in my opinion is a lexical tree, to address potential issues with pruning. *Jiang*, 2:47-51. As I noted above, *Ravishankar* teaches using a lexical tree was a well-known improvement for speech recognition systems. *Ravishankar*, 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 a *tree-structured lexicon* or *lexical tree*.”). However, *Jiang* still teaches that utilizing a lexical tree does not fully address the issue of large computational loads and lowered accuracy because pruning requires setting a threshold level and has either a higher or lower number of hypotheses based on the pruning threshold setting. *Jiang* recognizes that there are a plurality of “hypotheses” in the HMM representing various state sequences. *Jiang*, 11:8-24, 9:8-10. I note that pruning gets rid of some of the state sequences, i.e., hypotheses. However, a POSITA would have understood that it is also desirable to maintain a certain number of hypotheses to determine word for word sequences. *Id.* Per *Jiang*:

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.

Of course, a decision need not be made at the first level in the tree. Also, any particular threshold level can be employed in the search strategy. The lower the threshold level, the more hypotheses will be retained throughout the search, and thus the more accurate the recognition system will be. However, the threshold level is preferably determined empirically so as to gain an increase in computational savings, while

significantly reducing the error rate associated with the pruning technique.

*Jiang*, 11:8-24

176. I note that with pruning, one or more branches of the lexical tree are removed from consideration. However, at any node in the lexical (prefix) tree, when a branch is removed, there may be other branch(es) at lower levels of the tree that are saved for future consideration. This is because as more frames of speech input are received, the additional received frames further inform the likelihood of a particular phoneme. Thus, with subsequence speech frames, it may turn out that one of the lower branches is a better match to the input feature vectors. Therefore, *Jiang's* threshold level sets how many hypotheses are carried forward for evaluation for the next frame(s) and corresponding feature vector(s). *Jiang*, 11:21-24 teaches that "the threshold level is preferably determined empirically so as to gain an increase in computational savings, while significantly reducing the error rate associated with the pruning technique." *Jiang* therefore recognizes that the threshold level affects the recognition accuracy.

177. Further, I note that if the speech recognition system is programmed to carry more hypotheses forward in the tree (because a relatively lower threshold value is set), the system has higher recognition accuracy. But a POSITA would have understood there is a trade-off. If the computation per frame exceeds the frame rate

(e.g., 10 ms), the system is calculating distances increasingly behind real time. The system can, however, get time “back” with a second processor dedicated to distance calculation. With *Jiang*, the logical division is then that the tree search engine 74, which is described as performing both probability scoring and word identification, is divided into two processors, e.g., two DSPs. The part of the computation that involves computing the probability / score for each HMM versus the input feature vector is mathematically expensive (a lot of dimensions to the feature vector, etc.) and so would be performed most efficiently on a DSP. This would free up the tree search portion, running on another processor, to have more time to consider alternate hypotheses, and obtain better recognition results.

178. In my opinion, it would have been obvious to modify *Jiang*’s tree search engine 74 to comprise two dedicated processors (such as two digital signal processors (DSPs)) where one DSP/processor performs the computationally-intensive distance calculations and the other DSP/processor performs other speech recognition tasks. Such a modification would have achieved better accuracy and speed for the overall speech recognition process. Indeed, such a modification would have allowed the tree search portion running on a separate processor to have more time to consider alternate hypotheses and thus would have advantageously obtained better recognition results. As I noted above, there is a tradeoff between efficiency and accuracy when utilizing pruning. This modification would have allowed a lower

pruning threshold to be used which would have allowed for more hypotheses to be carried forward. By utilizing a second DSP/processor for performing distance calculations, the tree search process carried out on a first DSP/processor would have had more time to consider said additional hypotheses, thus increasing accuracy.

179. In my opinion, *Jiang* suggests to a skilled artisan that various system configurations were amenable to design and use considerations, such as utilizing various software and hardware configurations or architectures to achieve desired functionality. *See Jiang*, 4:60-5:3 (discussed above). Thus, a POSITA would have found it obvious to modify *Jiang* to utilize distinct DSPs/processors (such as *Smyth's* DSPs) to perform various speech recognition steps. I note that *Jiang* itself discloses such. *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.”). A POSITA also would have understood that DSPs are efficient and advantageous for performing distance calculations and generating feature vectors. A DSP performing feature vector generation is busy taking every speech sample, generating frames, and performing computationally intensive feature vector calculations, so it would not be advantageous to further burden said DSP with additional tasks in parallel.

180. Further, utilizing distinct processors/DSPs results in a more efficient, faster, and more accurate speech recognition process by segmenting higher-load

processing tasks. I note that *Ravishankar* teaches that such is a goal of speech recognition. *Ravishankar*, 19. It would have been obvious to modify *Jiang*'s tree search engine, which performs both distance calculation and word determination, to include *Smyth*'s distinct processors for performing each respective functionality. *Smyth*, 5:49-53, 6:5-8, 6:43-51. It is my opinion that even though modifying *Jiang* to add a separate DSP/processor (thus modified to have three DSPs or processors instead of only processors 66 and 74) would have arguably added some additional cost, *Jiang*'s teachings discussing a wide array of system configurations would have indicated to a skilled artisan that *Jiang* is not only concerned with cost. Indeed, *Jiang* contemplates configurations such as multiprocessor systems, minicomputers, and even mainframe computers. *Jiang*, 4:60-5:3. Additionally, it is my opinion that even though modifying *Jiang* to have a third processor may have incurred additional cost, that such cost would have been minimal in light of the advantageous gains in system speed, accuracy, and efficiency. Further, although modifying *Jiang* to include three processors instead of two processors (66 and 74) arguably adds cost for the third processor, *Jiang*'s own teachings discussing the multiplicity of possible computer configuration indicates to a POSITA that *Jiang* is not concerned with cost alone. Moreover, the cost for a separate DSP or processor for performing distance calculation would be minimal relative to the efficiency and accuracy gain.

181. In the *Jiang-Smyth* combination, *Smyth's* classifier processor 341 is added to *Jiang's* system to perform *Jiang's* distance calculations. In my opinion, as *Jiang's* prefix tree is traversed, probability scores (i.e., calculated distances) calculated by the additional processor (i.e., classifier processor 341) are received by tree search engine 74 (to the extent it is implemented as a single processor), such that tree search engine does not have to perform both distance calculations and word identification (via the tree search). In the modified *Jiang's* system, the tree search engine 74 would thus receive computed probability scores for each node associated with a phoneme branch and would therefore need to perform only word identification.

182. I noted above that *Smyth* refers expressly to implementing a DSP such as the Motorola DSP56000. As I also discussed above, *Kloker* (Ex. 1009) is a specification and feature sheet from Motorola disclosing details of the Motorola DSP56000 including expressly stated benefits, such as using DSPs for speech recognition. Further, I note that *Mariani* (Ex. 1013) teaches "DSP chips have allowed real time digital processing of speech signals, with various transforms, thus bringing consistent analysis, flexibility and higher integration." *Mariani*, 9. Therefore, in my opinion, a skilled artisan would have understood that DSPs were commonly known as being advantageous for speech recognition and that it was commonplace for DSPs to be utilized for speech recognition. In my opinion,

modifying *Jiang* in light of *Smyth's* teachings would have had a reasonable expectation of success at least because DSPs were known and routinely employed in speech recognition systems for performing speech recognition tasks. Specifically, DSPs, such as the Motorola DSP56000 expressly disclosed by *Smyth*, would have allowed for “consistent analysis, flexibility and higher integration” as taught by *Mariani*. Additionally, I note that incorporating multiple DSPs into the modified *Jiang's* speech recognition system for performing specific speech recognition functions would have predictably yielded flexibility in the system design and integration with other devices, with one example being *Jiang's* consumer electronics. As I discussed above, *Mariani* expressly discloses DSPs bring flexibility. Further, a POSITA would have understood and found obvious that, as taught by *Jiang*, DSPs are well suited to perform mathematical computations that are necessary for computing probabilities/distances/scores that indicate that a particular HMM produced a given observed feature vector.

**7. *Motivation to Modify Jiang to Include Smyth's Second Programmable Device Performing Distance Calculations***

183. In my opinion, to the extent Patent Owner argues that *Jiang* does not teach distance calculations, a POSITA would have been motivated and found it obvious to modify *Jiang* to replace *Jiang's* node scoring to determine a likelihood of similarity with *Smyth's* probability calculation of an input feature vector corresponding to a corresponding state. *Jiang*, 8:29-33; *Smyth*, 5:63-67. I note that

*Jiang* itself teaches that its speech recognition is performed “in a known manner” (*Jiang*, 8:51), which would have indicated to a skilled artisan that a well-known distance calculation would have been readily implemented. I also note that *Jiang* teaches determining a most likely phoneme by utilizing HMMs, which, in my opinion, would have further provided a background architecture for performing distance calculations and that such distance calculations would have had a reasonable expectation of success. Further, performing such a distance calculation for determining the most likely phoneme would have been using a known technique of calculating distances by comparing a feature vector to states of an acoustic model to improve *Jiang*’s speech recognition system in the same way as *Smyth*’s. In the modified *Jiang*, it would have been obvious to a skilled artisan to utilize *Smyth*’s probability calculations involving comparing an input feature vector to a state instead of *Jiang*’s “most likely phoneme” based on stored HMMs. In my opinion, such a substitution of one known element for another would have yielded predictable results. Further, such a simple substitution would have been readily implemented by a skilled artisan because distance calculations involving HMM states were well known, well understood, and commonly used. One example is *Ravishankar*, 28-29. *See also* Section V.E above where I opined on distance calculations in the Overview of the Technology.

**D. Claim 1(d): “wherein said feature vector is received by the second programmable device after it is calculated by the first programmable device; and”**

**1. Jiang’s Teachings**

184. In my opinion, *Jiang* teaches tree search engine 74 accesses information stored in the phonetic speech unit model memory 72 after receiving codewords from feature extraction module 66. Per *Jiang*:

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*, 7:30-43. Specifically, I note that *Jiang* teaches that the tree search engine 74 accesses stored HMMs “[u]pon receiving the codewords from feature extraction module 66” and “[b]ased 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[.]” *Id.* Thus, a POSITA would have understood from *Jiang’s* teachings that tree search engine 74 would have received the one or more

codewords (i.e., feature vector(s)) after feature extraction module 66 calculates said codeword(s).

## 2. *Smyth's Teachings*

185. As I discussed above for Claim 1(c), it is my opinion that *Smyth* teaches a feature vector is output from feature extractor 33 and is received at classifier 34. *See* Claim 1(c). Indeed, *Smyth* teaches classifier processor 341 calculates “the probability that the input feature set or vector corresponds to the corresponding state.” *Smyth*, 5:63-67. In my opinion, such feature vector is calculated by feature extractor 33 before being received at classifier processor 341. In my opinion, when modifying *Jiang* according to *Smyth's* teachings of a second programmable device for calculating distances, distances would have been received from *Jiang's* feature extraction module 66 after extracting the one or more feature vectors.

### E. **Claim 1(e): “a third programmable device programmed to identify spoken words in said digital audio stream using Hidden Markov Models and/or Neural Networks”**

186. In my opinion, *Jiang* alone or in combination with *Smyth* teaches or at least renders obvious Claim 1(e). In my opinion, *Jiang* discloses what the '377 Patent describes for phoneme and word identification, namely using HMMs to identify phonemes, and using phonemes to identify words. *Compare '377 Patent*, 11:40-42, 3:12-14, 3:25-29, 4:33-44 (describing Neural Networks in the only disclosure outside of the claims), *with Jiang*, 7:30-43, 8:16-42. Additionally, I note

other prior art teaches such. *Ravishankar*, 19-20, 27-30, 41-45 (discussing phone HMMs and word HMMs; in particular, constructing word HMMs by concatenating phone HMMs); *Nguyen*, 3:11-19 (“Speech recognition concludes with a decoding step 16. The probability that a particular phoneme was spoken is provided by the acoustic models as part of the pattern matching process. A sequence of words can then be constructed by concatenating the phonemes observed during the pattern matching process. The process of combining probabilities for each possible path and searching through the possible paths to select the one with highest probability is commonly referred to as decoding or searching.”). Additionally, *Smyth* teaches using HMMs to identify spoken words, in my opinion.

