

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. IPR2023-00036  
U.S. Patent No. 10,839,789

**DECLARATION OF CHRISTOPHER SCHMANDT  
IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF  
U.S. PATENT NO. 10,839,789**

**TABLE OF CONTENTS**

|                                                            |                  |
|------------------------------------------------------------|------------------|
| <b><i>I. INTRODUCTION AND QUALIFICATIONS.....</i></b>      | <b><i>14</i></b> |
| A. Educational Background and Professional Experience..... | 14               |
| <b><i>II. METHODOLOGY; MATERIALS CONSIDERED.....</i></b>   | <b><i>17</i></b> |
| <b><i>III. OVERVIEW AND LEGAL STANDARDS.....</i></b>       | <b><i>20</i></b> |
| A. Person of Ordinary Skill in the Art.....                | 20               |
| B. Obviousness .....                                       | 21               |
| C. Analogous Art.....                                      | 27               |
| D. Claim Construction .....                                | 27               |
| <b><i>IV. LEVEL OF A PERSON OF ORDINARY SKILL.....</i></b> | <b><i>28</i></b> |
| <b><i>V. OVERVIEW OF THE TECHNOLOGY.....</i></b>           | <b><i>29</i></b> |
| A. Speech Recognition .....                                | 29               |
| B. Feature Vectors .....                                   | 37               |
| C. Acoustic Models.....                                    | 46               |
| D. Hidden Markov Models.....                               | 47               |
| E. Distance Calculations.....                              | 56               |
| F. Gaussian Distribution and Probability .....             | 59               |
| G. Speech Recognition System Hardware.....                 | 62               |
| H. Pipelining .....                                        | 70               |
| I. Interrupts .....                                        | 74               |
| J. Prior Art Speech Recognition Systems .....              | 76               |
| <b><i>VI. OVERVIEW OF THE '789 PATENT.....</i></b>         | <b><i>77</i></b> |
| <b><i>VII. OVERVIEW OF THE PRIOR ART.....</i></b>          | <b><i>78</i></b> |
| A. Overview of Smyth.....                                  | 78               |
| B. Overview of Mozer .....                                 | 78               |
| C. Overview of Nguyen.....                                 | 79               |
| D. Overview of Houser.....                                 | 80               |
| E. Overview of Vensko .....                                | 80               |
| F. Overview of Comeau.....                                 | 80               |

|              |                                                                                                                                                                                                                                 |            |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>G.</b>    | <b>Overview of Boike .....</b>                                                                                                                                                                                                  | <b>81</b>  |
| <b>H.</b>    | <b>Overview of Whittaker .....</b>                                                                                                                                                                                              | <b>81</b>  |
| <b>I.</b>    | <b>Overview of Ichikawa .....</b>                                                                                                                                                                                               | <b>82</b>  |
| <b>J.</b>    | <b>Overview of Seshadri .....</b>                                                                                                                                                                                               | <b>82</b>  |
| <b>K.</b>    | <b>Overview of Steinbiss .....</b>                                                                                                                                                                                              | <b>83</b>  |
| <b>VIII.</b> | <b><i>PROPOSED GROUNDS OF UNPATENTABILITY.....</i></b>                                                                                                                                                                          | <b>83</b>  |
| <b>IX.</b>   | <b><i>OPINIONS REGARDING GROUND 1: CLAIMS 1, 10-11, 14, AND 17<br/>ARE OBVIOUS OVER SMYTH AND MOZER .....</i></b>                                                                                                               | <b>84</b>  |
| <b>A.</b>    | <b>Independent Claim 1 .....</b>                                                                                                                                                                                                | <b>84</b>  |
| 1.           | Claim 1[Pre]: “An acoustic coprocessor for processing data associated with an audio signal, comprising:” .....                                                                                                                  | 84         |
| 2.           | Claim 1(a): “a first interface for receiving at least one feature vector, wherein the feature vector is determined from the audio signal” .....                                                                                 | 103        |
| 3.           | Claim 1(b): “an acoustic model memory for storing an acoustic model defining a plurality of acoustic states;” .....                                                                                                             | 110        |
| 4.           | Claim 1(c): “a calculating apparatus for calculating distances indicating a similarity between the at least one feature vector and respective acoustic states of the acoustic model read from the acoustic memory; and” .....   | 120        |
| 5.           | Claim 1(d): “a second interface for sending at least one distance calculated by the calculating apparatus;” .....                                                                                                               | 126        |
| 6.           | Claim 1(e): “wherein the calculating apparatus and the acoustic model memory are fabricated on a single integrated circuit.” .....                                                                                              | 131        |
| <b>B.</b>    | <b>Independent Claim 10 .....</b>                                                                                                                                                                                               | <b>136</b> |
| 1.           | Claim 10[Pre]: “An acoustic coprocessor, comprising:” .....                                                                                                                                                                     | 136        |
| 2.           | Claim 10(a):” a first interface for receiving at least one feature vector;” .....                                                                                                                                               | 136        |
| 3.           | Claim 10(b): “a calculating apparatus for calculating distances indicating a similarity between the at least one feature vector and at least one acoustic state of an acoustic model read from an acoustic model memory;” ..... | 136        |

|           |                                                                                                                                                                                                                                          |                   |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| 4.        | Claim 10(c): “a second interface for sending at least one distance calculated by the calculating apparatus.” .....                                                                                                                       | 136               |
| <b>C.</b> | <b>Dependent Claim 11 .....</b>                                                                                                                                                                                                          | <b>136</b>        |
| 1.        | Claim 11: “The acoustic coprocessor of claim 10, wherein the calculating apparatus and the acoustic model memory are fabricated on a single integrated circuit.” .....                                                                   | 136               |
| <b>D.</b> | <b>Dependent Claim 14 .....</b>                                                                                                                                                                                                          | <b>137</b>        |
| 1.        | Claim 14:” The acoustic coprocessor of claim 10, wherein the second interface for sending distances is a serial bus.” .....                                                                                                              | 137               |
| <b>E.</b> | <b>Dependent Claim 17 .....</b>                                                                                                                                                                                                          | <b>138</b>        |
| 1.        | Claim 17: “The acoustic coprocessor of claim 10, wherein the acoustic coprocessor is capable of, in response to receiving a feature vector, autonomously calculating the distances for each acoustic state of the acoustic model.” ..... | 138               |
| <b>X.</b> | <b><i>OPINIONS REGARDING GROUND 2: CLAIMS 2, 4, 6-9, 12, 16, 18, AND 20-21 ARE OBVIOUS OVER SMYTH, MOZER, AND NGUYEN .....</i></b>                                                                                                       | <b><i>139</i></b> |
| <b>A.</b> | <b>Dependent Claim 2 .....</b>                                                                                                                                                                                                           | <b>139</b>        |
| 1.        | Claim 2: “The acoustic coprocessor of claim 1, wherein the acoustic model contains Gaussian parameters and the distances are probabilities” .....                                                                                        | 139               |
| <b>B.</b> | <b>Dependent Claim 4 .....</b>                                                                                                                                                                                                           | <b>141</b>        |
| 1.        | Claim 4: “The acoustic coprocessor of claim 2, wherein the second interface for sending distances is a serial bus.” ..                                                                                                                   | 141               |
| <b>C.</b> | <b>Dependent Claim 6 .....</b>                                                                                                                                                                                                           | <b>141</b>        |
| 1.        | Claim 6: “The acoustic coprocessor of claim 4, wherein the acoustic coprocessor, in response to receiving a feature vector, autonomously calculates the distance for an associated acoustic state of the acoustic model.” .....          | 141               |
| <b>D.</b> | <b>Dependent Claim 7 .....</b>                                                                                                                                                                                                           | <b>142</b>        |
| 1.        | Claim 7: “The acoustic coprocessor of claim 6, further comprising a result memory, wherein the distance for an associated acoustic state of the acoustic model is stored in the result memory.” .....                                    | 142               |

|           |                                                                                                                                                                                                                                                                                                                               |            |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>E.</b> | <b>Dependent Claim 8</b> .....                                                                                                                                                                                                                                                                                                | <b>143</b> |
| 1.        | Claim 8: “The acoustic coprocessor of claim 7, wherein the result memory is configured so that distances calculated from a first feature vector are sent over the second interface for sending while at least one distance calculated from a second feature vector is being stored in the one or more result memories.” ..... | 143        |
| <b>F.</b> | <b>Dependent Claim 9</b> .....                                                                                                                                                                                                                                                                                                | <b>145</b> |
| 1.        | Claim 9(Pre): “A speech recognition system comprising the acoustic coprocessor of claim 7, the speech recognition system further comprising at least one CPU, wherein:” .....                                                                                                                                                 | 145        |
| 2.        | Claim 9(a): “the CPU executes software to perform or initiate a speech recognition word search using the distances calculated by the acoustic coprocessor in response to receiving the first feature vector, and” .....                                                                                                       | 146        |
| 3.        | Claim 9(b): “a time period for the speech recognition word search overlaps a time period for the acoustic coprocessor to calculate the distance in response to receiving the second feature vector.” .....                                                                                                                    | 149        |
| <b>G.</b> | <b>Dependent Claim 12</b> .....                                                                                                                                                                                                                                                                                               | <b>156</b> |
| 1.        | Claim 12: “The acoustic coprocessor of claim 10, wherein the acoustic model contains Gaussian parameters and the distances are probabilities.” .....                                                                                                                                                                          | 156        |
| <b>H.</b> | <b>Dependent Claim 16</b> .....                                                                                                                                                                                                                                                                                               | <b>156</b> |
| 1.        | Claim 16 [Pre]: “A speech recognition system comprising the acoustic coprocessor of claim 10, the speech recognition system further comprising at least one CPU, wherein:” .....                                                                                                                                              | 156        |
| 2.        | Claim 16(a): “the CPU executes software to perform or initiate a speech recognition word search using one or more distances calculated by the acoustic coprocessor in response to receiving a first feature vector, and” .....                                                                                                | 156        |
| 3.        | Claim 16(b): “a time period for the speech recognition word search overlaps with a time period for the acoustic                                                                                                                                                                                                               |            |

|                                                                                                                                                                                                                |            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| coprocessor to calculate one or more distances in<br>response to receiving a second feature vector.” .....                                                                                                     | 156        |
| <b>I. Dependent Claim 18</b> .....                                                                                                                                                                             | <b>157</b> |
| 1. Claim 18: “The acoustic coprocessor of claim 17, further<br>comprising one or more result memories configured to<br>store the distances for each acoustic state of the acoustic<br>model.” .....            | 157        |
| <b>J. Dependent Claim 20</b> .....                                                                                                                                                                             | <b>157</b> |
| 1. Claim 20: “The acoustic coprocessor of claim 18,<br>wherein the calculating apparatus is capable of<br>computing distances using acoustic models representing<br>Gaussian probability distributions.” ..... | 157        |
| <b>K. Dependent Claim 21</b> .....                                                                                                                                                                             | <b>157</b> |
| 1. Claim 21: “The acoustic coprocessor of claim 20,<br>wherein the acoustic model uses a neural network to<br>calculate at least one of the distances.” .....                                                  | 157        |
| <b>XI. OPINIONS REGARDING GROUND 3: CLAIM 13 IS OBVIOUS OVER<br/>SMYTH, MOZER, AND HOUSER</b> .....                                                                                                            | <b>159</b> |
| <b>A. Dependent Claim 13</b> .....                                                                                                                                                                             | <b>159</b> |
| 1. Claim 13:” The acoustic coprocessor of claim 10,<br>wherein the acoustic model memory is a FLASH<br>memory.” .....                                                                                          | 159        |
| <b>XII. OPINIONS REGARDING GROUND 4: CLAIMS 22-24 ARE OBVIOUS<br/>OVER SMYTH, MOZER, NGUYEN, AND VENSKO</b> .....                                                                                              | <b>160</b> |
| <b>A. Dependent Claim 22</b> .....                                                                                                                                                                             | <b>160</b> |
| 1. Claim 22: “The acoustic coprocessor of claim 21,<br>wherein the acoustic coprocessor is capable of generating<br>an interrupt in response to a completion of calculation of<br>a set of distances.” .....   | 160        |
| <b>B. Dependent Claim 23</b> .....                                                                                                                                                                             | <b>161</b> |
| 1. Claim 23: “The acoustic coprocessor of claim 22,<br>wherein the acoustic model memory includes a plurality<br>of storage locations within a static random access<br>memory (SRAM).” .....                   | 161        |
| <b>C. Dependent Claim 24</b> .....                                                                                                                                                                             | <b>163</b> |

1. Claim 24: “The acoustic coprocessor of claim 23, wherein the SRAM and the calculating apparatus are fabricated on a single integrated circuit.” ..... 163

***XIII. OPINIONS REGARDING GROUND 5: CLAIMS 25-28 AND 37 ARE OBVIOUS OVER SMYTH, MOZER, NGUYEN, VENSKO, AND COMEAU 164***

**A. Dependent Claim 25 ..... 164**

1. Claim 25: “The acoustic coprocessor of claim 22, further comprising a status register, wherein the acoustic coprocessor is capable of setting one or more bits in the status register to values that indicate the completion of calculation of the set of distances.” ..... 164