**1. *Jiang’s Teachings***

**a) “a third programmable device”**

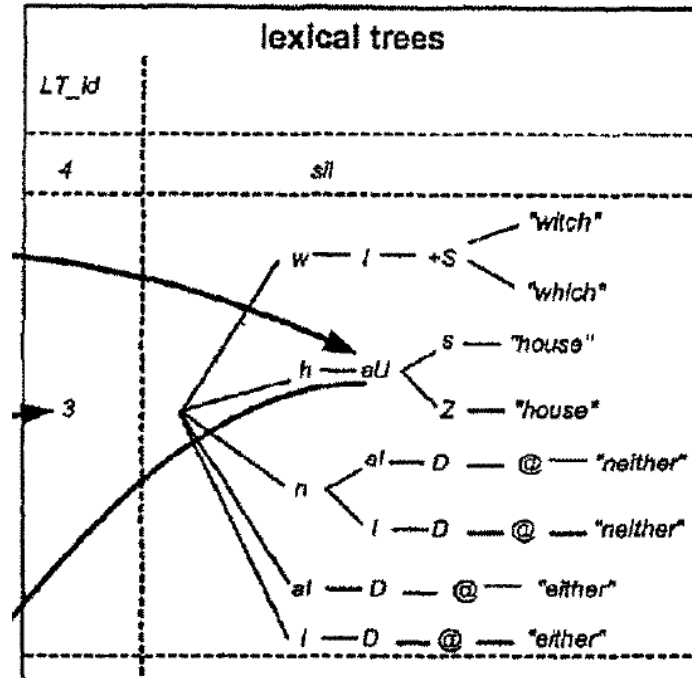
187. I note *Jiang* teaches a tree search engine 74 that “is preferably implemented in CPU 21 (which may include one or more processors) or may be performed by a dedicated speech recognition processor employed by personal computer 20.” *Jiang*, 6:39-42. Therefore, in my opinion, *Jiang* teaches that tree search engine 74 at least includes a processor. I note that processors are well understood to be programmable devices for performing particular tasks, such as identifying spoken words as taught in *Jiang*. Also, as I discussed above, Applicant admitted a CPU is a programmable device. *See* Section VII. In my opinion, *Jiang*

teaches tree search engine 74 is implemented in CPU 21 and includes a dedicated processor and therefore a skilled artisan would have understood or at least found obvious that tree search engine 74 is a third programmable device, as claimed. I also note that, as I discussed above, the tree search engine 74 is mapped as being divided into two DSPs/processors, with one used for performing the claimed distance calculations (*see* Claim 1(c)) and the other used for performing word recognition (Claim 1(e)). I also noted above that it was well understood that processors were programmable devices for performing tasks such as spoken word identification. *See* Sections V.G, X.A-C.

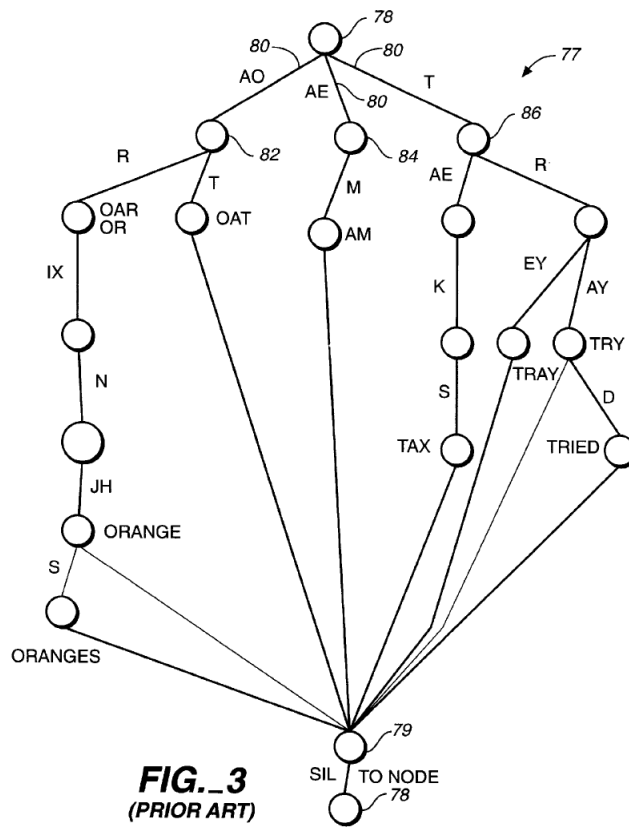
188. In my opinion, a POSITA would have been motivated and found it obvious to modify *Jiang's* tree search engine 74 to include two such programmable devices (e.g., DSPs/processors) for the reasons discussed above. *See* Sections X.A.3, X.C.

**b) “programmed to identify spoken words in said digital audio stream using Hidden Markov Models and/or Neural Networks”**

189. In my opinion, *Jiang's* prefix tree is a lexical tree, as claimed. First, compare exemplary lexical trees as depicted in the '377 *Patent*, Fig. 2B below, with *Jiang*, Figs. 3-5. In particular, I discuss *Jiang's* Fig. 3 depicted below.



'377 Patent, Fig. 2B (excerpt), 8:54-56, 14:34-36, 15:4-9.



*Jiang*, Fig. 3. I note that both the lexical trees depicted in the '377 Patent and *Jiang's* lexical tree example include a silence node and phonemes that lead to a word at the end (i.e., a "leaf"). For example, the '377 Patent depicts phonemes leading to "witch" and "which," and *Jiang's* phonemes lead to "orange," "oranges," and "oat." Also, I note that while the lexical tree is depicted horizontally in Fig. 2B of the '377 Patent and vertically in *Jiang's* prefix trees depicted in Figs. 3-5, a POSITA would have understood that such was merely a difference in style of depicting a lexical tree. For example, the '377 Patent depicts phonemes leading to "witch" and "which," and *Jiang's* phonemes lead to "orange," "oranges," and "oat."

190. Further, a POSITA would have understood that *Jiang's* prefix tree is a lexical tree because *Jiang* also teaches using the prefix tree (i.e., lexical tree) to "arrive at the word most likely indicative of the utterance of the user." *Jiang*, 7:65-8:4. *Jiang* teaches that its prefix tree represents phonemes (i.e., a subword unit) as a phoneme branch and the phoneme branches are associated with a phonetic model such as an HMM. *Jiang*, 7:65-8:4, 2:53-67. In another example of prior art teaching a lexical tree, *Robinson* teaches organizing HMMs into a lexical tree to reduce the computation and wherein each node and associated branch of the lexical tree represents at least one HMM. *Compare Jiang's* teachings with *Robinson* (Ex. 1057), 2:18-22, 2:31-39, Figs. 4-5. Also, *compare Robinson* with *Jiang's* teachings at 2:47-

59 disclosing a similar problem and technique for computation-reduction. Therefore, in my opinion, *Jiang*'s prefix tree is a lexical tree, as claimed.

191. In my opinion, *Jiang* teaches performing a word identification process using the prefix tree where the prefix tree includes at least one HMM for determining a score for a node and an associated phoneme branch of said prefix tree. A POSITA would have understood that the aforementioned score indicates a likelihood that a particular phoneme being examined for a phoneme branch is the phoneme indicated by the codeword. *Jiang*, 7:30-43, 8:16-24. Such teachings from *Jiang* are consistent with the conventional understanding of using a lexical tree for performing word identification. For example, *Ravishankar* teaches such. *Ravishankar*, 29 (“Word HMMs can be built by simply stringing together phonetic HMM models using NULL transitions as appropriate.”). *Ravishankar* further teaches express advantages of utilizing a tree structure. *Ravishankar*, 33.

192. In my opinion, *Jiang* teaches the prefix tree Figure 3, above, illustrates “only a portion of a prefix tree,” where “each of the plurality of branches is associated with a phoneme.” *Jiang*, Fig. 3, 8:5-15. Three phonemes—AO, AE, and T—are shown as nodes 82, 84, and 86, respectively.

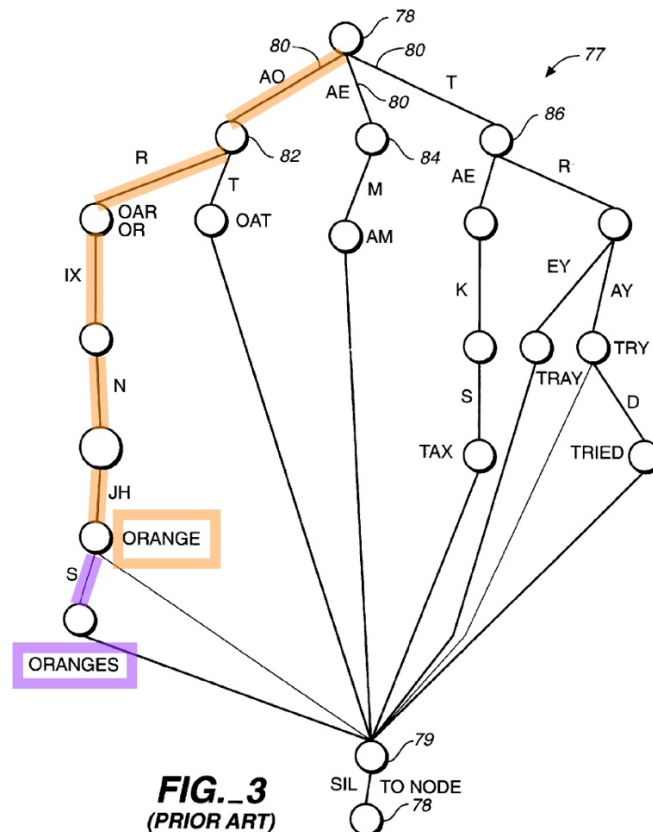
[A]s tree 77 is traversed from the input node 78 to the output node 79, a score is assigned to each node connected to a phoneme branch then under consideration by the speech recognition system. The score is indicative of a likelihood that the particular phoneme being examined

is the actual phoneme indicated by the codeword received from the feature extraction module 66.

*Jiang*, 8:16-24. *Jiang* provides the example of the spoken word being ORANGE, where the feature extraction module divides the word into the phonemes AO, R, IX, N, and JH. *Jiang*, 8:25-28. *Jiang* explains that for each phoneme branch, a score is that “represents the likelihood that the particular phone name encoded by the code word corresponds to the phoneme for the branch under consideration period.” *Jiang*, 8:29-33.

193. *Jiang* discloses the Fig. 3 prefix tree is a partial tree for “the sake of clarity.” *Jiang*, 8:6-7. In my opinion, in a complete prefix tree, each of the used phonemes for the word-search model, e.g., the approximately 40-50 phonemes in the English language (*Jiang*, 3:1-6), would be associated with a phoneme branch, and the phoneme branch would be scored. This is why, for the word ORANGE, *Jiang* discloses that the phonemes AE and T, illustrated in the Fig. 3 prefix tree, receive a lower score than the phone AO. *Id.* The phonemes AE and T are not as representative of the beginning phoneme for the word ORANGE as the phoneme AO identified at node 82. Therefore, “[u]nder ideal circumstances, the score assigned to nodes 84 and 86 [corresponding to phonemes AE and T] is lower than the score assigned to node 82.” *Jiang*, 8:33-43.

194. Regarding a third programmable device that is programmed to identify spoken words, I note that *Jiang* teaches that as the tree is traversed, the score assigned to each node is based “on the likelihood that the present codeword...under analysis is represented by the phoneme corresponding to the branch in tree 77 then being considered, and based on the score assigned to nodes further up the tree which are connected by phoneme branches to the present node.” *Jiang*, 8:44-51, 7:61-8:15. As shown in Fig. 3, the collective phoneme branches AO, R, IX, N, JH (all annotated orange), and S (annotated purple) identify the words ORANGE and ORANGES.



*Jiang*, Fig. 3 (annotated). Additionally, I note that *Jiang* teaches the words in the speech recognition system are stored in a lexicon 70 and that lexicon 70 is presented

to the tree search engine 74 in the form of a prefix tree. *Jiang*, 7:65-8:4. In my opinion, the tree search engine 74 performs several functions, including using the stored lexicon 70 and the most likely phonemes (*Jiang*, 7:30-47 teaches calculation of said phonemes) and the prefix tree to “determine a word which most likely represents the codewords received by feature extraction module 66.” *Jiang*, 7:47-54, 8:16-24. Thus, in my opinion, *Jiang* teaches tree search engine 74 (i.e., third programmable device) is programmed to perform identification of spoken words (i.e., to determine a word) in the codewords. I note that the codewords are the encoded claimed “digital audio stream.” Further, in my opinion, because *Jiang* teaches using a tree search engine that utilizes a prefix tree (i.e., lexical tree) comprising phoneme branches that are “associated with a particular phonetic model (such as an HMM)” (*Jiang*, 2:53-56), *Jiang* therefore teaches or at least renders obvious Claim 1(e).