**B. Dependent Claim 26 ..... 168**

1. Claim 26: “The acoustic coprocessor of claim 25, wherein the acoustic coprocessor further comprises a direct memory access (DMA) controller capable of copying distances from the one or more result memories to a buffer memory.” ..... 168

**C. Dependent Claim 27 ..... 170**

1. Claim 27: “The acoustic coprocessor of claim 26, wherein the DMA controller is capable of copying distances in a plurality of sequential bursts.” ..... 170

**D. Dependent Claim 28 ..... 175**

1. Claim 28[Pre]: “A speech recognition system comprising the acoustic coprocessor of claim 26, wherein:” ..... 175
2. Claim 28(a): “the speech recognition system further comprises at least one processor configured to execute software to perform or initiate a speech recognition word search using one or more distances calculated by the acoustic coprocessor in response to receiving a first feature vector, and” ..... 175
3. Claim 28(b): “a time period for the speech recognition word search overlaps with a time period for the acoustic coprocessor to calculate one or more distances in response to receiving a second feature vector.” ..... 175

**E. Dependent Claim 37 ..... 175**

1. Claim 37: “A speech recognition system comprising the acoustic coprocessor of claim 27, the speech recognition system further comprising at least one microphone.” ..... 175

***XIV.OPINIONS REGARDING GROUND 6: CLAIM 29 IS OBVIOUS OVER SMYTH, MOZER, NGUYEN, VENSKO, COMEAU, AND BOIKE ..... 176***

**A. Dependent Claim 29 ..... 176**

1. Claim 29: “The speech recognition system of claim 28, wherein the acoustic coprocessor and the at least one processor are fabricated on a single integrated circuit.” ..... 176

***XV. OPINIONS REGARDING GROUND 7: CLAIM 35 IS OBVIOUS OVER SMYTH, MOZER, NGUYEN, VENSKO, COMEAU, AND WHITTAKER..... 181***

**A. Dependent Claim 35 ..... 181**

1. Claim 35: “The acoustic coprocessor of claim 26, wherein the acoustic model includes parameters represented with integer data.” ..... 181

***XVI.OPINIONS REGARDING GROUND 8: CLAIMS 38-39 ARE OBVIOUS OVER SMYTH, MOZER, NGUYEN, VENSKO, COMEAU, AND ICHIKAWA 183***

**A. Dependent Claim 38 ..... 183**

1. Claim 38: “A speech recognition system comprising the acoustic coprocessor of claim 27, the speech recognition system further comprising a plurality of microphones.” ..... 183

**B. Dependent Claim 39 ..... 184**

1. Claim 39: “The speech recognition system of claim 38, further comprising the buffer memory.” ..... 184

***XVII. OPINIONS REGARDING GROUND 9: CLAIMS 42-44 ARE OBVIOUS OVER SMYTH, MOZER, NGUYEN, VENSKO, COMEAU, ICHIKAWA, AND SESHADRI ..... 184***

**A. Dependent Claim 42 ..... 184**

1. Claim 42: “A laptop or tablet personal computer comprising the speech recognition system of claim 39.” ..... 184

**B. Dependent Claim 43 ..... 185**

1. Claim 43: “A mobile phone comprising the speech recognition system of claim 39.” ..... 185

Declaration of Chris Schmandt  
Patent No. 10,839,789

|           |                                                                                                                |            |
|-----------|----------------------------------------------------------------------------------------------------------------|------------|
| <b>C.</b> | <b>Dependent Claim 44</b> .....                                                                                | <b>186</b> |
| 1.        | Claim 44: “An electronic entertainment product<br>comprising the speech recognition system of claim 39.” ..... | 186        |

***XVIII. OPINIONS REGARDING GROUND 10: CLAIM 45 IS OBVIOUS  
OVER SMYTH, MOZER, NGUYEN, VENSKO, COMEAU, ICHIKAWA, AND  
STEINBISS188***

|           |                                                                                         |            |
|-----------|-----------------------------------------------------------------------------------------|------------|
| <b>A.</b> | <b>Dependent Claim 45</b> .....                                                         | <b>188</b> |
| 1.        | Claim 45: “An appliance comprising the speech<br>recognition system of claim 39.” ..... | 188        |

***XIX.CONCLUSION* .....** **189**

**CLAIM LISTING**

**Claim 1:**

**Claim 1[Pre]** An acoustic coprocessor for processing data associated with an audio signal, comprising:

**1(a)** first interface for receiving at least one feature vector, wherein the feature vector is determined from the audio signal;

**1(b)** an acoustic model memory for storing an acoustic model defining a plurality of acoustic states;

**1(c)** a calculating apparatus for calculating distances indicating a similarity between the at least one feature vector and respective acoustic states of the acoustic model read from the acoustic model memory; and

**1(d)** a second interface for sending at least one distance calculated by the calculating apparatus;

**1(e)** wherein the calculating apparatus and the acoustic model memory are fabricated on a single integrated circuit.

**Claim 2:**

The acoustic coprocessor of claim 1, wherein the acoustic model contains Gaussian parameters and the distances are probabilities.

**Claim 4:**

The acoustic coprocessor of claim 2, wherein the second interface for sending distances is a serial bus.

**Claim 6:**

The acoustic coprocessor of claim 4, wherein the acoustic coprocessor, in response to receiving a feature vector, autonomously calculates the distance for an associated acoustic state of the acoustic model.

**Claim 7:**

The acoustic coprocessor of claim 6, further comprising a result memory, wherein the distance for an associated acoustic state of the acoustic model is stored in the result memory.

**Claim 8:**

The acoustic coprocessor of claim 7, wherein the result memory is configured so that distances calculated from a first feature vector are sent over the second interface for sending while at least one distance calculated from a second feature vector is being stored in the one or more result memories.

**Claim 9:**

**Claim 9[Pre]** A speech recognition system comprising the acoustic coprocessor of claim 7, the speech recognition system further comprising at least one CPU, wherein:

**9(a)** the CPU executes software to perform or initiate a speech recognition word search using the distances calculated by the acoustic coprocessor in response to receiving the first feature vector, and

**9(b)** a time period for the speech recognition word search overlaps a time period for the acoustic coprocessor to calculate the distance in response to receiving the second feature vector.

**Claim 10:**

**Claim 10[Pre]** An acoustic coprocessor, comprising:

**10(a)** a first interface for receiving at least one feature vector;

**10(b)** a calculating apparatus for calculating distances indicating a similarity between the at least one feature vector and at least one acoustic state of an acoustic model read from an acoustic model memory; and

**10(c)** a second interface for sending at least one distance calculated by the calculating apparatus.

**Claim 11:**

The acoustic coprocessor of claim 10, wherein the calculating apparatus and the acoustic model memory are fabricated on a single integrated circuit.

**Claim 12:**

The acoustic coprocessor of claim 10, wherein the acoustic model contains Gaussian parameters and the distances are probabilities.

**Claim 13:**

The acoustic coprocessor of claim 10, wherein the acoustic model memory is a FLASH memory.

**Claim 14:**

The acoustic coprocessor of claim 10, wherein the second interface for sending distances is a serial bus.

**Claim 16:**

**Claim 16 [Pre]** A speech recognition system comprising the acoustic coprocessor of claim 10, the speech recognition system further comprising at least one CPU, wherein:

**16(a)** the CPU executes software to perform or initiate a speech recognition word search using one or more distances calculated by the acoustic coprocessor in response to receiving a first feature vector, and

**16(b)** a time period for the speech recognition word search overlaps with a time period for the acoustic coprocessor to calculate one or more distances in response to receiving a second feature vector.

**Claim 17:**

The acoustic coprocessor of claim 10, wherein the acoustic coprocessor is capable of, in response to receiving a feature vector, autonomously calculating the distances for each acoustic state of the acoustic model.

**Claim 18:**

The acoustic coprocessor of claim 17, further comprising one or more result memories configured to store the distances for each acoustic state of the acoustic model.

**Claim 20:**

The acoustic coprocessor of claim 18, wherein the calculating apparatus is capable of computing distances using acoustic models representing Gaussian probability distributions.

**Claim 21:**

The acoustic coprocessor of claim 20, wherein the acoustic model uses a neural network to calculate at least one of the distances.

**Claim 22:**

The acoustic coprocessor of claim 21, wherein the acoustic coprocessor is capable of generating an interrupt in response to a completion of calculation of a set of distances.

**Claim 23:**

The acoustic coprocessor of claim 22, wherein the acoustic model memory includes a plurality of storage locations within a static random access memory (SRAM).

**Claim 24:**

The acoustic coprocessor of claim 23, wherein the SRAM and the calculating apparatus are fabricated on a single integrated circuit.

**Claim 25:**

The acoustic coprocessor of claim 22, further comprising a status register, wherein the acoustic coprocessor is capable of setting one or more bits in the status register to values that indicate the completion of calculation of the set of distances.

**Claim 26:**

The acoustic coprocessor of claim 25, wherein the acoustic coprocessor further comprises a direct memory access (DMA) controller capable of copying distances from the one or more result memories to a buffer memory.

**Claim 27:**

The acoustic coprocessor of claim 26, wherein the DMA controller is capable of copying distances in a plurality of sequential bursts.

**Claim 28:**

**Claim 28[Pre]** A speech recognition system comprising the acoustic coprocessor of claim 26, wherein:

**28(a)** the speech recognition system further comprises at least one processor configured to execute software to perform or initiate a speech recognition word search using one or more distances calculated by the acoustic coprocessor in response to receiving a first feature vector, and

**28(b)** a time period for the speech recognition word search overlaps with a time period for the acoustic coprocessor to calculate one or more distances in response to receiving a second feature vector.

**Claim 29:**

The speech recognition system of claim 28, wherein the acoustic coprocessor and the at least one processor are fabricated on a single integrated circuit.

**Claim 35:**

The acoustic coprocessor of claim 26, wherein the acoustic model includes parameters represented with integer data.

**Claim 37:**

A speech recognition system comprising the acoustic coprocessor of claim 27, the speech recognition system further comprising at least one microphone.

**Claim 38:**

A speech recognition system comprising the acoustic coprocessor of claim 27, the speech recognition system further comprising a plurality of microphones.

**Claim 39:**

The speech recognition system of claim 38, further comprising the buffer memory.

**Claim 42:**

A laptop or tablet personal computer comprising the speech recognition system of claim 39.

**Claim 43:**

A mobile phone comprising the speech recognition system of claim 39.

**Claim 44:**

An electronic entertainment product comprising the speech recognition system of claim 39.

**Claim 45:**

An appliance comprising the speech recognition system of claim 39.

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. 10,839,789 (“the ’789 Patent”) (Ex. 1001) including the specification and claims, and the ’789 Patent’s prosecution history (“’789 File History”) (Ex. 1002). In developing my opinions relating to the ’789 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. 10,839,789 (“’789 Patent”)                                                                                                |
| <b>Exhibit 1002</b>   | Prosecution History for the ’789 Patent (“’789 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 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”)                                |
| <b>Exhibit 1011</b>   | U.S. Patent No. 6,499,012 to Peters et al. (“Peters”)                                                                                     |

Declaration of Chris Schmandt  
Patent No. 10,839,789

|                     |                                                                                                                                                                                                             |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <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, <i>Fundamentals of Speech Recognition</i> , Pearson Education, 1993 (“ <i>Rabiner</i> ”)                                                                                                  |
| <b>Exhibit 1016</b> | Richard L. Klevans, <i>Voice Recognition</i> , Artech House Inc., 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, <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> , The MIT Press, 1997 (“ <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 R. Rabiner, <i>A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition</i> , Proceedings of the IEEE, VOL. 77, NO. 2, pp. 257-286, February 1989 (“ <i>Rabiner89</i> ”) |
| <b>Exhibit 1023</b> | U.S. Patent No. 6,182,037 to Maes (“ <i>Maes</i> ”)                                                                                                                                                         |
| <b>Exhibit 1024</b> | U.S. Patent Application Publication No. 2003/0220796 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 Stubley et al. (“ <i>Stubley</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 1040</b> | U.S. Patent No. 5,774,859 to Houser et al. (“ <i>Houser</i> ”)                                                                                                                                              |
| <b>Exhibit 1041</b> | U.S. Patent No. 7,171,358 to Whittaker et al. (“ <i>Whittaker</i> ”)                                                                                                                                        |

Declaration of Chris Schmandt  
Patent No. 10,839,789

|                     |                                                                                                                                                                                                                                                                               |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Exhibit 1042</b> | U.S. Patent No. 7,478,041 to Ichikawa et al. (“ <i>Ichikawa</i> ”)                                                                                                                                                                                                            |
| <b>Exhibit 1043</b> | U.S. Patent No. 7,587,318 to Seshadri (“ <i>Seshadri</i> ”)                                                                                                                                                                                                                   |
| <b>Exhibit 1044</b> | U.S. Patent Application Publication No. 2002/0091826 to Comeau et al. (“ <i>Comeau</i> ”)                                                                                                                                                                                     |
| <b>Exhibit 1045</b> | U.S. Patent Application Publication No. 2003/0012347 to Steinbiss et al. (“ <i>Steinbiss</i> ”)                                                                                                                                                                               |
| <b>Exhibit 1046</b> | U.S. Patent No. 6,832,194 to Mozer et al. (“ <i>Mozer</i> ”)                                                                                                                                                                                                                  |
| <b>Exhibit 1047</b> | U.S. Patent No. 6,879,954 to Nguyen et al. (“ <i>Nguyen</i> ”)                                                                                                                                                                                                                |
| <b>Exhibit 1048</b> | U.S. Patent No. 5,150,330 to Hag (“ <i>Hag</i> ”)                                                                                                                                                                                                                             |
| <b>Exhibit 1049</b> | U.S. Patent No. 5,724,587 to Carmon et al. (“ <i>Carmon</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.), Automatic Speech and Speaker Recognition (“ <i>Ney</i> ”)                                                                     |
| <b>Exhibit 1054</b> | U.S. Patent No. 7,058,678 to Luick et al. (“ <i>Luick</i> ”)                                                                                                                                                                                                                  |
| <b>Exhibit 1058</b> | Abdulmesih Aktas et al., <i>Multi-DSP and VQ-ASIC Based Acoustic Front-End for Real-Time Speech Processing Tasks</i> , First European Conference on Speech Communication and Technology EUROSPEECH '89, Paris, France, September 1989 ( <i>Atkas 1989</i> )                   |
| <b>Exhibit 1059</b> | Abdulmesih Aktas et al. <i>Speaker Independent Continuous Hmm-Based Recognition of Isolated Words on A Real-Time Multi-Dsp System</i> , Second European Conference on Speech Communication and Technology EUROSPEECH '91, Genova, Italy, September 1991 ( <i>Atkas 1991</i> ) |
| <b>Exhibit 1060</b> | U.S. Patent No. 6,029,221 to Wu et al. (“ <i>Wu</i> ”)                                                                                                                                                                                                                        |
| <b>Exhibit 1062</b> | U.S. Patent No. 5,734,604 to Akamatsu et al. (“ <i>Akamatsu</i> ”)                                                                                                                                                                                                            |
| <b>Exhibit 1063</b> | U.S. Patent No. 6,130,462 to Yang et al. (“ <i>Yang 462</i> ”)                                                                                                                                                                                                                |
| <b>Exhibit 1064</b> | U.S. Patent No. 6,128,311 to Poulis et al. (“ <i>Poulis</i> ”)                                                                                                                                                                                                                |
| <b>Exhibit 1065</b> | U.S. Patent No. 6,185,634 to Wilcox (“ <i>Wilcox</i> ”)                                                                                                                                                                                                                       |
| <b>Exhibit 1066</b> | U.S. Patent No. 6,567,953 to Pomerantz (“ <i>Pomerantz</i> ”)                                                                                                                                                                                                                 |

11. I have considered these materials from the viewpoint of a POSITA as of the priority date of the '789 Patent. For the purposes of this declaration, I have been asked to assume that the priority date of the '789 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 '789 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 ’789 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 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 inter partes 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 ’789 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, 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 inter partes 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 '789 Patent, the prior art cited herein, and the ordinary skill factors described in this section, a POSITA in the field of the '789 Patent at the time of the earliest possible priority date (September 14, 2004) would have been knowledgeable regarding the field of speech recognition technology. In my experience working in this field, most workers of ordinary skill in the art as of the earliest possible priority date of September 14, 2004, would have had a master's degree in computer engineering, computer science, electrical engineering, or a related field, with at least two years of experience in the field of speech recognition, or a bachelor's degree in the same fields with at least four years of experience in the field of speech recognition. Additional education or experience might substitute for the above requirements. *See, e.g.*, the '789 Patent (Ex. 1001), 1:20-2:23 (describing the "Field of the Invention" and providing a "Description of the Related Art"). 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 '789 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.C 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.

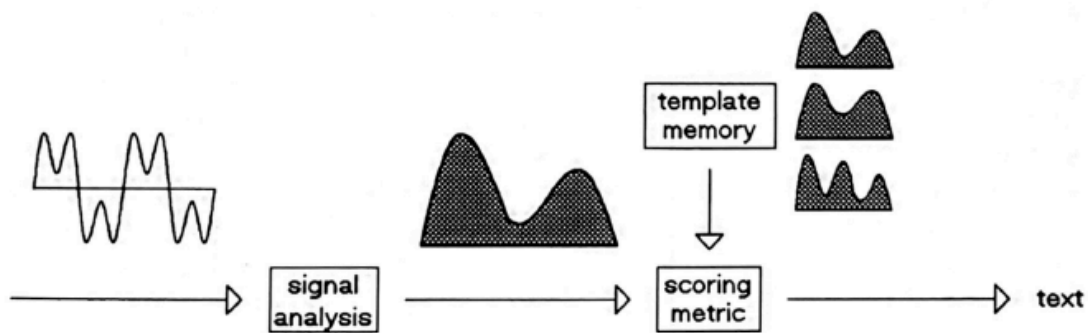


**Figure 7.1.** The functional elements of a speech recognizer. The user's speech is digitized and converted to the recognizer's internal representation. The captured speech is then compared with the words stored in the recognizer's template memory. The pattern matching algorithm then determines which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

Declaration of Chris Schmandt  
Patent No. 10,839,789

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.*

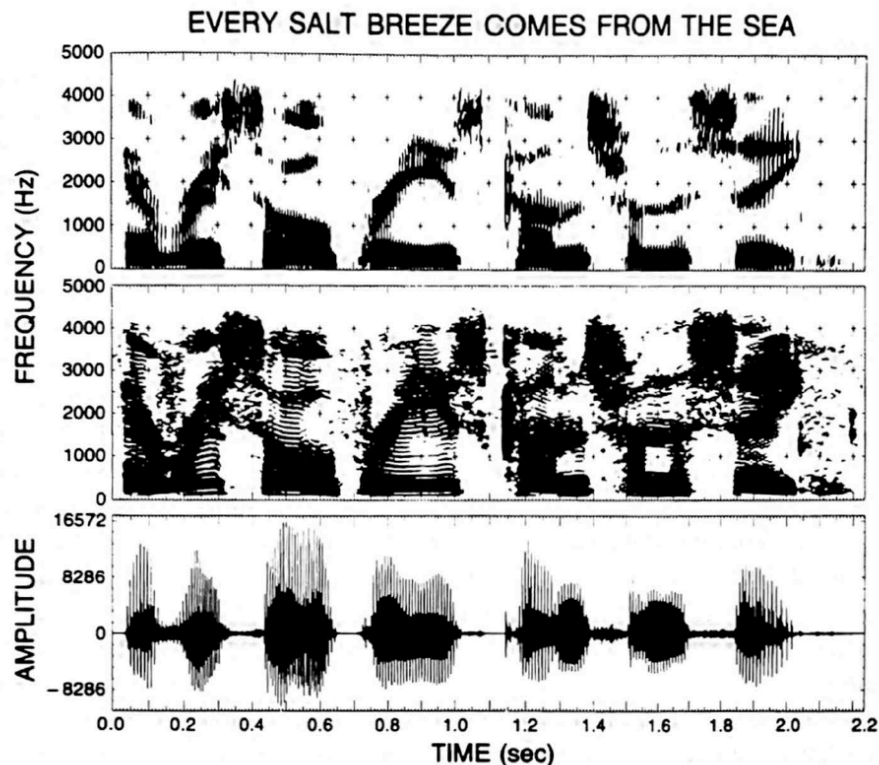


**Figure 2.8** Wideband and narrowband spectrograms and speech amplitude for the utterance "Every salt breeze comes from the sea."

*Rabiner*, 50 ("Figure 2.8").

33. One speech recognition technique known well before the priority date of the '789 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 ’789 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.

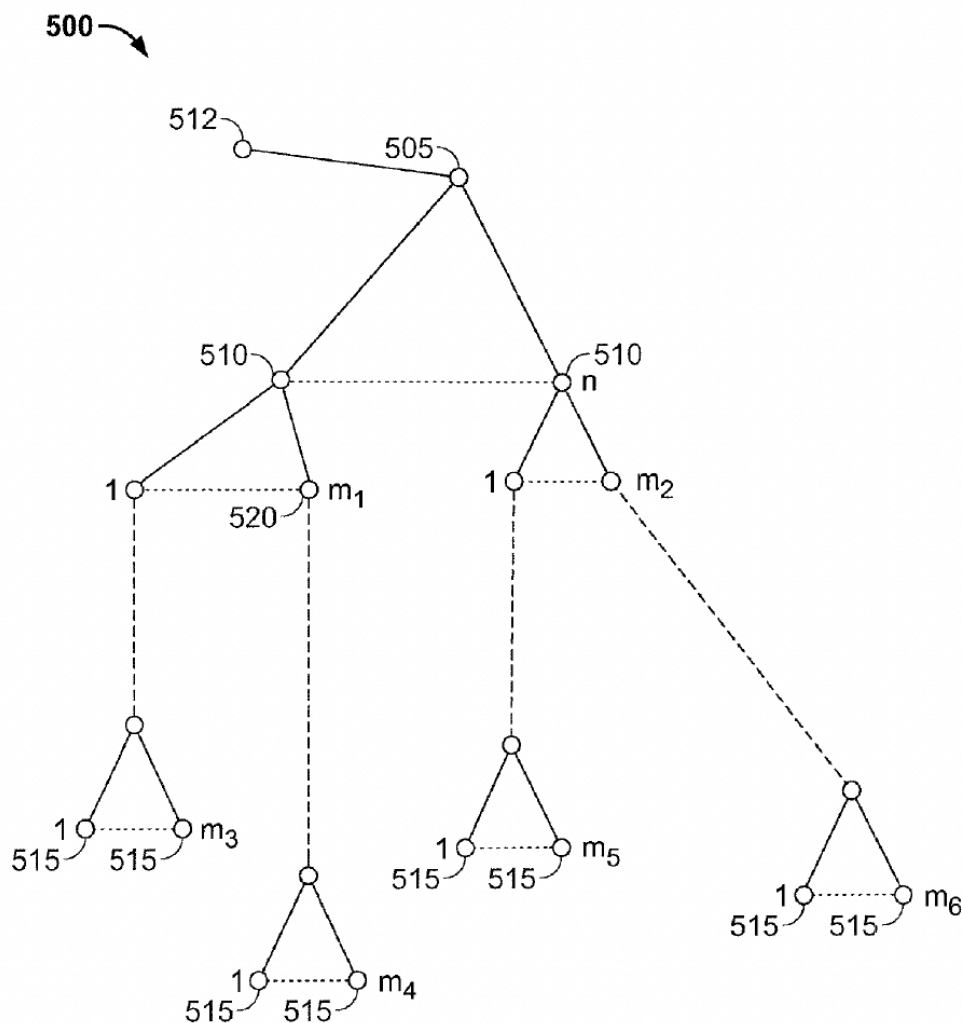


**FIG. 5**  
**(Prior Art)**

Young, Fig. 5.

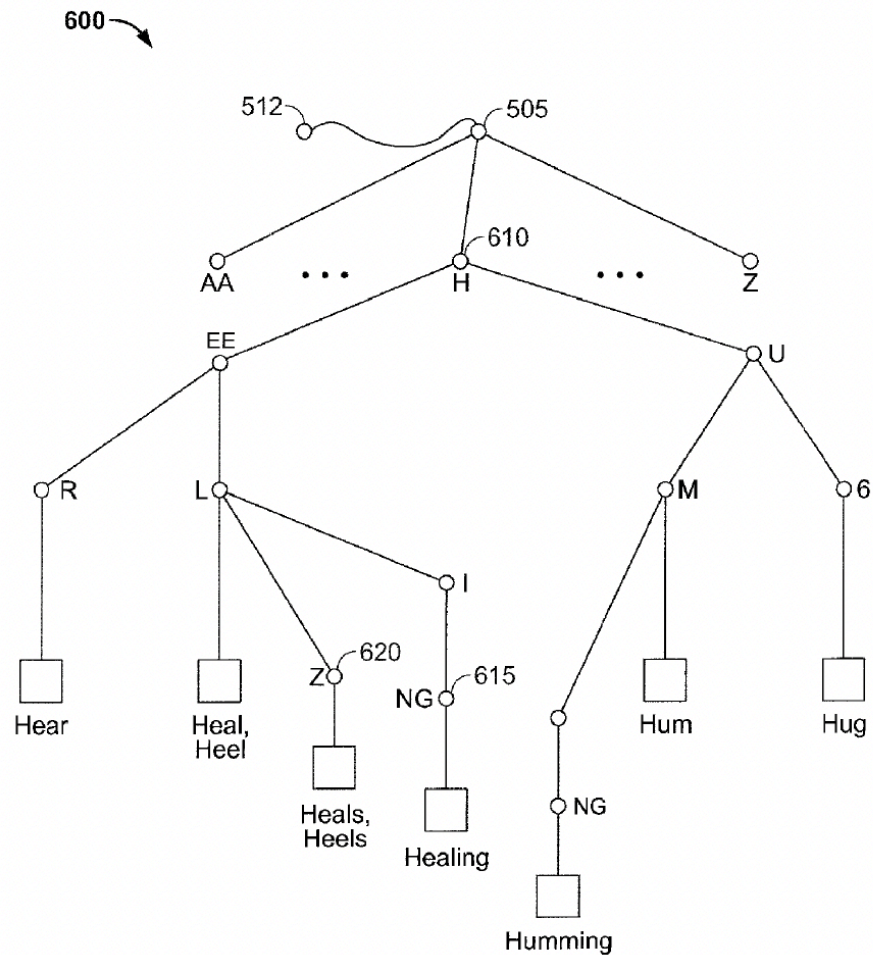


FIG. 6  
(Prior Art)

Young, Fig. 6.