## 2. *Smyth’s Teachings*

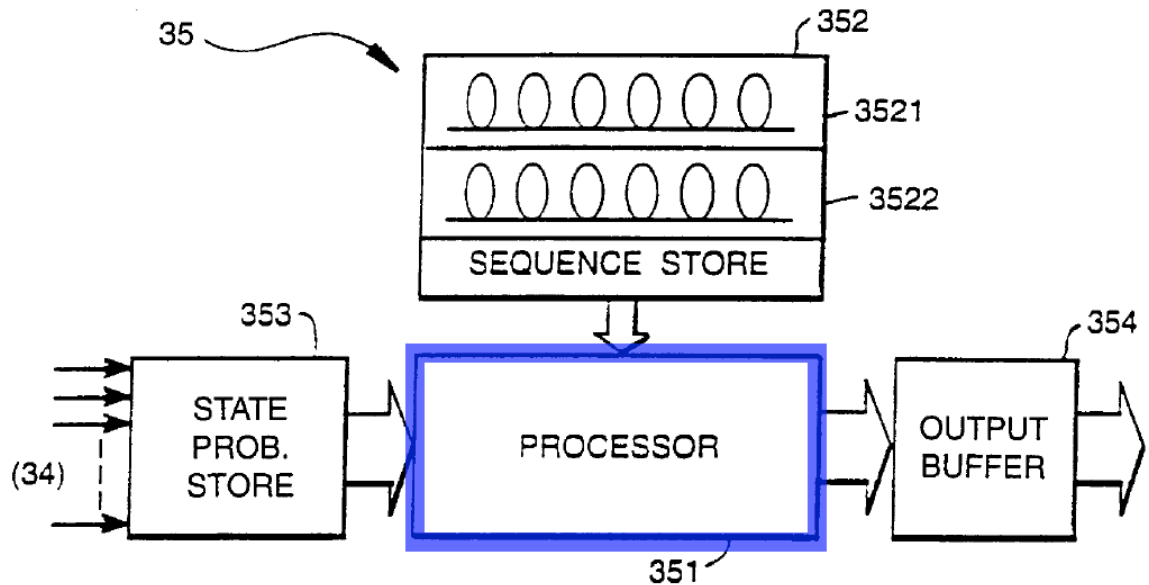
195. First, it is my opinion that a POSITA would have understood that *Smyth’s* “parser processor 351” refers to “parsing processor 351.” In particular, *Smyth* refers to element 351 as a “parser” processor only one time but otherwise refers to element 351 as a “parsing processor 351.” *Smyth*, 6:54, 6:11-12, 6:45-51, 6:58-61, 6:65. Therefore, in my opinion, a POSITA would have understood “parser

processor 351” and “parsing processor 351” to be used interchangeably with no difference in meaning.

**a) “a third programmable device”**

196. In my opinion, *Smyth* teaches a third programmable device. I note that *Smyth* teaches a sequencer 35 that comprises a parsing processor 351, a state sequence memory 352, a state probability memory 353, and a sequencer output buffer 354. *Smyth*, 6:10-15. *Smyth* also teaches that “[t]he parsing processor 351 may, for example, be a microprocessor ... or may alternatively be a DSP device (for example, the same DSP device as is employed for any of the preceding processors).” *Smyth*, 6:47-51. In my opinion, *Smyth* teaches parsing processor 351 is a separate and distinct microprocessor or DSP from the previous two DSPs. As I opined above, it was well known in the prior art that DSPs were programmable devices. *See* Section V.G (discussing *Kloker* and *Newton*); *see* Sections X.A, X.C regarding Claims 1(a), 1(c). In my opinion, parsing processor 351 implemented as a processor/DSP (whether as a “physically distinct processing circuit” as recited in Claim 3 or as the “first programmable device and the third programmable device comprise one physical uni-processor or multi-processor” as recited in Claim 5) is thus a third programmable device, as claimed. Parsing processor 351 (annotated blue) is depicted in Fig. 4 below.

Fig. 4



*Smyth*, Fig. 4 (annotated).

**b) “programmed to identify spoken words in said digital audio stream”**

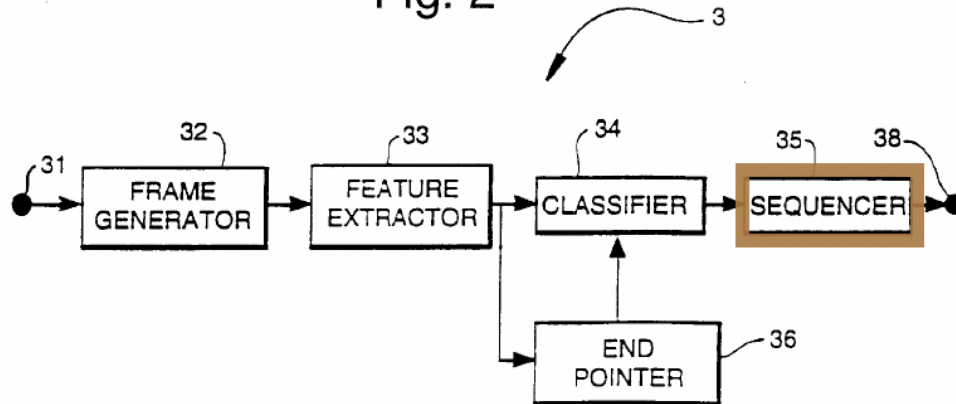
197. As I discussed above for Claim 1(a), *Smyth* teaches a digital audio stream. *See* Claim 1(a). For example, *Smyth* teaches a telecommunications system with a microphone 1 and recognition processor 3 and wherein a speaker speaks into a microphone and the speech signals are analyzed in recognition processor 3. *See Smyth*, 4:22-26, 4:38-41, 4:50-53, 5:1-15. Regarding the claimed “programmed to identify spoken words in said digital audio stream,” in my opinion, *Smyth* teaches a sequencer 35 that comprises parsing processor 351 (i.e., a third programmable device) that performs “name, word, or phrase” recognition on the speech from the

digital audio stream. *Smyth* 6:59-60. *Smyth* teaches an audio input 31 is processed by a frame processor 32, feature extractor 33, classifier 34, and a sequencer 35 that “determine[s] the predetermined utterance to which the sequence of classifier [34] output indicates the greatest similarity[.]”

Referring to FIG. 2, the recognition processor 3 comprises an input 31 for receiving speech in digital form (either from a digital network or from an analog to digital converter), a frame processor 32 for partitioning the succession of digital samples into a succession of frames of contiguous samples; a feature extractor 33 for generating from a frame of samples a corresponding feature vector; a classifier 34 receiving the succession of feature vectors and operating on each with a plurality of model states, to generate recognition results; an end pointer 36 for determining the end of speech, either by detecting the end of speech or after a period of silence, known as a ‘time out’; a sequencer 35 which is arranged to receive the classification results from [sic] the classifier 34 and to determine the predetermined utterance to which the sequence of classifier output indicates the greatest similarity; and an output port 38 at which a recognition signal is supplied indicating the speech utterance which has been recognised.

*Smyth*, 5:1-18. Annotated Fig. 2 below depicts speech input 31, frame processor 32, feature extractor 33, classifier 34, and sequencer 35 (annotated brown).

Fig. 2



*Smyth*, Fig. 2 (annotated).

198. In particular, *Smyth* teaches parsing processor 351 (of sequencer 35) reads probabilities (i.e., distances) output by classifier processor 341 and compares said probabilities with state sequences stored in state sequence memory 352.

The parsing processor 351 is arranged to read, at each frame, the state probabilities output by the classifier processor 341, and the previous stored state probabilities in the state probability memory 353, and to calculate the most likely path of states to date over time, and to compare this with each of the state sequences stored in the state sequence memory 352.

*Smyth*, 6:36-42. *Smyth* further teaches that each state sequence to be recognized corresponds to a “word, phrase or other speech sequence to be recognized” and parsing processor 351 outputs a probability score “at each frame of input speech,” and wherein “the most probable state sequence” indicating the “name, word or

phrase” that was recognized is output from parsing processor 351 to an output port 38.

Accordingly for each state sequence (corresponding to a word, phrase or other speech sequence to be recognised) a probability score is output by the parser processor 351 at each frame of input speech. For example the state sequences may comprise the names in a telephone directory. When the end of the utterance is detected, a label signal indicating the most probable state sequence is output from the parsing processor 351 to the output port 38, to indicate that the corresponding name, word or phrase has been recognised.

The parsing processor 351 comprises a network which is specifically configured to recognise certain phrases or words for example a string of digits.

*Smyth*, 6:52-63. Therefore, in my opinion, *Smyth* teaches parsing processor 351 implemented as a DSP/microprocessor (i.e., a third programmable device) identifies spoken words in said digital audio stream.

199. Further, as I discussed above, *Smyth* teaches the first and second DSPs (i.e., feature extractor 33 mapped as the first programmable device and classifier processor 341 mapped as the second programmable device). *See* Sections X.A, X.C. In my opinion, a POSITA would have understood that *Smyth*’s parsing processor 351 implemented as a DSP would also have been similarly programmed to perform its functions. Indeed, it was well known that DSPs were programmable device. *See*

Section V.G (discussing *Kloker* and *Newton*); see Sections X.A, X.C. In my opinion, a POSITA would have understood that parsing processor 351 implemented as a DSP/processor would have been programmed to perform the functions that *Smyth* teaches. For example, as discussed above, *Smyth* teaches parsing processor 351 identifies words in a digital audio stream. *Smyth*, 5:1-18, 6:36-42, 6:47-63, Fig. 2. Thus, a POSITA would have understood that *Smyth*'s parsing processor 351 mapped as the claimed "third programmable device" would have been programmed to perform identification of words in a digital audio stream.

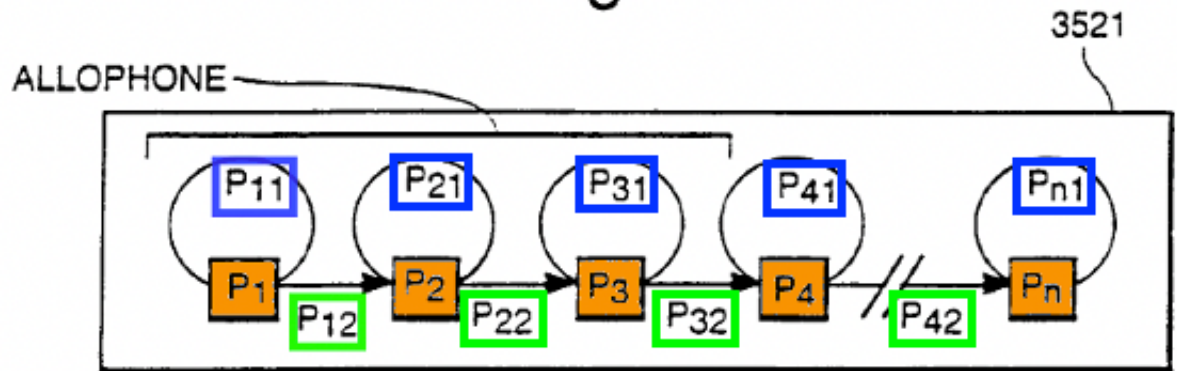
200. As I opined above, in my opinion, *Smyth* teaches utilizing a first DSP for the frame generator 32 and feature extractor 33, a second DSP for the classifier processor 341 of classifier 34, and a third DSP (or microprocessor) for the parsing processor 351 of sequencer 35. See Section X.C.5.

**c) "using Hidden Markov Models and/or Neural Networks"**

201. In my opinion *Smyth* teaches "a third programmable device programmed to identify spoken words in said digital audio stream using Hidden Markov Models and/or Neural Networks," as claimed. In particular, it is my opinion that *Smyth* teaches a standard Hidden Markov Model. For example, annotated Fig. 5 of *Smyth*, below, depicts a standard Hidden Markov Model with states ( $P_1$ ,  $P_2$ ,  $P_3$ ,  $P_4$  ...  $P_n$ ) (annotated orange) and the probability of remaining within a state (e.g.,  $P_{11}$ ,

$P_{21}$ , ... annotated blue) as well as the probability of transitioning to the next state (e.g.,  $P_{12}$ ,  $P_{22}$ , ... annotated green).

Fig. 5



*Smyth*, Fig. 5 (annotated); *Smyth*, 5:59-62, 6:10-26. I note that I discussed Hidden Markov Models in detail above in the Overview of the Technology, including discussing how HMMs are statistical representations of language (e.g., words, phonemes), that HMMs have a number of internal states, and that HMMs pass from an initial state to a final state via transitions. *See* Section V.D. In particular, see my discussion of *Schmandt* and a standard Hidden Markov Model, which is nearly identical to *Smyth*'s Hidden Markov Model depicted in Fig. 5.