## B. Feature Vectors

35. The '789 Patent describes feature vectors as comprising “a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame[.]” ’789 Patent, 6:2-4; see also *id.*, 6:27-29, 6:42-45, 6:55-58, 6:66-7:1.

Using feature vectors in speech recognition systems was well known before the

priority date of the '789 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.

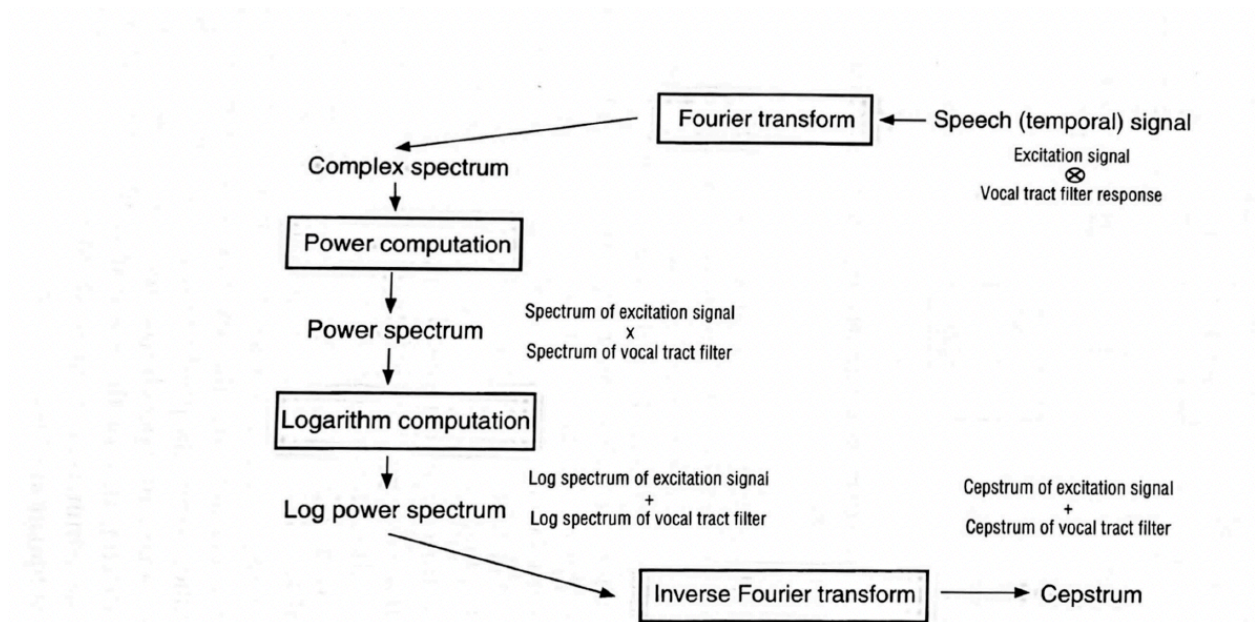


**Figure 3.21** Relationships among a speech signal, a vocal tract filter, and a vocal source excitation signal.

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

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. *Rabiner89* discloses in detail the process of calculating observation vectors  $O$  (i.e., feature vectors) from speech samples (i.e., a digital audio stream) by performing a linear predictive coding (LPC) “front end spectral analysis.” *Rabiner89* (Ex. 1022), 21. In particular, “[t]he overall system is a block processing model in which a frame of  $N_A$  samples is processed and a vector of features  $O_i$  is computed.” *Id.*

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 into the microphone 51 based on such as the Hidden Markov model (HMM) law.

---

<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.

*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 '789 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.” '789 Patent, 13:22-26.

42. *Stubbley* 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.

*Stubbley* (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 '789 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 *Stubley*. *Stubley* teaches “comparing a series of observations representing unknown speech, to stored models representing known speech[.]” *Stubley*, 1:5-10, 1:60-62. *Stubley* teaches that “the models comprise representations of elements of speech[.]” *Stubley*,

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 '789 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.

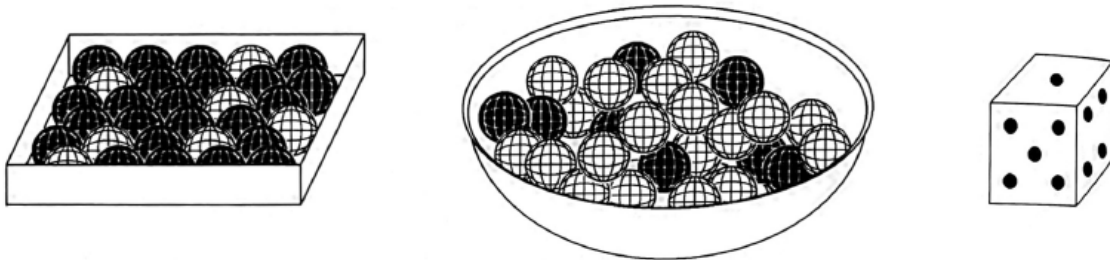


**Figure 7.10.** A box and a bowl full of colored balls illustrate a Hidden Markov Model.

*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.

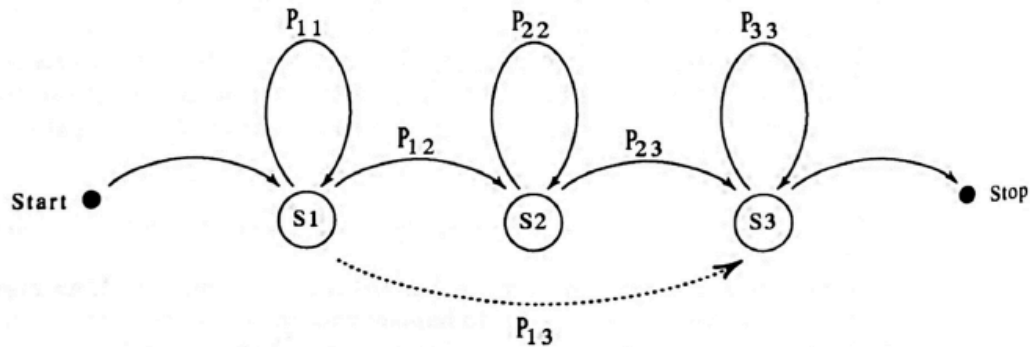


Figure 7.11. A Markov Model.

*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 '789 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 '789 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 '789 Patent. *Dudemaine* teaches that a Mahalanobis distance “gives a representation of the probability of a particular feature vector matching a given transition in the finite state model of the word in the [] dictionary.” *Dudemaine* (Ex. 1030), 8:44-47. In another example, *Ittycheriah* '447 defines a Mahalanobis distance. *Ittycheriah* '447, 6:60-64 (“[A] distance measure (e.g. along a Mahalanobis distance which is a weighted Euclidian distance in which weights are assigned according to the inverse of the variances of each of the dimensions or parameters of the feature vector[.]”). Additionally, *Kosaka* teaches the “Mahalanbois’ [*sic*] distance between the input pattern and the reference

pattern for each class was calculated, and the vowel class giving the minimum distance is selected as the recognition result.” *Kosaka* (Ex. 1031), 23:3-6.

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 '789 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 '789 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).**

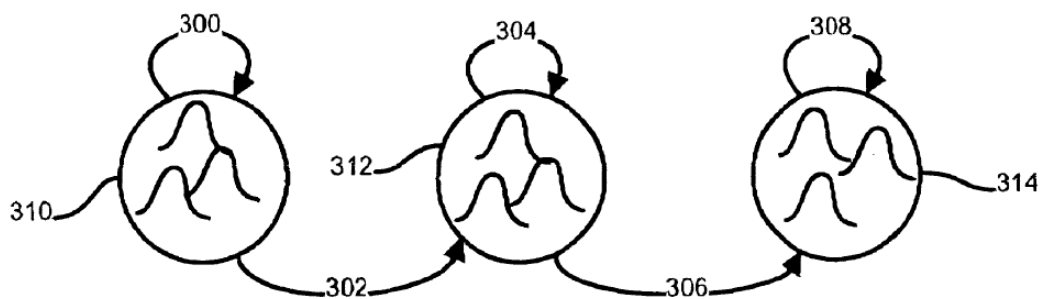
*Peters* (Ex. 1011), 1:35-48. *Peters* further teaches states of Hidden Markov Models associated with parameters that define Gaussian mixture of probability density functions.

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 '789 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 '789 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.

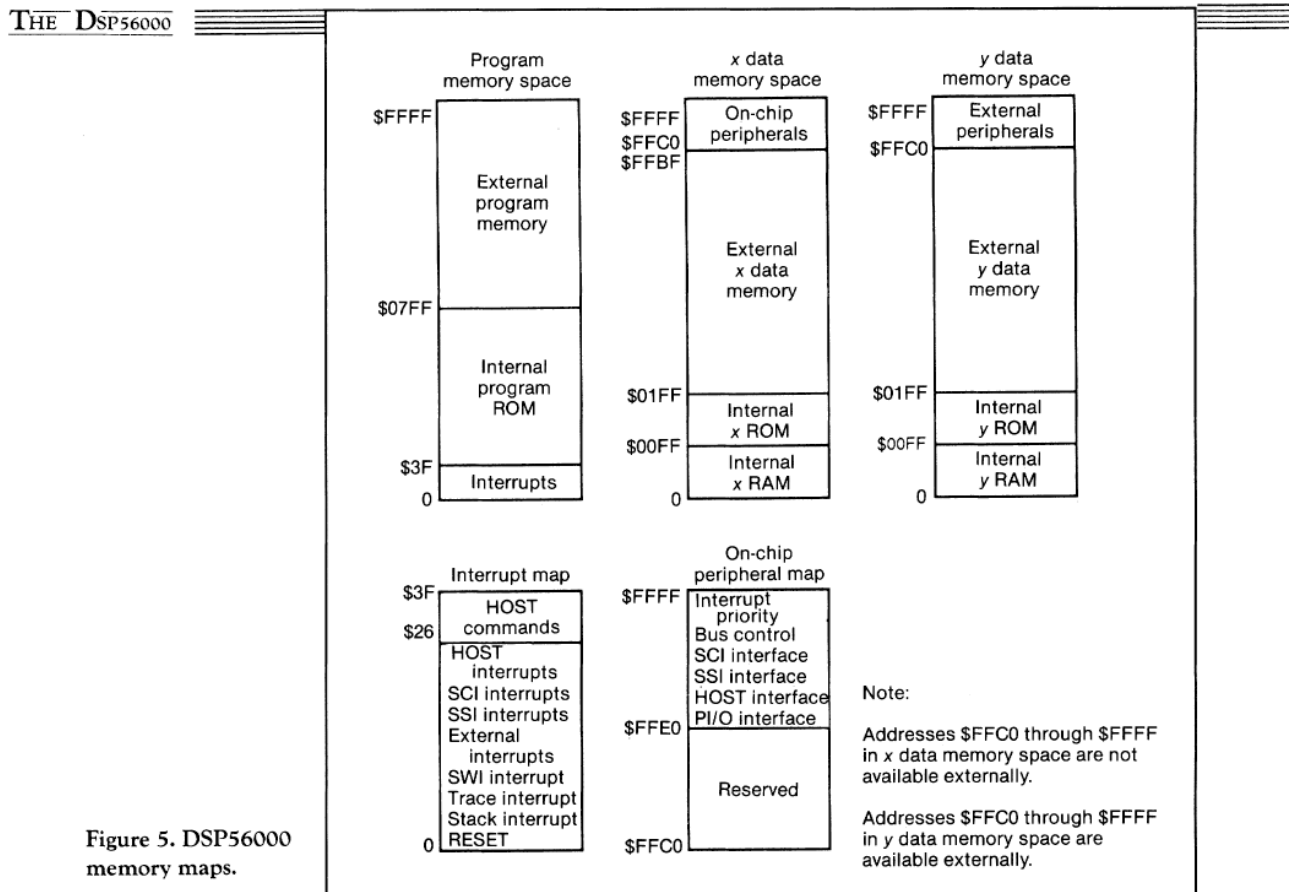


Figure 5. DSP56000 memory maps.

*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 ’789 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.

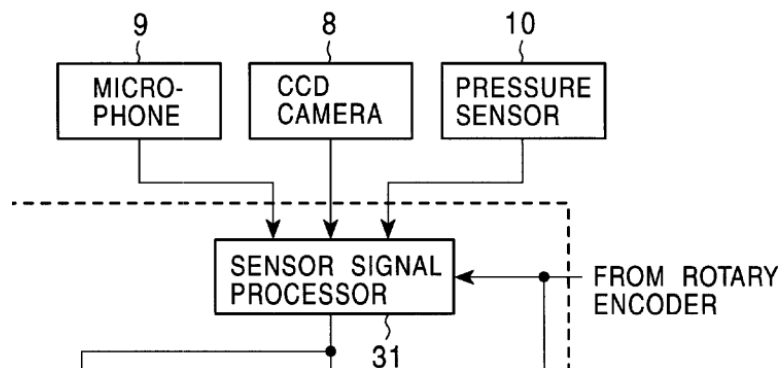
74. *Toyoda* teaches a voice recognition module (sensor signal processor 31) that uses an HMM. *Toyoda*, 7:48-50. Furthermore, *Toyoda* teaches referencing an acoustic model memory that stores an acoustic model representing phonemes while a matching unit 42 recognizes voice input.