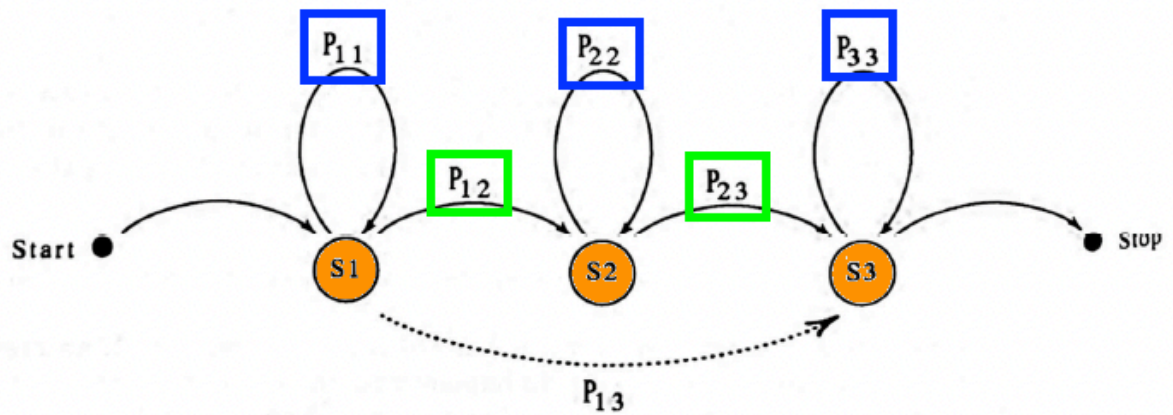


Figure 7.11. A Markov Model.

See Section V.D (discussing *Schmandt*, 59 (“Figure 7.11”)). I note that *Smyth* uses slightly different notation to represent the probabilities (i.e.,  $P_{n1}$  to indicate remaining in a state and  $P_{n2}$  to indicate transitioning to the next state), but in my opinion *Smyth*’s Fig. 5 represents a standard Hidden Markov Model.

202. In detail, *Smyth* expressly teaches using a Hidden Markov Model. *Smyth*, 3:15-17 (“The word representation models may be Hidden Markov Models (HMMs), templates, or any other suitable word representation model.”), *Smyth*, 6:43-46 (“The calculation employs the well known Hidden Markov Model method described in the above referenced Cox paper. Conveniently, the HMM processing performed by the parsing processor 351 uses the well known Viterbi algorithm.”). Additionally, *Smyth* discusses the operations of parsing processor 351 in detail, including a number of states and probabilities for repeat and transition to a following state. See *Smyth*, 6:10-63, Figs. 4-5. *Smyth* also teaches the “state sequence memory

352 comprises a plurality of state sequence fields 3521; 3522 ..., each corresponding to a word or ph[r]ase sequence to be recognised consisting of a string of allophones.” *Smyth*, 6:9-19. Further, *Smyth* teaches the “states of the sequence are a plurality of groups of three states each relating to a single allophone.” *Smyth*, 6:24-26. Thus, in my opinion, *Smyth* teaches “a third programmable device programmed to identify spoken words in said digital audio stream using Hidden Markov Models and/or Neural Networks,” as claimed.

**d) Motivation to Combine *Jiang* and *Smyth***

203. In my opinion, a POSITA would have been motivated and found it obvious to modify *Jiang* to utilize *Smyth*’s word identification using HMMs in *Jiang*’s prefix tree. Such a modification would have advantageously improved the accuracy of threshold scoring and therefore would also have achieved more precise lexical tree pruning than *Jiang* alone. A POSITA would have understood that having more precise pruning would have resulted in more efficient and accurate speech recognition results.

204. A POSITA also would have understood that *Smyth* teaches a state sequence that corresponds to a word or phrase to be recognized. *Smyth*, 6:15-19, 6:53-56. Additionally, *Smyth* teaches that the states of a sequence are a plurality of groups of three states and that each group of three states relates to one allophone. *Smyth*, 6:24-26, 6:15-19, 5:58-62. As I discussed above, allophones are alternative

pronunciations of phonemes. *See* Sections X.C.5-6. I also note that *Smyth* teaches that the states of each state sequence are modeled by an HMM. *Smyth*, 6:43-47. Indeed, *Smyth* therefore teaches that a state sequence comprises a plurality of states and each of the groups of three states relates to one allophone and that the state sequences are modeled by HMMs. *Smyth* also teaches that two probabilities are output for each of the states in the state sequences. One of the output probabilities is a repeat probability and the other is a transition probability to transition to the following state. *Smyth*, 6:20-24. I note that a POSITA would have understood that transition probabilities offer helpful information for speech recognition because said transition probabilities indicate a likely duration of speech corresponding to a state. *See Robinson* (Ex. 1057), 11:12-22, *Jiang*, 1:24-27, *Smyth*, 6:15-19 (disclosing that each state of a state sequence represents an allophone and that a word is a string of allophones). Thus, the transition probabilities in *Smyth* inform the likelihood of spoken speech transitioning to the next state, and in particular the next allophone/phoneme. *See Smyth*, Fig. 5. The HMM is used for word identification and achieves increased accuracy because the HMM includes transition probabilities for each recognized phoneme/allophone.

205. In my opinion, a POSITA would have understood or at least found it obvious to modify *Jiang's* word identification process to include HMMs for word

identification in the prefix tree. I note that *Jiang* suggests using HMMs for word identification.

This corresponding HMM is thus associated with the observed sequence. **This technique can be extended**, such that if each distinct sequence of states in the HMM is associated with a sub-word unit, such as a phoneme, then **a most likely sequence of sub-word units can be found**.

*Jiang*, 1:49-53 (emphasis added). It is also my opinion that *Jiang* teaches using an HMM to identify the most likely sequence of states for a phoneme/allophone, and the technique can be extended to find the most likely sequence of phonemes/allophones, and a POSITA would have understood that a sequence of sub-word units would indicate a word. *See Jiang*, 2:14-16 (disclosing converting the most likely state sequence into a corresponding spoken subword unit, word, or word sequence), 7:45-46 (disclosing a phonetic HMM tree search). In my opinion, *Jiang's* teachings suggest that *Jiang* contemplates using HMMs to perform word identification and also that *Jiang* contemplates using HMMs within a lexical tree search. Indeed, *Jiang* also indicates to a POSITA the importance of transition probabilities in HMMs. *Jiang*, 1:21-24, 1:25-2:15.

206. Therefore, in my opinion, *Jiang* teaches employing transition probabilities and using HMMs for word identification to identify the most likely sequence of sub-word units. I note that *Smyth* provides additional details that

supplement *Jiang's* suggestion regarding using transition probabilities to perform tree searches using HMMs and for word identification. Indeed, in my opinion, transition probabilities are used in *Jiang's* distance calculations when HMMs are utilized for word identification. It is also my opinion that this modification identifies words more accurately. I note *Jiang* teaches distance calculation scores. *Jiang*, 8:16-36. However, each node in *Jiang's* prefix tree does not have a transition probability without using HMMs. The prefix tree without using HMMs has a first score for the observation at the node where the score indicates the likelihood that the codeword's encoded phoneme corresponds to the phoneme for the branch being considered and a second score based on the first score and other first scores for nodes further up the tree. *Jiang*, 8:16-37, 8:44-51.

207. *Jiang* as modified by *Smyth* teaches determining a score that indicates the most likely phoneme for each node in the prefix tree. *Jiang*, 8:16-37. Further, *Jiang* in view of *Smyth* teaches determining a transition probability. *Jiang*, 1:21-24; *Smyth*, 6:20-24. Thus, *Jiang* in view of *Smyth* teaches or at least renders obvious advantageously using transition probability from one node of the prefix tree to the next node to inform the word identification process and prune branches that are unlikely because the modified *Jiang* uses HMMs for word identification and HMMs have transition probability at each state. Therefore, in my opinion, in the modified *Jiang* the transition probability for each node of the prefix tree allows pruning less

likely paths earlier in the recognition process because the transition probability better informs the recognition analysis. Thus, a higher pruning threshold for pruning the prefix tree may be used, which advantageously saves computational time and expense and overall improves the accuracy of the speech recognition process. *Jiang*, 11:6-15.

208. In my opinion, such a modification would have had a reasonable expectation of success because the modification utilizes the well-known technique of using HMMs to model words. Such a technique would have optimized *Jiang's* speech recognition process by applying well known word HMM techniques. While additional memory would need to be utilized for word HMMs, adding additional memory to *Jiang's* system would have resulted in minimal monetary costs and scaling and a POSITA would have been motivated and found it obvious to modify *Jiang* to achieve increased accuracy and resulting speech recognition processing efficiency.

**F. Claim 1(f): “wherein said word identification uses one or more distances that were calculated by the second programmable device,”**

209. I discussed *Jiang's* word identification teachings above. See Claim 1(e). I note that *Jiang* teaches using distances calculated by the second programmable device in its word identification process. Per *Jiang*:

Tree search engine 74 also accesses the lexicon stored in memory 70.  
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 between word boundaries as indicated by silence detection module 68.

*Jiang*, 7:48-54. In my opinion, a POSITA would have understood from *Jiang's* teachings of “information ...based on [] accessing of the phonetic speech unit models” (*Jiang*, 7:49-54) that *Jiang's* probability or likelihood score for each node disclosed at 8:16-42 would have been computed using the HMMs disclosed at 7:30-47 (including disclosing determining a phoneme indicated by the codeword utilizes HMMs). In my opinion, *Jiang* teaches calculating the score that indicates the likelihood that a phoneme being examined in the prefix tree is the phoneme indicated by the codeword (i.e., feature vector received from feature extraction module 66) and that said probability/likelihood score calculation involves using HMMs. Per *Jiang*:

In accordance with one searching technique, as tree 77 is traversed from the input node 78 to the output node 79, 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. Therefore, in my opinion, tree search engine 74 receives information from accessing HMMs that represent phonemes (*Jiang's* phonetic speech unit

models) and said information is used to search lexicon 70 to determine a word representative of the speech utterance. *Jiang*, 7:48-54.

210. In the modified *Jiang*, where the functionality previously performed by the tree search engine is divided into two distinct programmable processors, the distance calculation and word identification would work as follows. In the newly-added programmable processor for performing distance calculations, the most likely phoneme at each node in the prefix tree is determined, per *Jiang*, 7:30-43 and 8:15-42. Then, the second programmable device passes this distance calculation to the third programmable device (e.g., the tree search engine running on a dedicated processor), which then uses the probability score for word identification by accessing the lexicon. Because distance calculation is so computationally-intensive (per *Smyth*), offloading the distance calculation to the second programmable device allows for more efficient word identification by the third programmable device. A POSITA would have understood that as the *Jiang* prefix tree is traversed, the probability scores for each node are constantly being updated as more information comes in for the spoken utterance and as more phonemes are identified. *Jiang*, 8:29 (discussing “[a]s the tree search engine traverses tree 77”).

211. *Jiang* also teaches performing word identification by identifying phonemes, such as AO-R-IX-N-JH for the word ORANGE, which involves using one or more calculated distances. The distances are the probability scores assigned

to each node that represents the likelihood of the phoneme, such as the score for node 82 that indicates the phoneme that corresponds to the codeword is highly likely to be AO. *Jiang*, 8:25-43. I note that in the modified *Jiang*, the score calculations of the codeword to the HMM states (*Jiang*, 7:30-43) are calculated by the distinct DSP (i.e., second programmable device). *See* Sections X.C.4-5. I also note the third programmable device receives the calculated score from the DSP (i.e., second programmable device) for performing word identification via traversing the prefix tree (i.e., lexical tree). In my opinion, in one example, the codeword comparison to HMMs produces a probability score that the phoneme AO was scored as highly likely and therefore the prefix tree eventually uses the AO phoneme that was identified to perform word identification via tree traversal, with the end result being “ORANGE.” *Jiang*, 8:5-28.

FIG. 3 illustrates a prefix tree used in accordance with prior art speech recognition systems. For the sake of clarity, only a portion of a prefix tree is illustrated by FIG. 3. A root node (or input node) 78 is encountered at a first word boundary. A plurality of branches 80 lead from root node 78 to a remainder of the prefix tree. Each of the plurality of branches is associated with a phoneme. In FIG. 3, the branches leaving root node 78 represent only the phonemes represented by the letters AO, AE, and T. The tree extends through other nodes and branches and terminates in an output node 79.

In accordance with one searching technique, as tree 77 is traversed from the input node 78 to the output node 79, 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.

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.

*Jiang*, 8:5-28; *see also Ney* (Ex. 1053), 4 (“[t]he word models are then obtained by concatenating the subword models according to the phonetic transcription of the words in a pronunciation lexicon or dictionary.”).

212. *Jiang* also teaches the score determined for an initial branch of the prefix tree is or is at least based on the score used to determine the initial phoneme of the codeword (feature vector). *See Jiang*, 8:5-43 (disclosing initial nodes include nodes for phonemes such as AO, AE, T and further discussing that said nodes would have each had an assigned score). I note that *Jiang* teaches a relatively “small number of initial branches of the tree” because the initial branches correspond to the number of phonemes in English (approximately 40-50). *Jiang*, 3:1-21. I also note that Fig. 3 depicts initial branches of the prefix tree that is used for word identification. Further, the initial branches of the prefix tree use one or more

probability scores (calculated distances) for the beginning one or more phonemes of the word (e.g., AO, AE, T) for performing word identification.

213. Therefore, in my opinion, because *Jiang's* modified second programmable device (the modified tree search engine 74, *see* Claim 1(c)) calculates the probability score and tree search engine 74 performs word identification using said one or more calculated distances, *Jiang* teaches performing word identification utilizing the prefix tree (i.e., lexical tree) as claimed.

**G. Claim 1(g): “wherein said identification of spoken words uses one or more distances calculated from a first feature vector; and”**

214. I discussed above that identification of spoken words uses a probability score (i.e., distance). *See* Claim 1(f). I also discussed that word identification involves using distances that are calculated from a first feature vector. *See* Claim 1(c). In my opinion, identification of spoken words is for an utterance that corresponds to a first feature vector because the speech utterance is converted from analog to digital samples and said digital samples are encoded as a “first” feature vector. *See Jiang*, 1:58-67, 6:62-7:10. Therefore, in my opinion, the word identification process uses “one or more distances” that are calculated from a first feature vector at least because the said word identification involves using a phoneme that is determined from said first feature vector.

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

215. I note that I discussed a search stage, calculated distances, and lexical tree above. *See* Sections X.F-G. In particular, *Jiang* teaches distance calculations including a probability score that indicates a most likely phoneme based on the codeword (i.e., feature vector) and the HMM. *Jiang*, 7:30-43, 8:15-42, 7:49-54. Then, *Jiang* teaches using said probability score to perform word identification during a search stage including traversing a prefix tree (i.e., lexical tree) and where most likely phonemes are concatenated together to identify a word. *Jiang*, 8:25-51. In my opinion, *Jiang* teaches determining a word by traversing a prefix tree (e.g., as depicted in *Jiang*, Fig. 3) and that such is a search stage, as claimed, because *Jiang* teaches a “tree search” engine 74 for performing word identification involving such tree traversal.

In a preferred embodiment, 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.

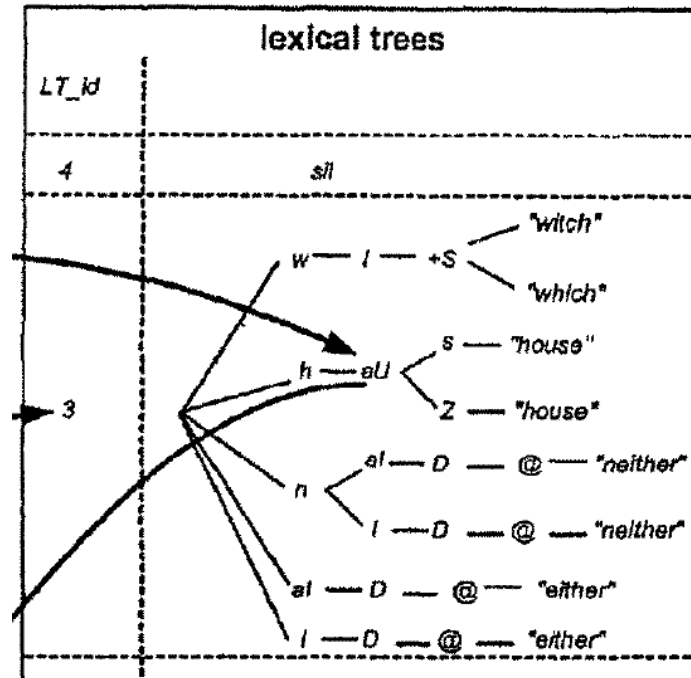
*Jiang*, 7:64-8:4.

216. It is also my opinion that the '377 Patent explains lexical trees were well-known in the art of speech recognition at the time of the alleged invention: "A lexical tree is commonly used in speech recognition[.]" '377 Patent, 11:22, 11:28-33. The '377 Patent also describes lexical trees as comprising states and nodes: "[f]or example, a state ... may be referenced by a node identifier which directly links to a particular *node or group of nodes in the lexical tree.*" '377 Patent, 3:8-12, 7:67-8:2. The '377 Patent describes a lexical tree "includes a model of words." '377 Patent, 2:32-33, 6:3-6, 6:27-30, 7:23-24 ("a lexical tree data structure comprising a model of words[.]"), 7:24-25 ("a state model corresponding to each phone or each group of phones in the lexical tree[.]"). Specifically, the '377 Patent explains "[a] model of words may comprise actual words, phrases, silence, lipsmack sounds, and/or other sounds for recognition by the apparatus." '377 Patent, 38:33-35.

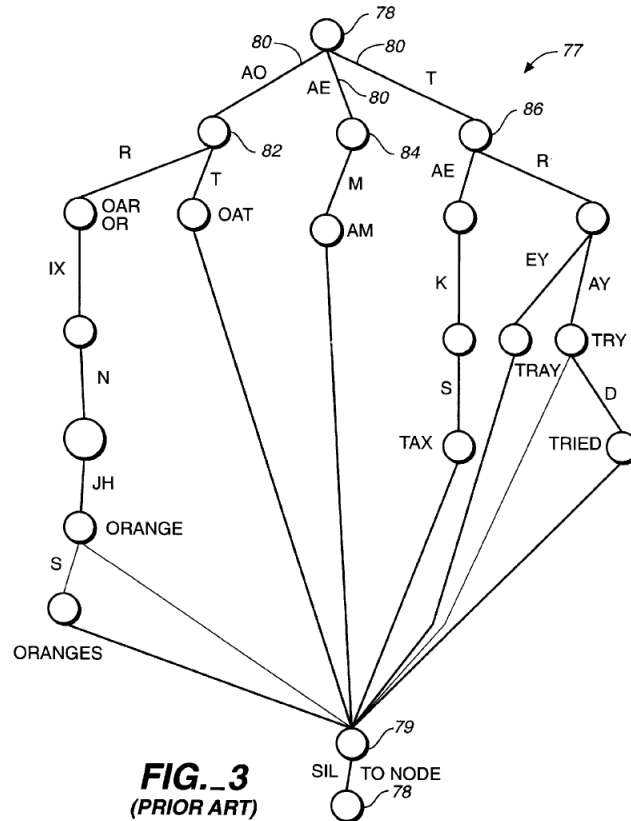
217. It is also my opinion that *Jiang's* prefix tree is a lexical tree, as claimed. In my opinion, *Jiang's* prefix tree is simply a particular form of a lexical tree, where each branch is associated with a phoneme. *Jiang* teaches the prefix tree "represents the lexicon of the speech recognition system 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. Indeed, *Jiang* teaches a model of words, as claimed, because *Jiang's* prefix tree comprises lexicon 70 which includes all the words in the speech recognition system's vocabulary. Further, the '377 Patent describes the model of

words as comprising words. '377 Patent, 38:33-35 (reproduced above). Compare

Jiang's prefix tree with the '377 Patent's exemplary lexical tree.



'377 Patent, Fig. 2B (excerpt), 8:54-56, 14:34-36, 15:4-9.



*Jiang*, Fig. 3.

218. In my opinion, *Jiang*'s prefix tree is a conventional tree structure comprising branches and nodes. *Jiang*, 7:67-8:51 (including 8:51 disclosing that the search and speech recognition process are "all done in a known manner."), Figs. 3-5; *Ravishankar*, Figs. 4.1(b) (illustrating an exemplary basephone lexical tree), 4.2 (illustrating an exemplary triphone lexical tree). Further, the words stored in *Jiang*'s lexicon 70 are "presented to tree search engine 74 in the form of a prefix tree." *Jiang*, 7:64-8:4 (reproduced above). Therefore, in my opinion, *Jiang*'s prefix tree is a lexical tree, as claimed.

**XI. OPINIONS REGARDING GROUND 2: CLAIMS 2-3 ARE OBVIOUS OVER *JIANG*, *SMYTH*, AND *NGUYEN***

**A. Dependent Claim 2**

**“The speech recognition system of claim 1 wherein the time period taken for said identification of spoken words overlaps the time period taken for said second programmable device to calculate one or more distances from a second feature vector.”**

219. *Jiang* in view of *Smyth* teaches or at least renders obvious Claim 1. *See* Claim 1. In my opinion, *Jiang-Smyth-Nguyen* teaches Claim 2, as I opine below.

220. In my opinion, *Nguyen* teaches a speech recognition system with a pattern classification step 12 for computing a similarity measure between frames of input speech and reference patterns.

Next, acoustic pattern matching occurs at step 12. During this step, a similarity measure is computed between each frame of input speech and each reference pattern. The process defines a local measure of closeness between acoustic feature vectors and further involves aligning two speech patterns which may differ in duration and rate of speaking. The pattern classification step uses a plurality of acoustic models 14 generated during the training phase.

*Nguyen*, 2:30-38. *Nguyen* also discloses a decoding step 16 for determining the “probability that a particular phone was spoken[.]”

Speech recognition concludes with a decoding step 16. The probability that a particular phoneme was spoken is provided by the acoustic models as part of the pattern matching process. A sequence of words can then be constructed by concatenating the phonemes observed

during the pattern matching process. The process of combining probabilities for each possible path and searching through the possible paths to select the one with highest probability is commonly referred to as decoding or searching. In other words, the decoding process selects a sequence of words having the highest probability given the observed input speech. A variety of well known searching algorithms may be used to implement the decoding process.

*Nguyen*, 3:11-24. Fig. 1 of *Nguyen* below depicts said pattern classification step 12 and decoding step 16.

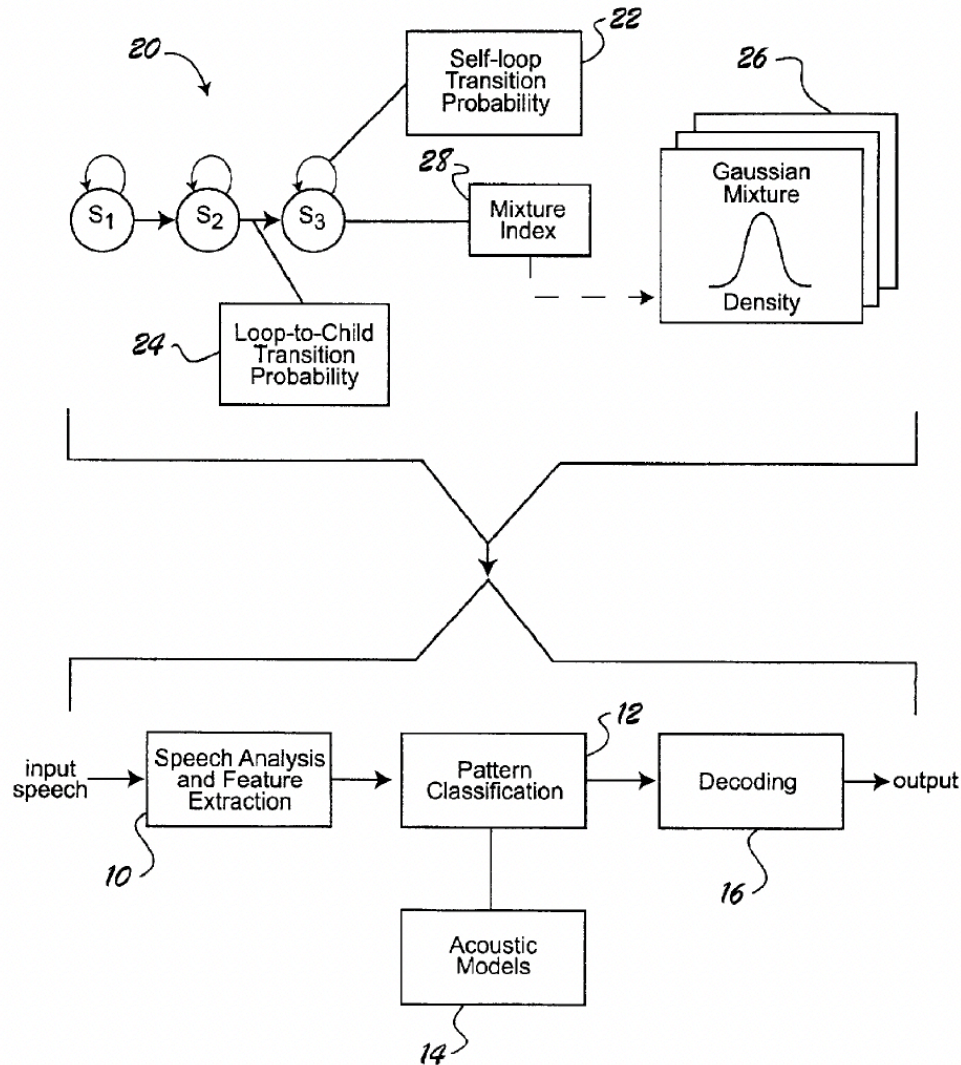


FIG. 1

*Nguyen*, Fig. 1.