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.

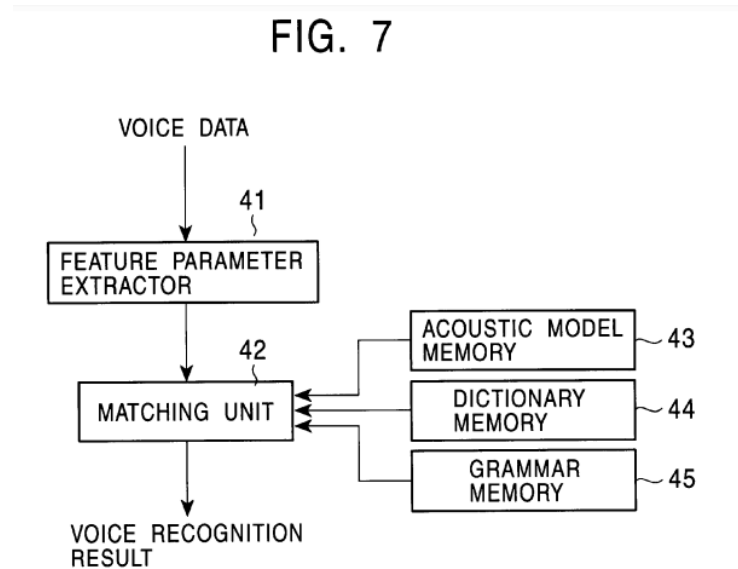
The acoustic model memory 43 stores an acoustic model that represents an acoustic feature such as phonemes and syllables in a voice to be recognized. Since voice recognition is here carried out using the continuous HMM, an HMM is employed.

*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 '789 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).

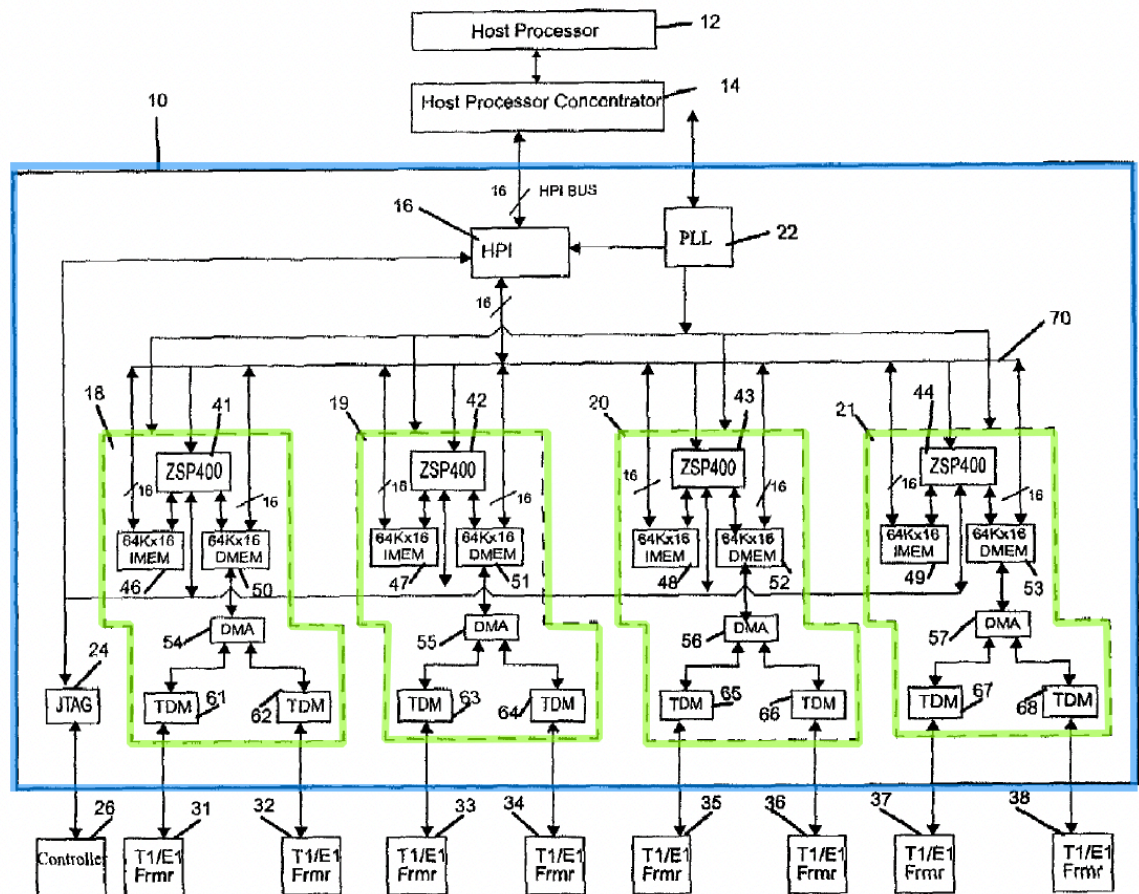


FIGURE 1

*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 '789 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.

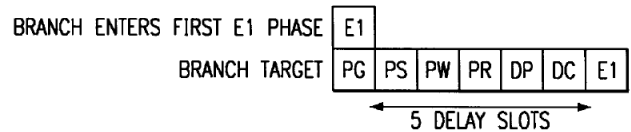
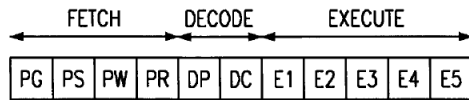
78. One example of pipelining data in general is *Luick*. For example, *Luick* teaches embodiments including Arithmetic Logic Units (ALUs) in a pipelined processor.” *Luick*, 1:5-10. *Luick* teaches:

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:



| 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 grammar processor 130, and a traceback processor 135. The

---

<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.

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 '789 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 '789 PATENT**

91. I have reviewed the '789 Patent (Ex. 1001). Descriptions of the '789 Patent that may be pertinent to my analysis are provided herein, and my descriptions of the '789 Patent are intended only as a summary of the '789 Patent and/or intended as being pertinent to the Challenged Claims of the '789 Patent. My opinions provided herein are not intended as a discussion of my entire understanding of the '789 Patent or any other claims in any related patent to the '789 Patent. The '789 Patent describes a speech recognition system comprising an acoustic coprocessor for performing steps of the speech recognition process. '789 Patent, 5:67-6:9, 6:24-37, Fig. 16, Claim 1. The '789 Patent describes an acoustic coprocessor comprises a first interface for receiving at least one feature vector, a calculating apparatus that calculates distances indicating a similarity between a feature vector and a plurality of acoustic vectors, and a second interface for sending at least one distance. '789 Patent, 6:61-7:5, 12:58-65, 13:4-38, 24:14-20, Claim 1.

## VII. OVERVIEW OF THE PRIOR ART

### A. Overview of Smyth

92. *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* teaches embodiments utilizing a combination of multiple DSPs and a processor, and three separate DSPs for components such as feature extractor 33, classifying processor 341, and parsing processor 351. *Smyth*, 5:49-53, 6:5-8, 6:47-51.

93. 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 ’789 Patent. For example, *compare Smyth*, 5:20-46, 5:58-6:4, 6:36-63 *with the ’789 Patent*, 1:20-21, 12:58-65, 13:3-38, 24:14-20. It is also my opinion that *Smyth* is reasonably pertinent to a problem faced by the inventors of the ’789 Patent because both recognize challenges with performing speech recognition tasks while using a limited set of computing resources.

### B. Overview of Mozer

94. In my opinion, *Mozer* teaches a speech recognition system that utilizes an audio recognition peripheral integrated circuit and calculates feature vectors and

distances between feature vectors and template vector. Therefore, *Mozer* (Ex. 1046) is in the same field of endeavor as the '789 Patent. *Compare, Mozer*, Abstract, 1:6-24 *with '789 Patent*, Abstract, 1:20-23. Further, because *Mozer* is directed to a problem of performing speech recognition, *Mozer* is reasonably pertinent to a problem faced by the inventors of the '789 Patent. *Compare Mozer*, 7:65-8:9, 8:58-66, 9:22-26, 9:38-42, 10:17-20 *with '789 Patent*, 1:20-33, 2:24-27.

### C. Overview of Nguyen

95. *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.

96. 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 '789 Patent. *Compare, Nguyen*, Abstract, 1:1-26 *with '789 Patent*, Abstract, 1:20-23. 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 '789 Patent. *Compare Nguyen*, 1:8-50, 2:8-4:36 *with '789 Patent*, 1:20-33, 2:24-27.

**D. Overview of Houser**

97. In my opinion, *Houser* teaches a speech recognition system. Therefore, *Houser* is in the same field of endeavor as the '789 Patent. *Compare, Houser*, Abstract, 1:6-12 *with '789 Patent*, Abstract, 1:20-23. Further, because *Houser* is directed to a problem of performing speech recognition, *Houser* is reasonably pertinent to a problem faced by the inventors of the '789 Patent. *Compare Houser*, 2:20-39, 15:33-34 *with '789 Patent*, 1:20-33, 2:24-27.

**E. Overview of Vensko**

98. In my opinion, *Vensko* teaches a speech recognition system that utilizes a DMA direct memory access controller. Therefore, *Vensko* is in the same field of endeavor as the '789 Patent. *Compare, Vensko*, Abstract, 1:6-11 *with '789 Patent*, Abstract, 1:20-23. Further, because *Vensko* is directed to a problem of performing speech recognition, *Vensko* is reasonably pertinent to a problem faced by the inventors of the '789 Patent. *Compare Vensko*, 1:64-2:37, 6:22-33, 10:10-19, *with '789 Patent*, 1:20-33, 2:24-27.

**F. Overview of Comeau**

99. In my opinion, *Comeau* teaches a speech recognition system that utilizes an audio recognition peripheral integrated circuit and calculates feature vectors and distances between feature vectors and template vector. Therefore, *Comeau* is in the same field of endeavor as the '789 Patent. *Compare, Comeau*, Abstract, [0002], [0274] *with '789 Patent*, Abstract, 1:20-23. Further, because

*Comeau* is directed to a problem of performing speech recognition, *Comeau* is reasonably pertinent to a problem faced by the inventors of the '789 Patent. *Compare Comeau*, [0153]-[0161], [0151]-[0152], [0168], [0181]-[0182], [0342] *with '789 Patent*, 1:20-33, 2:24-27.

#### **G. Overview of Boike**

100. *Boike* teaches multiple digital signal processors (DSPs) on a single integrated circuit (IC). *Compare '789 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 '789 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 '789 Patent. *Compare '789 Patent*, 24:37-42 *with Boike*, 2:21-35, 2:52-63.

#### **H. Overview of Whittaker**

101. In my opinion, *Whittaker* teaches a speech recognition system. Therefore, *Whittaker* is in the same field of endeavor as the '789 Patent. *Compare, Whittaker*, Abstract, 1:8-10, 2:20-31 *with '789 Patent*, Abstract, 1:20-23. Further, because *Whittaker* is directed to a problem of performing speech recognition, *Whittaker* is reasonably pertinent to a problem faced by the inventors of the '789

Patent. *Compare Whittaker*, 2:20-28, 2:33-44, 3:8-15, 3:27-32 with '789 Patent, 1:20-33, 2:24-27.

#### **I. Overview of Ichikawa**

102. In my opinion, *Ichikawa* teaches a speech recognition system that utilizes an audio recognition peripheral integrated circuit and calculates feature vectors and distances between feature vectors and template vector. Therefore, *Ichikawa* is in the same field of endeavor as the '789 Patent. *Compare, Ichikawa*, Abstract, 1:8-10 with '789 Patent, Abstract, 1:20-23. Further, because *Ichikawa* is directed to a problem of performing speech recognition, *Ichikawa* is reasonably pertinent to a problem faced by the inventors of the '789 Patent. *Compare Ichikawa*, 1:8-10, 1:28-29, 1:40-42, 3:63-66, 8:66-67, 20:65-67, 22:11-22, Figs. 17, 18 with '789 Patent, 1:20-33, 2:24-27.

#### **J. Overview of Seshadri**

103. In my opinion, *Seshadri* teaches a speech recognition system that utilizes an audio recognition peripheral integrated circuit and calculates feature vectors and distances between feature vectors and template vector. Therefore, *Seshadri* is in the same field of endeavor as the '789 Patent. *Compare, Seshadri*, Abstract, 1:18-27 with '789 Patent, Abstract, 1:20-23. Further, because *Seshadri* is directed to a problem of performing speech recognition, *Seshadri* is reasonably

Declaration of Chris Schmandt  
Patent No. 10,839,789

pertinent to a problem faced by the inventors of the '789 Patent. *Compare Seshadri*, 3:22-24, 3:28-30, 3:34-36, 3:54-65, *with '789 Patent*, 1:20-33, 2:24-27.