221. I note that *Nguyen* teaches performing a similarity measure for each acoustic feature vector in the first group of vectors and storing the resulting similarity measures in an output memory space.

A similarity measure can then be computed at step 36 for each acoustic feature vector in the first group of vectors. For example, a Gaussian

computation may be performed for each acoustic feature vector as is well known in the art. Resulting similarity measures may be stored in an output memory space which is also accessible to the processor performing the computations. By performing the similarity computation for a group of acoustic feature vectors, the present invention reduces the number of I/O operations required to load and unload each acoustic model.

*Nguyen*, 3:51-60. *See also Nguyen*, 3:61-4:7. *Nguyen* teaches “[a]dditional acoustic models are then loaded into the memory space at step 42. Similarity measures are computed for each acoustic feature vector in the first vector group in relation to each of the additional acoustic models resident in the memory workspace at step 36.”

*Nguyen*, 3:63-4:1. *Nguyen* further teaches “[t]his process is repeated via step 40 until similarity measures are computed for the first group of acoustic feature vectors in relation to each of the acoustic models employed by the speech recognition system.”

*Nguyen*, 4:3-7.

222. After the similarity measures are calculated, *Nguyen* teaches performing a “decoding” search process using a searching algorithm.

Once similarity measures have been determined for the first group of acoustic feature vectors, the search process is performed at step 44. In particular, the search process updates the search space based on the similarity measures for the first group of acoustic feature vectors. It is to be understood that this aspect of the present invention is not limited

to a particular searching algorithm, but may be implemented using a variety of well known searching algorithms.

*Nguyen*, 4:8-16. As discussed above, “[t]he process of combining probabilities for each possible path and searching through the possible paths to select the one with the highest probability is commonly referred to as decoding or searching.” *Nguyen*, 3:11-24 (reproduced above).

223. *Nguyen* expressly teaches that contemporaneous with the search process occurring, a similarity measure is computed for a subsequent group of acoustic feature vectors, and that this process is performed iteratively “for each of the acoustic feature vectors that represents the input speech.”

Contemporaneous with the search process, a subsequent group of acoustic feature vectors may be retrieved into the memory workspace at step 48. A similarity measure is computed for each acoustic feature vector in this subsequent group as described above. In other words, acoustic models are loaded and unloaded into the memory workspace and a Gaussian computation is performed for each acoustic feature vector in relation to the acoustic models resident in the memory workspace. This process is repeated via step 40 until similarity measures are computed for the subsequent group of acoustic feature vectors in relation to each of the acoustic models employed by the speech recognition system. It is envisioned that the first group of acoustic feature vectors is removed from the memory workspace prior to loading the subsequent group of acoustic feature vectors into the memory workspace. One skilled in the art will readily recognize that

this is an iterative process that is performed for each of the acoustic feature vectors that represents the input speech.

*Nguyen*, 4:17-36. In my opinion, a POSITA would have understood that some acoustic feature vector in a first group of acoustic feature vectors would have been a “first” feature vector and that some other feature vector in a subsequent group of acoustic feature vectors would have been a “second feature vector,” as claimed. Also, in my opinion, a POSITA would have understood that because a subsequent “pattern classification” step 12 (i.e., distance calculation process) occurs contemporaneously with the decoding step 16 (i.e., identification of said spoken words), *Jiang-Smyth-Nguyen* teaches “the time period taken for said identification of spoken words overlaps the time period taken for said second programmable device to calculate one or more distances from a second feature vector,” as claimed.

224. In my opinion, a POSITA would have been motivated and found it obvious to modify *Jiang-Smyth* according to *Nguyen*’s teachings. Specifically, modifying *Jiang-Smyth* to utilize *Nguyen*’s “group” processing of a group of acoustic feature vectors would have advantageously “reduce[d] the number of I/O operations required to load and unload each acoustic model” as expressly taught by *Nguyen*. *Nguyen*, 3:57-60.

225. Additionally, modifying *Jiang-Smyth* to use its multiple DSPs to perform *Nguyen*’s contemporaneous distance calculation and searching steps would

have advantageously increased *Jiang-Smyth's* system performance by performing both steps in parallel. As I noted above, *Jiang-Smyth* teaches three DSPs (or two DSPs and a microprocessor) are utilized (one DSP for frame generator 32 and feature extractor 33, a second DSP for classifier processor 341, and a third microprocessor/DSP for parsing processor 351). *See* Claims 1(a), 1(c), 1(e). In my opinion, a POSITA would have been motivated to modify *Jiang-Smyth's* multiple DSPs to use *Nguyen's* contemporaneous processing to cause *Jiang-Smyth's* three *Jiang-Smyth's* DSPs to perform tasks in parallel. Such parallel processing would have advantageously increased system performance and would have required only minor modifications. I note that such "contemporaneous" processing was well known in the art. One example is taught by *Ravishankar*, which teaches "exploit[ing] the large degree of concurrency within its algorithmic steps" to achieve "a speed up of 3.9[.]" *Ravishankar* (Ex. 1012), 105.

226. A POSITA would have understood that modifying *Jiang-Smyth* according to *Nguyen's* teachings would have had a reasonable expectation of success. For example, *Nguyen* teaches that it "is envisioned that the number of acoustic feature vectors associated with the first group and the number of acoustic models loaded into the memory workspace should be selected to optimize use of the available memory space." *Nguyen*, 3:46-50. When combining *Jiang-Smyth* with *Nguyen's* teachings, a POSITA would have optimized the memory space for a

number of feature vectors associated with the first group and the number of acoustic models. Additionally, *Jiang-Smyth* would have been modified to generate feature vectors in groups, batch the results, and transmit the group of acoustic feature vectors to classifier processor 341 to perform distance calculations. Then, results from the distance calculations would have been batched and transmitted to parsing processor 351 to perform speech recognition processing. Such batch processing would have had a reasonable expectation of success because it was well known in the art, as taught by *Nguyen*, and because *Jiang-Smyth* teaches utilizing multiple DSPs to perform its speech recognition process. Further, such a modification would have required only minor modifications to transmit multiple feature vectors and distance calculations at a time.

227. Additionally, a POSITA would have had a reasonable expectation of success when modifying *Jiang-Smyth* to use *Nguyen's* “contemporaneous” processing technique. *Jiang-Smyth* itself already teaches an embodiment with three separate DSPs, such that modifying *Jiang-Smyth* to have the distinct DSPs perform processing tasks contemporaneously would have required only minor modifications. Furthermore, such contemporaneous processing was well known in the art, as taught by *Nguyen*. Thus, a POSITA would have had a reasonable expectation of success for said combination.

**B. Dependent Claim 3**

**“The speech recognition system of claim 2 wherein the first programmable device, the second programmable device, and the third programmable device are physically distinct processing circuits.”**

228. I note that the '377 Patent describes “processing circuit” only when discussing a speech processing circuit. *'377 Patent*, 24:5-10 (“The distances calculated by the distance calculation engine are then transferred to the search stage 106 of the speech processing circuit, which uses models such as one or more word models and/or language models to generate and output recognised text.”). The '377 Patent does describe a distance calculation stage and a search stage that are implemented on a CPU or a DSP. *'377 Patent*, 5:52-54 (“In particular, the other stages may be implemented on a CPU, or on a DSP and CPU. The “DSP” and “CPU” may each be implemented as software programmable devices.”), 24:28-33 (“The speech recognition apparatus may also operate with a CPU and DSP (digital signal processor), where the DSP implements the “front-end” signal processing to produce a feature vector, and the CPU implements the search stage, reading the distance results from the distance calculation engine.”). I discussed above that a POSITA would have understood a “programmable device” to include a DSP or a CPU. *See* Sections V.G, VII, X.A, X.C, X.E. In my opinion, the '377 Patent does not describe any particular functional or structural distinction between a processing circuit and a programmable device, and therefore a POSITA would have understood that a

programmable device, such as a DSP, either meets or renders obvious the claimed “processing circuit.”

229. In my opinion, a POSITA would have understood that a DSP is a processing circuit. Indeed, as I discussed above, DSPs were well known processing circuits. *See* Section V.G. For example, a POSITA would have understood the DSPs referenced in *Smyth*, such as the Motorola DSP56000, are processing circuits. *Smyth*, 5:49-53, 6:5-8, 6:43-51. I note that *Kloker* (Ex. 1009), an article from Motorola, teaches that the Motorola DSP56000 is “an 88-pin integrated circuit[.]” *Kloker*, 16. *Kloker* further teaches DSP advantages and that DSPs are digital circuitry. *Kloker*, 2 (“The primary advantage of digital signal processing is that it transforms analog functions into digital hardware. It extends the advantages of digital circuitry—high density, precision, stability, and testability—to those parts of systems previously served by analog components.”

230. As I discussed above, *Smyth* teaches three physically distinct processing circuits (i.e., three separate, distinct DSPs), one for each of the first, second, and third programmable devices. *Smyth*, 5:49-53, 6:5-8, 6:43-51. In my opinions regarding the mapping of Claim 1, I noted that *Jiang*’s feature extraction module 66 is implemented as an array processor or would have been modified to include *Smyth*’s DSP. *See* Claims 1(a), 1(c), 1(e). I also opined that *Jiang*’s tree search engine would have been modified to be two separate and distinct

processors/DSPs per *Smyth*'s teachings. See Claims 1(a), 1(c), 1(e). Therefore, in my opinions regarding Claim 1, I opined that the first, second, and third programmable devices are "physically distinct processing circuits." In more detail, *Jiang*'s feature extractor module 66 is implemented as an array processor (i.e., first programmable device), which is physically distinct from the modified *Jiang*'s tree search engine 74 (modified to include *Smyth*'s DSP) (i.e., second programmable device) to perform distance calculations. *Jiang*'s tree search engine 74 that performs word search/identification is mapped as the claimed "third programmable device." In my opinion, each of these processors/DSPs is a separate programmable device that is physically distinct from each other.

## **XII. OPINIONS REGARDING GROUND 3: CLAIM 4 IS OBVIOUS OVER JIANG, SMYTH, NGUYEN, AND BOIKE**

### **A. Dependent Claim 4**

**"The speech recognition system of claim 3 wherein said physically distinct processing circuits are all formed on a single integrated circuit."**

231. In my opinion, *Boike* (Ex. 1006) teaches a plurality of digital signal processors on a single system-on-a-chip (SoC) integrated circuit. I note that *Boike* discloses such numerous times. *Boike*, Abstract ("The present invention is an integrated circuit containing multiple digital signal processors (DSPs)."). The "Field of the Invention" of *Boike* discloses the same. *Boike*, 1:6-8 ("The invention relates to digital signal processing, and more particularly to integrated circuits containing multiple digital signal processing cores."). *Boike* refers to an embodiment with

multiple DSPs called the DSP/SoC, where multiple DSPs are included on a single integrated circuit. *Boike*, 2:3-9 (“The present invention is a system-on-a-chip (SoC) integrated circuit containing multiple digital signal processors (DSPs). In an embodiment of the invention, hereafter referred to as the DSP/SoC, the integrated circuit includes two or more DSPs and a single host processor interface. Each DSP includes its own memory unit and a direct memory access (DMA) device.”), 2:52-55 (“The present invention is a system-on-a-chip (SoC) integrated circuit 10 containing multiple digital signal processors (DSPs). A preferred embodiment of the invention, hereafter referred to as the DSP/SoC, is shown in FIG. 1.”). I note that I am reproducing annotated Fig. 1 of *Boike* from the Petition, below, which depicts an integrated circuit 10 (annotated blue) with a plurality of DSPs 18-21 (annotated green).

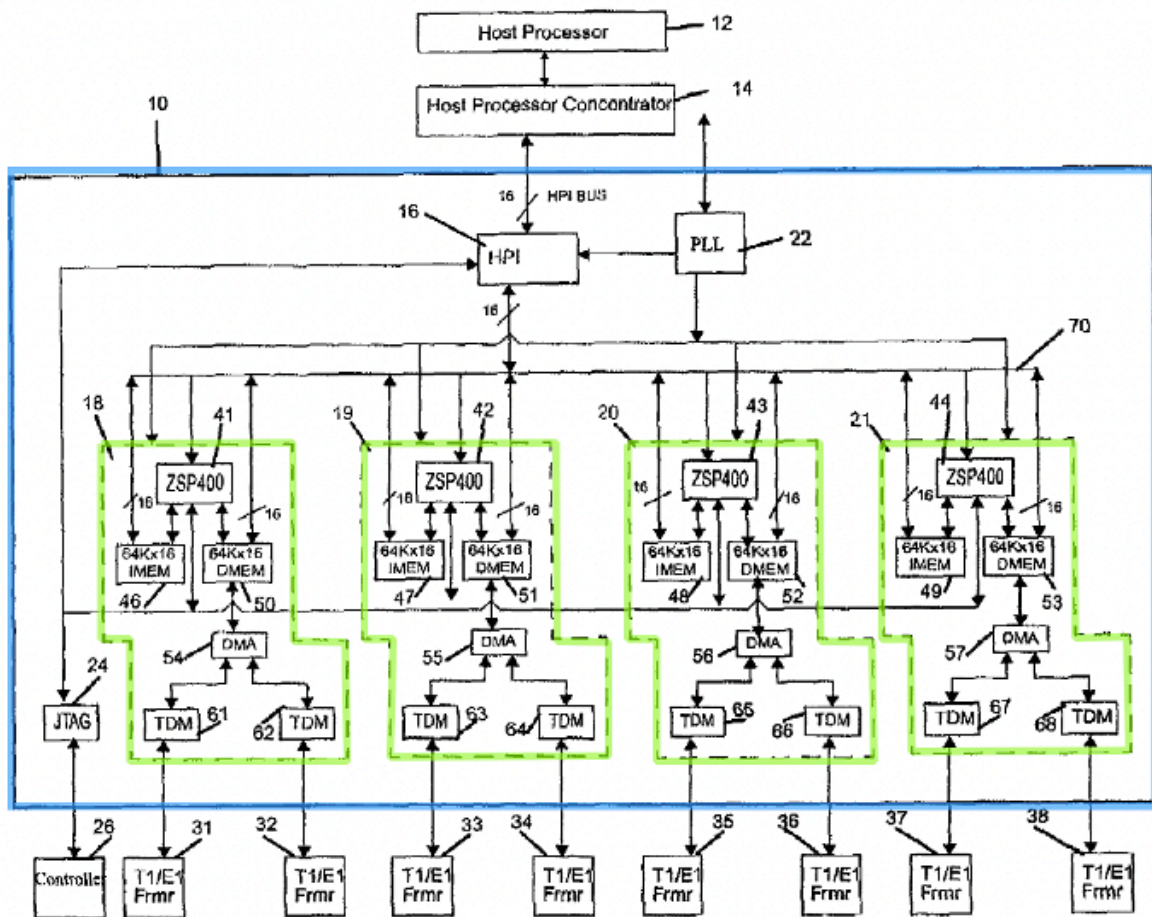


FIGURE 1

*Boike*, Fig. 1 (annotated).

232. I also note that *Boike* expressly teaches several advantages to placing multiple DSPs on a single integrated circuit (IC).

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.

By placing more than one DSP on a semiconductor chip, the DSP/SoC system reduces the number of chips needed on a circuit board to perform digital signal processing functions. This reduction in the number of chips in turn leads to a decrease in power consumption and heat generation and a savings in costs. Processing speed is increased since the distance between DSPs and the number of interconnections among DSPs is decreased.

*Boike*, 2:21-29; *see also Ozelik* teaching advantages for placing multiple DSPs on a single IC. *Ozelik* (Ex. 1034), 1:47-51 (“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.”).

233. In my opinion, a POSITA would have been motivated and found it obvious to modify *Jiang* (as modified by *Smyth* as discussed above) such that *Jiang* would have included its three distinct processing circuits (e.g., DSPs) on a single IC per *Boike*’s teachings. As I noted above, *Boike* discloses numerous express advantages for making such a modification. *See Boike*, 1:19-42, 2:21-29. A POSITA would have been motivated and found it obvious to also modify the modified *Jiang* to capture any of the expressly stated advantages including reducing the space needed on a circuit board, increasing processing speed, decreasing power consumption, decreasing heat generation, and cost savings. A POSITA would further have been motivated and found it obvious because *Boike* teaches that *all* of the aforementioned advantages are a result of placing multiple DSPs on a single integrated circuit, such that a POSITA would have been motivated to capture all such advantages. Therefore, in my opinion, *Boike*’s expressly stated teachings, suggestions, and motivations are sufficient to motivate a POSITA to make such a modification to the modified *Jiang*. Further, a POSITA would have understood that the use of a technique of placing multiple DSPs on a single IC would have

advantageously improved the modified *Jiang's* system in the same way as *Boike's* system.

234. I also note that a POSITA would have had a reasonable expectation of success when making such a modification. *Boike* 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*, 1:43-48. Thus, a POSITA would have understood and found obvious that at least one off-the-shelf DSP component already existed that employed multiple DSPs on a single IC. Therefore, in my opinion, a POSITA would have understood and found obvious to place multiple DSPs on a single integrated circuit and that such a modification would have had a reasonable expectation of success because *Boike* teaches such and because such was already known in the prior art, as *Boike* expressly teaches and because the Texas Instruments TMS320VC5441 chip was “known to exist” and “improved on the traditional configuration.”

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

**A. Dependent Claim 5**

**“The speech recognition system of claim 2 wherein the first programmable device and the third<sup>5</sup> programmable device comprise one physical uni-processor or multi-processor, and the second programmable device is a physically distinct processing circuit.”**

235. Regarding the claimed “multi-processor,” *Jiang* teaches a speech recognition system 60 comprising computer 20 including CPU 21. *Jiang*, Figs. 1-2, 5:4-9. CPU 21 may include multiple processors: “CPU 21 (which may include one or more processors).” *Id.* In my opinion, a CPU including multiple processors is a multi-processor, as claimed, or such is obvious.

236. Regarding the first programmable device, *Jiang* teaches feature extraction module 66 performing the feature extraction of the claimed first programmable device may be hardware modules in computer 20 or software modules accessible by CPU 21, and in a “preferred embodiment, feature extraction module 66 is a conventional array processor.” *Jiang*, 6:30-36, 6:53-57. *Smyth* teaches a single suitably programmed DSP that performs feature vector calculation.

---

<sup>5</sup> I have been informed by counsel that at Claim 5, 39:22, the second programmable device should read --the third programmable device--.

*Smyth*, 5:49-52. *See* Section X.C.5. In my opinions for Claim 1(a), I opined that the first programmable device is met by *Jiang*'s array processor. *See* Section X.A.

237. I discuss below that *Baumgartner* teaches the functionalities of feature vector calculation and word identification being performed by one physical processor, namely processor core 1910, and the functionality of distance calculation being performed by a distinct speech label accelerator including a circuit for distance calculation. In the proposed modification for Claim 5, it would have been obvious to perform the functionalities of the *Jiang* processor or *Smyth* DSP (feature vector calculation) and *Jiang* tree search engine (word identification) using a multiprocessor (as otherwise taught by *Jiang*), such as *Baumgartner*'s processor core 1910. It further would have been obvious to perform computationally-intensive distance calculation via a separate processor, such as *Baumgartner*'s SLA including a circuit.

238. *Jiang*, 6:31-36 already teaches that feature extraction module is "accessible" by CPU or another suitable processor, indicating to a POSITA that CPU performs the functions of feature vector calculation. In my opinion, when collectively reading *Jiang*'s statements that (1) the feature extraction module is accessible by CPU, (2) CPU may include multiple processors, and (3) feature extraction module may be a conventional array processor, *Jiang* teaches or renders

obvious that one of the processors of CPU is a processor for performing feature vector calculation.

239. When *Jiang* is modified to include the DSP of *Smyth* for performing feature vector calculation, it would have similarly been obvious to have the DSP be one of the processors of the *Jiang* CPU 21 including multiple processors. Such a configuration is a mere design choice, as *Jiang* already expressly states the CPU includes multiple processors, and CPUs including DSPs was common.

240. Regarding the third programmable device, I opined above for Claim 1(e) that *Jiang*'s tree search engine (which may be a dedicated processor) modified to perform only the word identification (Section X.E.1) or *Smyth*'s DSP 351 (Section X.E.2) met the claimed third programmable device. In the proposed modification for Claim 5, it would have been obvious to employ *Jiang*'s modified dedicated processor performing the word identification of tree search engine or *Smyth*'s DSP 351 as one of the "multiple processors" of *Jiang*'s CPU 21. Similar to my rationale for the first programmable device, the *Jiang* CPU includes multiple processors. Moreover, *Jiang* teaches "tree search engine 74 is preferably implemented in CPU 21," further indicating that CPU (including one of its multiple processors) performs the functionality of tree search engine. *Jiang* also teaches tree search engine may be a dedicated processor, and it would have been obvious to employ the functionality of the modified tree search engine to perform word identification in a dedicated

processor that is one of the multiple processors of CPU—at least because tree search engine is preferably implemented in CPU.

241. I note that *Baumgartner* teaches using a separate, physically distinct processing circuit (i.e., a speech label accelerator (SLA)) to perform distance calculations and a separate processing circuit for performing feature vector calculations and word identification on one processor core. Per *Baumgartner*:

As previously discussed, the Speech Label Accelerators (SLAs) of the present invention speed the labeling phase of the speech recognition computation. Due to the length of the discussion here, a number of headings are used for readability purposes. The headings should not be considered limiting in any way.

## 1. Introduction

The organization of this document is as follows. Section 2 begins with a review of concepts and notation, and also describes the mathematical task that comprises labeling. Here, the key characteristic of separability, which a function should exhibit for the present invention to apply, is identified. In Section 3, the basic theory behind the invention is explained, and it is shown how the invention exploits separability. Then in Section 4, it is shown how the techniques of the previous sections may be designed as an apparatus. This section includes a discussion of the wide variety of hardware designs that are possible for the same basic idea. In Section 5, various extensions and refinements are described. In Section 6, a variety of practical issues, such as interaction with a speech

recognition system and the load that the SLA places on a system bus, are discussed.

*Baumgartner*, [0031]-[0033]. Further, per *Baumgartner*:

This is more easily explained with reference to a figure. Turning now to FIG. 19, a system 1900 is shown comprising a processor core 1910, an SLA 1950, and a main memory 1760, all interconnected through bus 1948. FIG. 19 is used mainly to illustrate a sequence of steps that are performed, and which device performs which step. Processor core 1910 comprises steps 1905 through 1935. SLA 1950 comprises three steps 1960 through 1980. As before, main memory 1760 comprises atom value memory copies 1940-1 and 1940-2. The sequence of steps that will now be described is merely exemplary, and the sequence should not be construed to be limiting.