**K. Overview of Steinbiss**

104. In my opinion, *Steinbiss* teaches a speech recognition system that utilizes an audio recognition peripheral integrated circuit and calculates feature vectors and distances between feature vectors and template vector. Therefore, *Steinbiss* is in the same field of endeavor as the '789 Patent. *Compare, Steinbiss*, Abstract, [0001]-[0002], [0025]-[0026], [0033] *with '789 Patent*, Abstract, 1:20-23. Further, because *Steinbiss* is directed to a problem of performing speech recognition, *Steinbiss* is reasonably pertinent to a problem faced by the inventors of the '789 Patent. *Compare Steinbiss*, [0001]-[0002], [0025]-[0026], [0033] *with '789 Patent*, 1:20-33, 2:24-27.

**VIII. PROPOSED GROUNDS OF UNPATENTABILITY**

105. I have reproduced the table of the Proposed Grounds of Unpatentability from the '789 Petition for ease of reference.

| <b>Proposed Grounds of Unpatentability</b> |                                                                                                                                                                         |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b><u>Ground 1:</u></b>                    | Claims 1, 10-11, 14, 17 are <b>obvious</b> under § 103(a) over <i>Smyth</i> and <i>Mozer</i>                                                                            |
| <b><u>Ground 2</u></b>                     | Claims 2, 4, 6-9, 12, 16, 18, and 20-21 are <b>obvious</b> under § 103(a) over <i>Smyth</i> , <i>Mozer</i> , and <i>Nguyen</i>                                          |
| <b><u>Ground 3</u></b>                     | Claim 13 is <b>obvious</b> under § 103(a) over <i>Smyth</i> , <i>Mozer</i> , and <i>Houser</i>                                                                          |
| <b><u>Ground 4</u></b>                     | Claims 22-24 are <b>obvious</b> under § 103(a) over <i>Smyth</i> , <i>Mozer</i> , <i>Nguyen</i> , and <i>Vensko</i>                                                     |
| <b><u>Ground 5</u></b>                     | Claims 25-28 and 37 is <b>obvious</b> under § 103(a) over <i>Smyth</i> , <i>Mozer</i> , <i>Nguyen</i> , <i>Vensko</i> , and <i>Comeau</i>                               |
| <b><u>Ground 6</u></b>                     | Claim 29 is <b>obvious</b> under § 103(a) over <i>Smyth</i> , <i>Mozer</i> , <i>Nguyen</i> , <i>Vensko</i> , <i>Comeau</i> , and <i>Boike</i>                           |
| <b><u>Ground 7</u></b>                     | Claim 35 is <b>obvious</b> under § 103(a) over <i>Smyth</i> , <i>Mozer</i> , <i>Nguyen</i> , <i>Vensko</i> , <i>Comeau</i> , and <i>Whittaker</i>                       |
| <b><u>Ground 8</u></b>                     | Claims 38-39 are <b>obvious</b> under § 103(a) over <i>Smyth</i> , <i>Mozer</i> , <i>Nguyen</i> , <i>Vensko</i> , <i>Comeau</i> , and <i>Ichikawa</i>                   |
| <b><u>Ground 9</u></b>                     | Claims 42-44 are <b>obvious</b> under § 103(a) over <i>Smyth</i> , <i>Mozer</i> , <i>Nguyen</i> , <i>Vensko</i> , <i>Comeau</i> , <i>Ichikawa</i> , and <i>Seshadri</i> |
| <b><u>Ground 10</u></b>                    | Claim 45 is <b>obvious</b> under § 103(a) over <i>Smyth</i> , <i>Mozer</i> , <i>Nguyen</i> , <i>Vensko</i> , <i>Comeau</i> , <i>Ichikawa</i> , and <i>Steinbiss</i>     |

**IX. OPINIONS REGARDING GROUND 1: CLAIMS 1, 10-11, 14, AND 17 ARE OBVIOUS OVER SMYTH AND MOZER**

**A. Independent Claim 1**

**1. Claim 1[Pre]: “An acoustic coprocessor for processing data associated with an audio signal, comprising:”**

106. It is my opinion that *Smyth* teaches a DSP that “process[es] data associated with an audio signal,” as claimed. Specifically, *Smyth* teaches receiving speech in digital form and generating frames of contiguous digital samples. *Smyth*, 5:1-7. In my opinion, *Smyth*’s speech that is in digital form and partitioned into digital samples is the claimed “audio signal.”

107. 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.

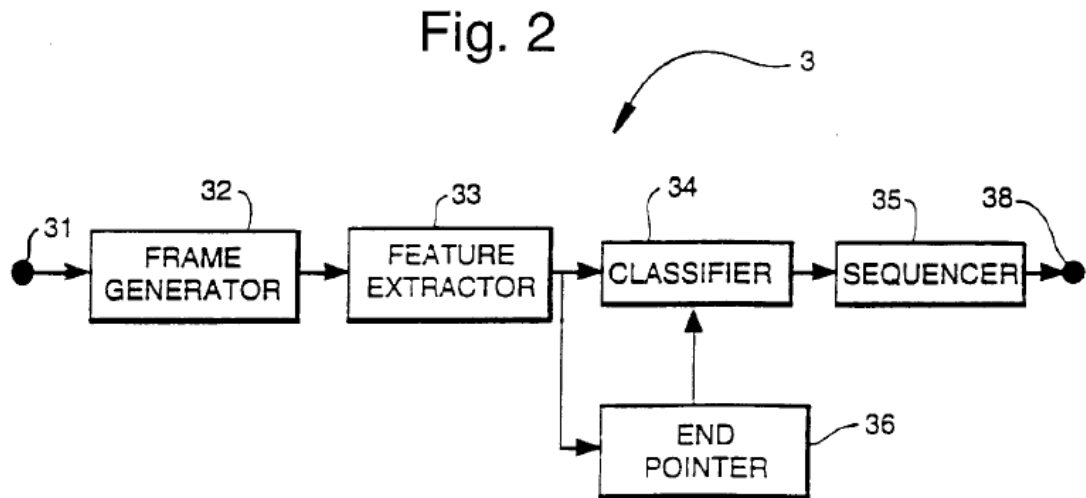
*Smyth*, 1:24-39. Also, as I discussed above, it was a well-known and common practice to transduce analog speech into a series of digital signal samples. *See* Section V.A (discussing *Klevans* teaches “the sound wave created by an individual’s speech is transduced into an analog electrical signal via 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.

108. Regarding the claimed “data associated with an audio signal,” 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, 5:6-7 (“feature extractor 33 for generating from a frame of samples a corresponding feature vector[.]”).

109. 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*, 5:31-46, 1:24-28. Specifically, *Smyth* teaches that feature extractor 33 “generat[es] from a frame

of samples a corresponding feature vector[.]” *Smyth*, 5:1-7. Therefore, in my opinion, *Smyth* teaches the claimed data associated with an audio signal, namely, extracted feature vectors.



*Smyth*, Fig. 2.

110. 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 '789 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.

'789 Patent, 12:58-65.

111. 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 '789 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; '789 Patent, 13:22-26). 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).

112. In my opinion, *Smyth* teaches feature vectors may "comprise cepstral coefficients" or "differential values of such coefficients[.]" *Smyth*, 5:31-38. As I discussed above in detail, cepstral coefficients are calculated from a digital audio stream. *See* Section V.B. Thus, in my opinion, a POSITA would have understood that *Smyth* teaches the DSP (which includes feature extractor 33) is programmed to generate (i.e., calculate) a feature vector including cepstral coefficients or differential values of such coefficients from the succession of digital samples (i.e., digital audio stream). *Smyth*, 5:3-5 ("a frame processor 32 for partitioning the succession of digital samples into a succession of frames of contiguous samples[.]"), 5:31-46 (disclosing feature extractor 33 generating feature vectors).

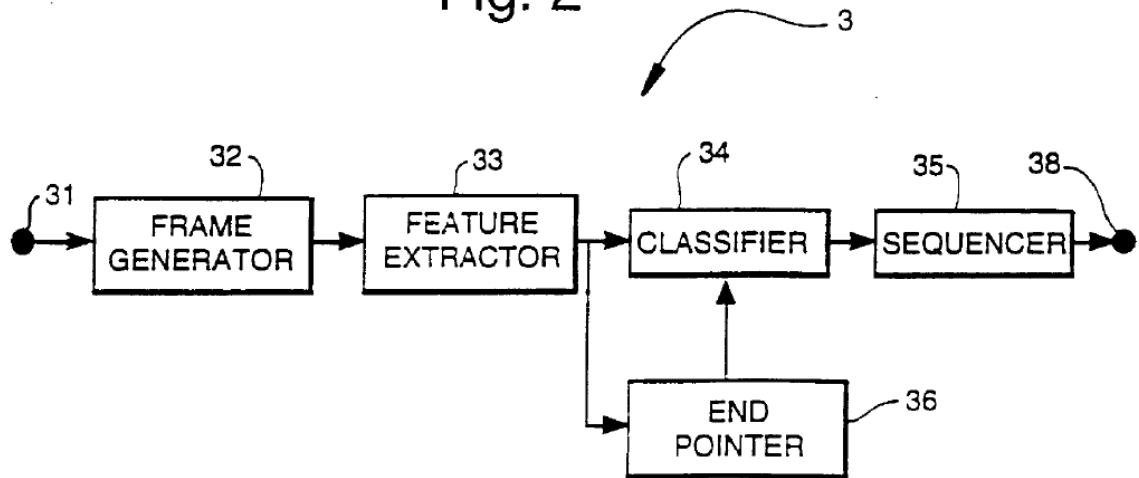
113. In my opinion, *Smyth* teaches feature vectors that comprise a plurality of extracted quantities from a digital audio signal. Indeed, *Smyth*'s teachings of cepstral coefficients are representations of spectral components extracted from an audio signal. As I discussed above, cepstral coefficients are "the coefficients of the Fourier transform representation of the log magnitude spectrum[.]" *Rabiner*, 146-147; *see* V.B. I note that cepstral coefficients were commonly used to represent audio signals in a compact manner. *See* Section V.B. Additionally, I note that *Maes* teaches extracting feature vectors as "MEL cepstra, delta and delta-delta coefficients along with normalized log-energies" from speech input. *Maes*, 5:6-10; ¶ 39. Further, I note that I discussed *Katagiri*'s teachings of cepstral coefficients as being extracted from a speech signal. *See* ¶ 37. Therefore, in my opinion, *Smyth* teaches feature vectors that comprise a plurality of extracted quantities from the digital audio signal.

114. It is also my opinion that *Smyth* teaches that feature extractor 33 generates feature vectors and that the first derivative of cepstral coefficients is calculated. In particular, *Smyth* teaches that feature vectors "comprise ... differential values of such [cepstral] coefficients comprising, for each coefficient, the differences between the coefficient and the corresponding coefficient value in the preceding vector." *Smyth*, 5:33-41. Thus, in my opinion, *Smyth* teaches feature vectors generated in feature extractor 33 also comprise a plurality of derived quantities from a digital audio signal (discussed above).

115. 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, 1:24-32 (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.

116. Regarding the claimed “acoustic coprocessor,” it is my opinion that *Smyth*’s classifier 34 and its associated signal lines or other communication paths that are for transmitting/receiving data is mapped as the acoustic coprocessor. I note that *Smyth* teaches classifier 34 receives feature vectors from the feature extractor 33 and also classifier 34 calculates probabilities and outputs said probabilities to sequencer 35. *Smyth*, 5:6-9, 5:12-15, 6:1-4, Figs. 2-3.