Assume that there are 20 dimensions, and the SLA 1950 is designed for 10 dimensions; also assume that there are 64 atoms per dimension and that the SLA 1950 is designed for 64 atoms per dimension. Cepstrum computation 1905 is performed first. Next, atom value computation 1910 is performed. Step 1910 creates two sets of atom values, each set having 10 dimensions. Atom value memory copy 1940-1 contains atom values for the first 10 dimensions (i.e., dimensions 0 through 9), while atom value memory copy 1940-2 contains atom values for the next 10 dimensions (i.e., dimensions 10 through 19). In step 1920, additional Gaussians, from the first level, are selected for evaluation. The SLA performs step 1960, which performs a level one density computation, as previously described. The processor core 1910 then selects some of the Gaussians for the next level (step 1925), and SLA 1950 performs a

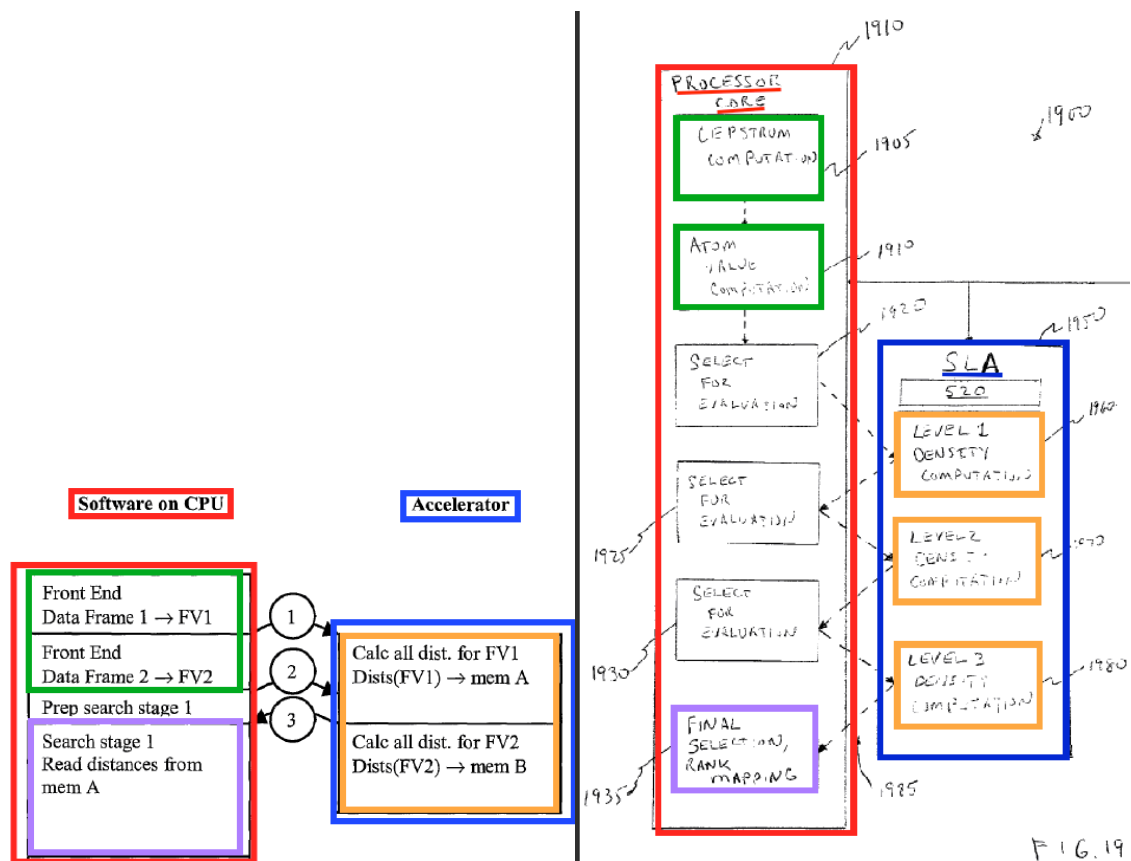
level two density computation in step 1970. Again, more Gaussians are selected (step 1930 by processor core 1910), this time from the last level, and SLA 1950 performs a level 3 density computation in step 1980. The processor core 1910 performs step 1935, where it performs a final selection and rank mapping. In FIG. 19, lines 1985 indicated the flow of operations, while lines 1990 indicate the “zig-zag” that occurs with atom value memory copies 1940-1 and 1940-2.

The zig-zag method works as follows. When SLA 1950 performs step 1960, main memory 1760 loads the atom value memory copy 1940-1 into the atom value memory 520 and performs calculations for the first set of dimensions. Then, SLA 1950 loads the atom value memory copy 1940-2 into the atom value memory 520 and performs calculations for the second set of dimensions. When SLA 1950 performs step 1970, the atom value memory copy 1940-2 is already loaded into the atom value memory 520. Consequently, the level two density computation 1970 proceeds backwards, in that atom value memory copy 1940-2 is used first for calculations for the second set of dimensions. Then, atom value memory copy 1940-1 is loaded into atom value memory 520 and used for calculations for the first set of dimensions. When level three density computation 1980 is performed, it again proceeds forwards, in that atom value memory copy 1940-1 is used first for calculations for the first set of dimensions. Then, atom value memory copy 1940-1 is loaded into atom value memory 520 and used for calculations for the second set of dimensions. Lines 1990 indicate the “zig-zag” that occurs.

*Baumgartner*, [0156]-[0158], Fig. 19. I have reproduced the comparison between annotated Fig. 21 of the '377 Patent and annotated Fig. 19 of *Baumgartner* from the Petition below. I repeat the description from the Petition for simplicity:

*Baumgartner's* Fig. 19 (annotated below) illustrates a cepstrum computation (i.e., feature vector calculation) 1905 and a final selection rank mapping (i.e., word identification) 1935. Both the cepstrum computation 1905 and the rank mapping 1935 are implemented on processor core 1910. Density computations (i.e., distance calculations) 1960-1980 are implemented on Speech Label Accelerator (SLA) 520.

'377 Petition, Claim 5.



'377 Patent, Fig. 21 (excerpt, left, annotated); *Baumgartner*, Fig. 19 (right, annotated).

242. In my opinion, *Baumgartner* teaches a similar hardware architecture to that described in the '377 Patent and depicted at Fig. 21, and also recited in Claim 5. Additionally, regarding the claimed “physically distinct processing circuit,” I note that the Petition at Section VI.B discusses that “the '377 Patent does not ascribe any particular distinction, either structural or functional, to a ‘processing circuit’ as compared to a ‘programmable device[.]’” Petition at Claim 3. In my opinion, a POSITA would have understood that because no particular distinction is ascribed to a processing circuit to distinguish such from a programmable device, that a programmable device would have met or at least rendered obvious the claimed processing circuit.

243. Regardless, I note that *Baumgartner* teaches “the heart of the SLA is a circuit that permits the fast approximate computation of a large collection of values  $[P]$  for a fixed observation vector  $v$ ,” thus expressly teaching a physically distinct circuit, as claimed. *Baumgartner*, [0043].

244. Further, it is my opinion that a POSITA would have understood a DSP is a processing circuit. For example, a POSITA would have understood the DSPs referenced in *Smyth*, such as the Motorola DSP56000, are processing circuits. See Section XI.B (Claim 3). Therefore, *Smyth* teaches three physically distinct

processing circuits (i.e., three separate, distinct DSPs), one for each of the first, second, and third programmable devices.

245. In my opinion, because *Jiang* already teaches CPU 21 includes multiple processors, it would have been obvious to modify *Jiang*, to the extent any modification is required, to configure the multiple processors of CPU as the processor core 1910 of *Baumgartner* including (1) *Jiang*'s processor or *Smyth*'s DSP for performing feature vector calculation; and (2) *Jiang*'s tree search engine in the form of a dedicated processor and modified to only perform word identification or *Smyth*'s DSP 351. A POSITA would have been motivated and found it obvious to further modify the *Jiang-Smyth-Nguyen* combination according to *Baumgartner* so that the computationally-intensive processing task of distance calculation was performed by a physically distinct circuit or programmable device. Per *Baumgartner*:

Speech recognition, in which speech is translated to text, is currently a computationally intensive task. Additionally, much of the recognition is performed in software. For a personal computer or a workstation, recognition of the speech of a single person generally does not adversely affect the operation of the computer. This is true because the person usually has only the speech recognition system running, along with perhaps a word processor. Moreover, current processors are quite fast and memory is relatively inexpensive.

However, for some systems, such as a Personal Digital Assistant (PDA), speech recognition is problematic. Similar devices include other small, hand-held devices and set-top boxes. These devices generally do not contain enough processing power to run a speech recognizer and other software. Consequently, speech recognition is relatively unused on these devices.

Another system that can have problems with speech recognition is a large-scale recognition system. In this type of system, multiple utterances from different speakers are translated to text simultaneously. While it is possible to purchase larger and faster systems in order to help translate this amount of speech, it would be beneficial to reduce the load on the processors of the system when the system performs speech recognition. The latter would allow the system to use the extra processing power on other tasks or to accommodate additional simultaneous speakers.

Consequently, a need exists for techniques to reduce the computational load of a processor during speech recognition.

*Baumgartner*, [0003]-[0006]. In my opinion, a POSITA would have understood that *Baumgartner's* invention “frees one or more processors from performing labeling” because the “invention provides Speech Label Accelerators (SLAs), ... [that] speed[] processing during the labeling stage of a speech recognizer[.]” *Baumgartner*, [0007]. I note that *Baumgartner's* labeling stage that the SLA performs is similar to a distance calculation. *Baumgartner*, [0038]. Further, a POSITA would have understood and found obvious that *Baumgartner's* hardware architecture yields an

efficient solution to issues identified in the prior art regarding computational loads and utilizing an accelerator to perform expensive calculations. *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.”); *see also Jiang*, 11:17-24 (discussing similar issues regarding the desirability of retraining more hypotheses to achieve better recognition; the faster the putative phoneme distances can be calculated, the more time is available for word evaluation while still performing in real time); *Ravishankar*, 33, 105.

#### **XIV. OPINIONS REGARDING GROUND 5: CLAIM 6 IS OBVIOUS OVER *JIANG, SMYTH, NGUYEN, BAUMGARTNER, AND BOIKE***

##### **A. Dependent Claim 6**

**“The speech recognition system of claim 5 wherein said one physical uni-processor or multi-processor and said second programmable device are all formed on a single integrated circuit.”**

246. As I opined above, the *Jiang-Smyth-Nguyen-Boike-Baumgartner* combination teaches Claims 1, 2, and 5. In my opinion, the *Jiang-Smyth-Nguyen-*

*Boike-Baumgartner* combination teaches Claim 6, as I discuss below. First, I note that Claim 6, similar to Claim 4, recites a single IC with processors. *See* Claim 4.

247. In my opinion, a POSITA would have been motivated and found it obvious to modify *Jiang-Smyth-Nguyen-Boike-Baumgartner* to include a multi-processor that comprises first and third DSPs and also includes a distinct second DSP that is mapped for performing distance calculations. *See* Claim 5. In my opinion, *Baumgartner* expressly teaches placing said multi-processor and second distinct DSP on a single integrated circuit. *Baumgartner*, [0180] (“CPU core 2005 and the SLA 500 reside on the same physical die.”), [0177] (“System 200 shows a system 2000 where SLA 500 is built on the same silicon substrate as CPU core 2005.”), Fig. 20. A POSITA would have been motivated and found it obvious to thus modify *Jiang-Smyth-Nguyen-Boike-Baumgartner* per *Baumgartner’s* teachings. Such a use of a known technique would have advantageously captured advantages of placing multiple processing units on a single IC, as expressly taught by *Boike*. *Boike*, 1:19-42, 2:21-29; *see also Ozelik* teaching advantages for placing multiple DSPs on a single IC. *Ozelik* (Ex. 1034), 1:47-51 (“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.”); *see* Claim 4.

248. I also note that while the mapping above focuses on the embodiment of Fig. 19 of *Baumgartner*, a POSITA would have been motivated and found it obvious to combine this aspect of *Baumgartner*'s embodiment of Fig. 20 (placing the CPU core and SLA on a single "silicon substrate"/"physical die") with the embodiment of Fig. 19 and would have done so with a reasonable expectation of success. *Baumgartner* expressly contemplates such a hardware architecture in the embodiment of Fig. 20, and the embodiment of Fig. 20 likewise utilizes separate processing units for the SLA and processor core as does the embodiment of Fig. 19. *Baumgartner*, Figs. 19-20. Further, *Baumgartner* teaches that in "Section 6.1, the integration of the SLA with software and hardware is described." *Baumgartner*, [0173]. Thus, a POSITA would have understood that the embodiment of Fig. 20 provides more details on *Baumgartner*'s system and does not teach an embodiment that would teach away or otherwise disparage combining the embodiments of Fig. 19 and Fig. 20. *Baumgartner*, [0033] (disclosing Section 6 addresses a "variety of practical issues, such as interaction with a speech recognition system and the load that the SLA places on a system bus[.]"). Therefore, in my opinion, *Jiang-Smyth-Nguyen-Boike-Baumgartner* teaches or at least renders obvious Claim 6.

**XV. CONCLUSION**

249. For the aforementioned reasons, in my opinion, to a POSITA, the claims of the '377 Patent are rendered obvious by the prior art.

250. 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. If called to testify as to the truth of the matters stated herein, I could and would testify competently.

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 on user input 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. Anilonyte, 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.

Amazon / Zentian Limited

Exhibit 1003

Page 209

*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. Angebrannndt, 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.

---