Fig. 2

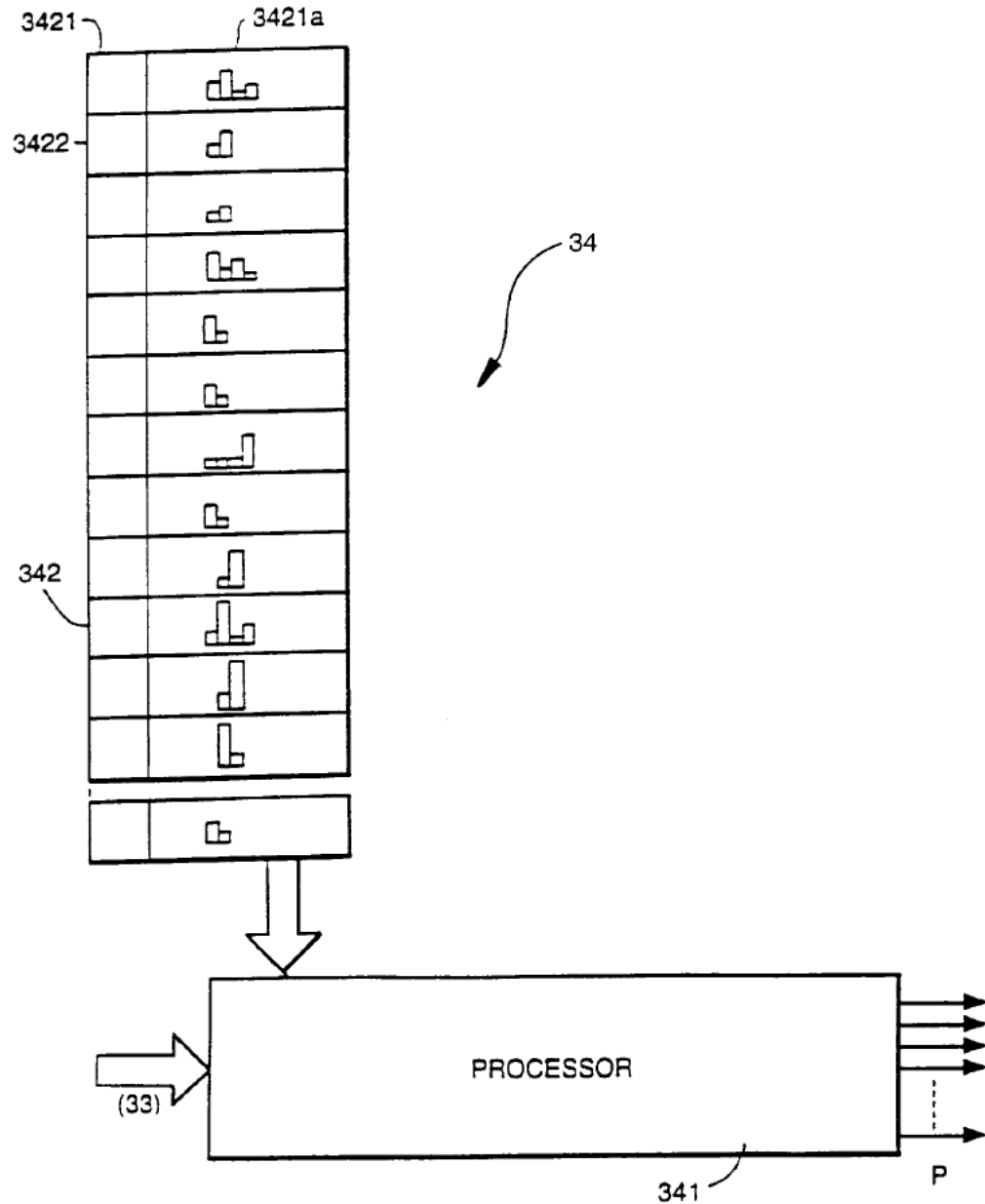


*Smyth*, Fig. 2.

117. In my opinion, feature extractor 33, classifier 34, and sequencer 35 are all communicatively coupled because data is transmitted from feature extractor 33 to classifier 34 and from classifier 34 to sequencer 35. *Smyth*, 5:6-18, 5:63-67, 6:1-4, Figs. 2-4 (where Fig. 2 depicts arrows between said components, Fig. 3 depicts an arrow from 33 entering processor 341 of classifier 34 and arrows P representing probabilities exiting processor 341, and Fig. 4 depicts probabilities (arrows) from classifier 34 entering state probability store of sequencer 35).

118. I further note that classifier 34 comprises a classifier processor 341 and a state memory 342. *Smyth*, 5:55-57, 4:7-8. Such is depicted in Fig. 3 below.

Fig. 3



*Smyth*, Fig. 3. Also, in my opinion, a POSITA would have understood *Smyth*'s disclosures of "classification processor 34" to refer to "classifier processor 341." In my opinion, "classification processor 34" should have read

“classification/classifier/classifying processor 341,” as element 34 refers to classifier 34 and element 341 refers to the processor of classifier 34. *Smyth*, 5:55-57, 5:12-13, 5:63, 6:1-8. Indeed, *Smyth* refers to classifier processor 341 and classifying processor 341 each several times, such that a POSITA would have understood both terms to be interchangeable with no difference in meaning. *Id.* Specifically, a POSITA would have understood “classifier 34 comprises a classifying processor 341” such that classifier processor 341 would have performed the calculating functions of classifier 34. *Smyth*, 5:12-13, 5:55-57, 5:63, 6:1-8.

119. 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 classifying 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 classifying 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.

120. In my opinion, *Smyth*’s teaching, 5:49-53 is informing the reader of suitable exemplary DSP models, such as the Motorola DSP 56000, 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.

121. 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. 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, 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 classification processor, and the parsing processor.

122. 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.

123. 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.”

124. 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 *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). 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.

125. *Smyth*, 6:36-42 teaches the parsing processor 351 reads probabilities output by classifier processor 341. I also note that both processors 341 and 351 have separate element numbers and have their own respective memories 342 and 352 (and state probability store 353), respectively. *Smyth*, Figs. 3-4.

126. In my opinion, *Smyth* teaches classifier 34 receives feature vectors from feature extractor 33 and sequencer 35 receives probabilities from classifier 34. A POSITA would have understood that such transmission of data (e.g., feature vectors, probabilities) would have included transmission along one or more signal lines or other data communication paths. *Smyth* teaches classifier 34 receives data and sequencer 35 receives data. *Smyth*, 5:7-8, 5:12-14, 6:1-4. Furthermore, I note that a POSITA would have understood that it was well known that DSPs had input/output

(I/O) ports for receiving and transmitting data. Specifically, one of the DSPs expressly taught by *Smyth* is disclosed in detail by *Kloker*, an article from Motorola, the manufacturer of the DSP56000. *Smyth*, 5:49-53, 6:5-8, 6:43-51; *Kloker* (Ex 1009), 3 (“A block diagram of the DSP56000 architecture is shown in Figure 2. The architecture consists of a core processor that executes the DSP56000 instruction set and other on-chip resources such as memory and peripherals that provide storage and I/O capability to the core processor.”), *Id.* (“Invariably, DSP users identify processor speed and system performance in a real-time, I/O-intensive environment as the most important factors in their designs.”). Annotated Fig. 2 below from *Kloker* depicts I/O ports of the DSP56000.

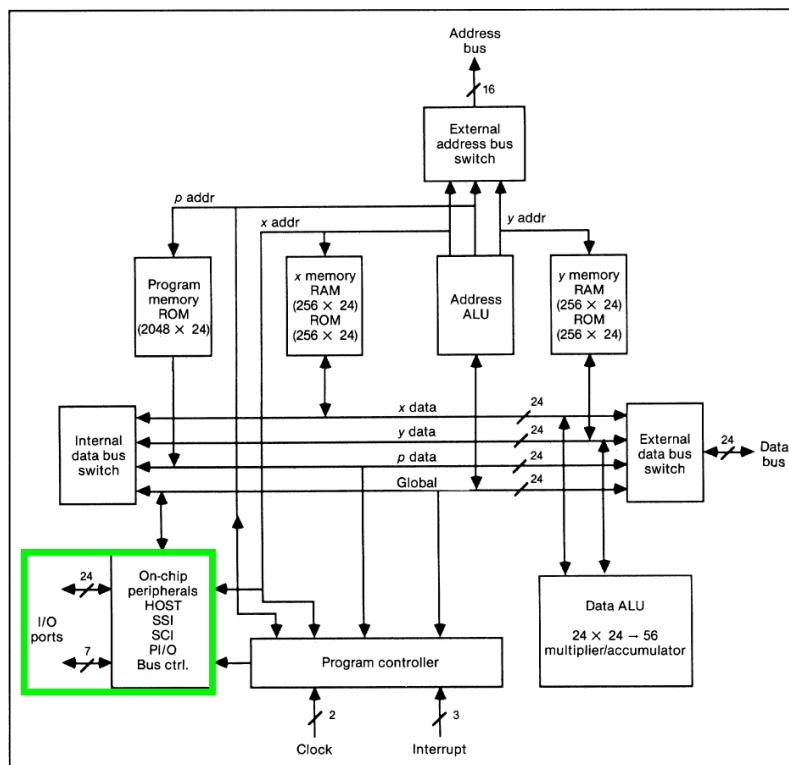


Figure 2. DSP56000 block diagram.

*Kloker*, 4 (“Figure 2. DSP56000 block diagram.”) (annotated). Additionally, *Kloker* teaches the “DSP56000 provides a large set of on-chip memory and I/O peripheral resources to support the core processor.” *Kloker*, 14. *Kloker* teaches “[m]icrocomputer-style I/O capability is provided by three on-chip peripherals—the parallel host MPU/DMA interface (host), the asynchronous serial communications interface (SCI), and the synchronous serial interface (SSI)—on 24 programmable, general-purpose port pins.” *Id.* I note Fig. 9 below depicts pins for I/O operations.

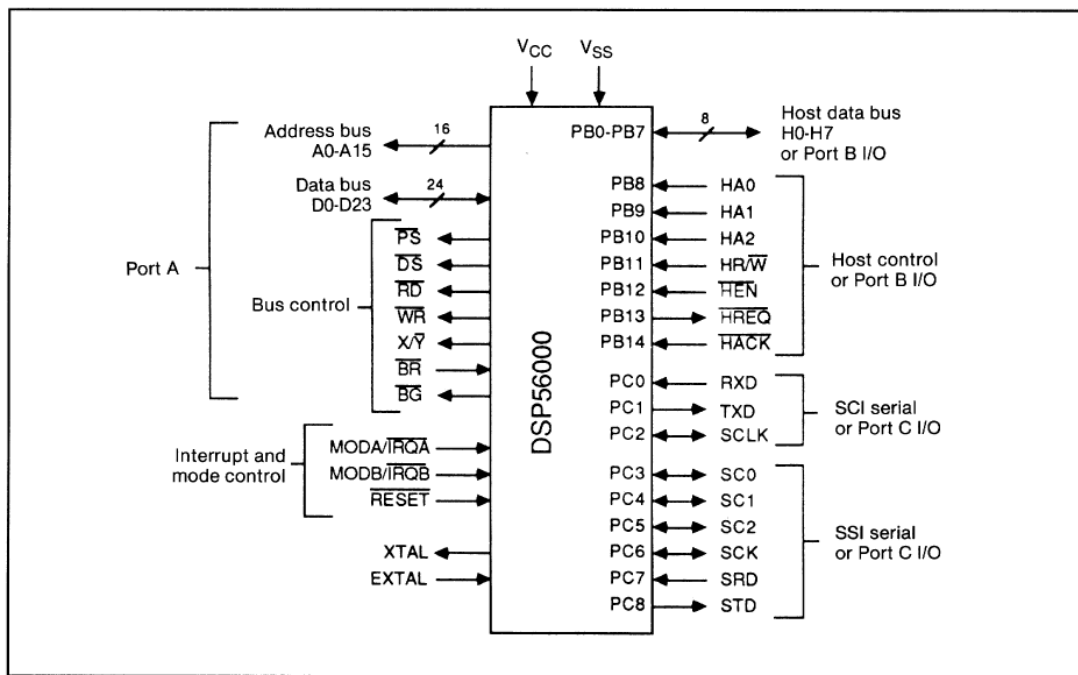
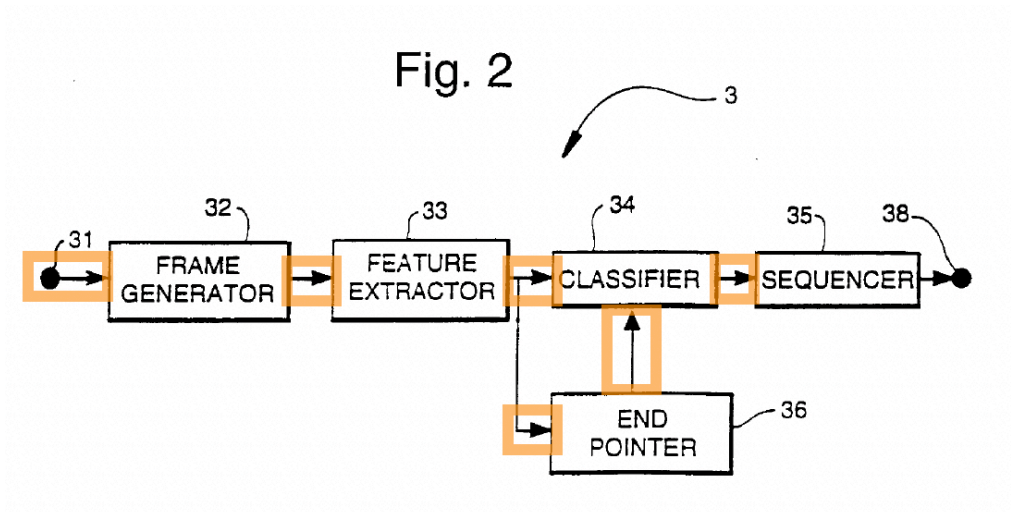


Figure 9. DSP56000 pinout.

*Kloker*, 14 (“Figure 9. DSP56000 pinout.”). See also *Kloker*, 15 (disclosing general purpose I/O pins).

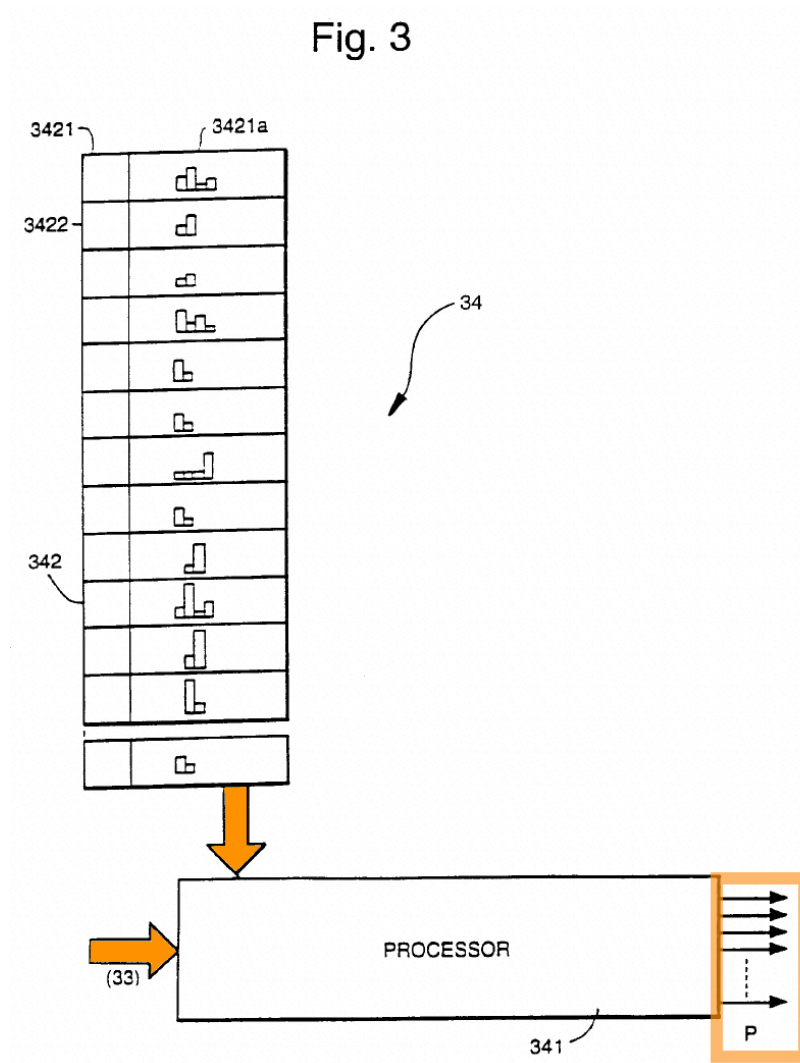
127. *Smyth* further illustrates data transmission between various components of recognition processor 3. I have reproduced annotated Fig. 2 below from the Petition. Annotated Fig. 2 depicts such data transmission.



*Smyth*, Fig. 2 (annotated).

128. *Smyth* teaches transmitting and receiving signals carrying data between various components of the speech recognition system and in particular the components of recognition processor 3. *Smyth*, 3:58-61, 4:38-45, Fig. 3. In particular, *Smyth* teaches data transmitted from memory 342 and feature extractor 33 to classifier processor 341, and transmitted out from classifier processor 341 (to state probability memory 353 of sequencer 35, as depicted in Fig. 4), as depicted in annotated Fig. 3 below, which I have reproduced from the Petition.

Fig. 3

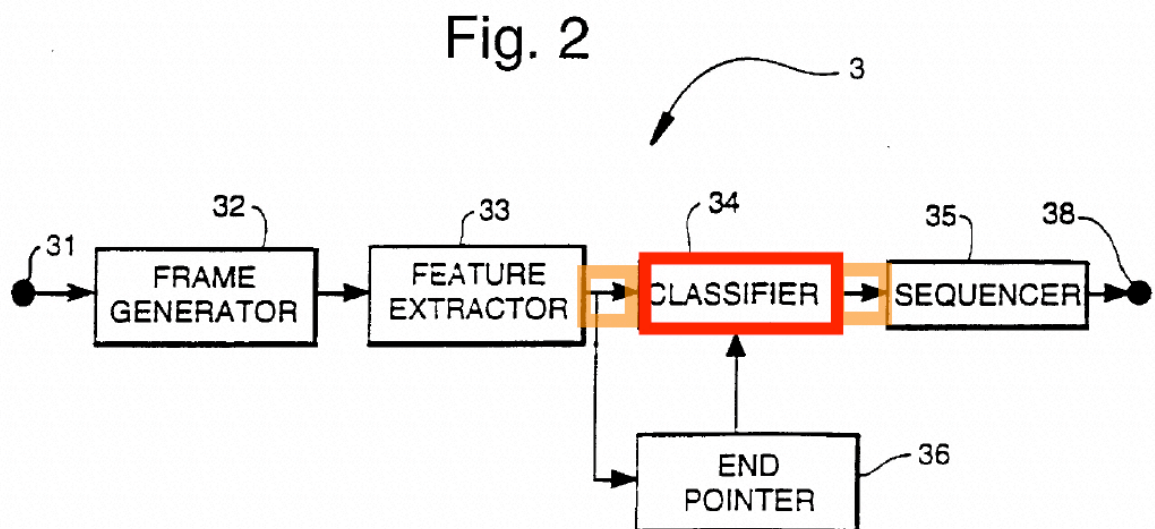


*Smyth*, Fig. 3 (annotated), *see also* Fig. 4.

129. In my opinion, classifier 34 is an acoustic coprocessor that is associated with at least a first signal line/communication path between the feature extractor 33 and classifier 34. Further, classifier 34 is associated with a second signal line/communication path between the classifier 34 and sequencer 35. Therefore, *Smyth* teaches an acoustic coprocessor that includes classifier 34 (which comprises classifier processor 341 and state memory 342) and the associated signal lines.

Furthermore, *Smyth's* collective classifier 34 and associated signal lines is a coprocessor, in my opinion, because classifier 34 calculates distances and performs other claimed functions by operating with feature extractor 33 and sequencer 35 (and other components) of speech recogniser 3. I note that *Smyth's* classifier 34 and associated signal lines are an “acoustic” coprocessor because said classifier 34 performs processing involving the input speech signal (e.g., feature vectors), which are acoustic data because said signal relates to the sounds of speech utterances. *Smyth*, 5:1-18.

130. As depicted in annotated Fig. 2 below, data enters classifier 34 and is transmitted to sequencer 35 as output. I have reproduced annotated Fig. 2 from the Petition.



*Smyth*, Fig. 2 (annotated);

131. Therefore, in my opinion, a POSITA would have understood or at least found obvious that *Smyth*'s classifier 34 is an acoustic coprocessor for processing feature vectors ("data associated with an audio signal"). *Smyth*, 5:63-67 ("Classifying processor 341" 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.").

2. ***Claim 1(a): "a first interface for receiving at least one feature vector, wherein the feature vector is determined from the audio signal"***

a) ***Smyth's Teachings***

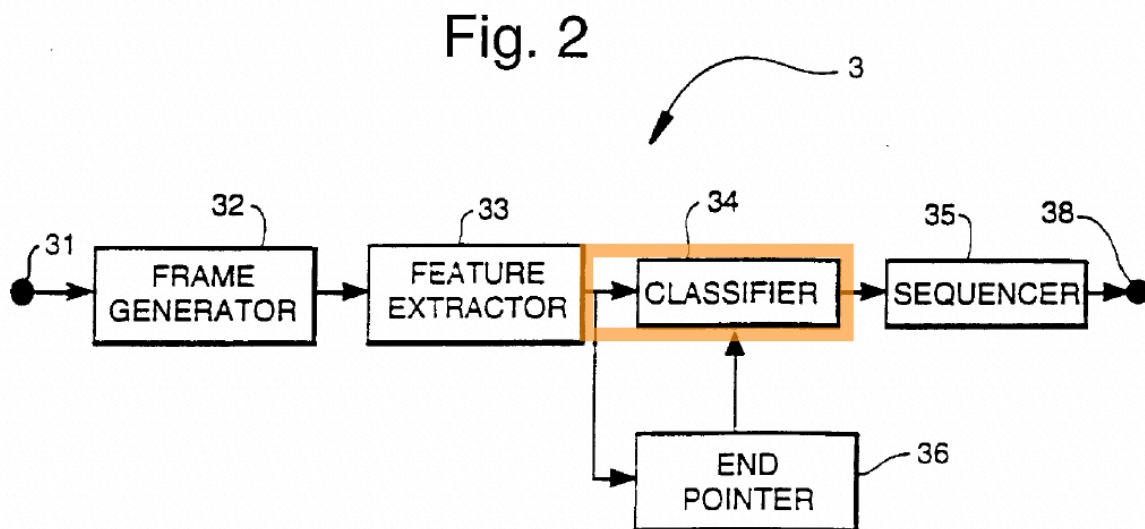
(1) **"wherein the feature vector is determined from the audio signal"**

132. As I opined above for Claim 1(Pre), 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*, 5:31-46, 1:24-32 (disclosing it was common in speech recognition to input digital speech data into a front-end processor to derive a more compact, perceptually significant set of data referred to as a feature set or vector and that speech was segmented into frames of length 10-20 ms and a set of coefficients

was calculated for each frame.). Specifically, *Smyth* teaches that feature extractor 33 “generat[es] from a frame of samples a corresponding feature vector[.]” *Smyth*, 5:1-7, Fig. 2. Therefore, in my opinion, *Smyth* teaches the claimed data associated with an audio signal, namely, feature vectors determined from a digital audio signal (the “succession of digital samples” in *Smyth*). *Id.*

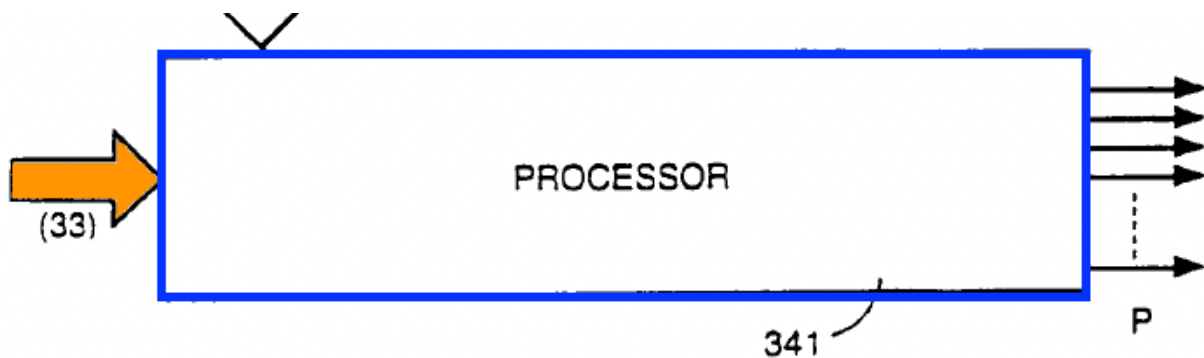
(2) **“a first interface for receiving at least one feature vector”**

133. In my opinion, *Smyth* teaches or at least renders obvious the claimed “a first interface for receiving at least one feature vector.” Specifically, *Smyth* teaches classifier 34 receives a “succession of feature vectors” from feature extractor 33. *Smyth*, 5:6-9. I also note that *Smyth*’s Fig. 2 depicts classifier 34 receiving said succession of feature vectors from feature extractor 33 as indicated by an arrow (annotated orange). I have reproduced annotated Fig. 2 from the Petition.



*Smyth*, Fig. 2 (annotated), 5:7-9. Therefore, in my opinion, *Smyth* teaches or at least renders obvious the collective classifier 34 and signal line/communication path between classifier 34 and feature extractor 33 is the claimed “first interface[.]” Specifically, I note that said collective classifier 34 and signal line/communication path receives at least one feature vector because feature vectors are transmitted from feature extractor 33 to classifier 34 via a signal line/communication path.

134. In more detail, *Smyth* teaches classifier 34 comprises classifier processor 341 and state memory 342. *Smyth*, 5:55-57. I have reproduced annotated Fig. 3 from the Petition below with accompanying description. Annotated Figure 3, below, depicts classifier processor 341 (annotated blue) receiving feature vectors, represented by an arrow (annotated orange), from feature extractor 33.



*Smyth*, Fig. 3 (excerpt, annotated). In my opinion, classifier 34 and the signal line/communication path between said classifier 34 and feature extractor 33 meets the claimed “interface” because *Smyth* teaches classifier processor 341 and state memory 342 receive feature vectors, perform processing on said feature vectors, and output probabilities P as depicted in Fig. 3.

135. Further, I note that the '789 Patent does not explicitly define “interface,” but through usage a POSITA would have understood the '789 Patent to describe standard computer interfaces used for communicating between computer components. The following are some illustrative examples of the '789 Patent's references to interfaces.

FIG. 17 shows a block diagram of a distance calculation accelerator 200, used in an embodiment of the invention. **A feature vector is received at input 212** of the accelerator 200.

**The feature vector is loaded into feature vector (FV) memory A 208 via a bus 211.**

*789 Patent*, 25:6-10. (emphasis added).

The distance engine may support the pass-through of data from the front-end to the search-stage. ... The data passed through may be for any purpose. Examples might include ... data from another input device such as a button or keyboard in a multi-modal **interface**.

*'789 Patent*, 5:34-48 (emphasis added).

“The distance engine may implement one or more additional pipeline stages to overcome delays introduced by low bandwidth, high latency, or conflicted bus **interfaces**.”

*'789 Patent*, 5:58-60 (emphasis added).

FIG. 11 shows a Language Model **interface** architecture.

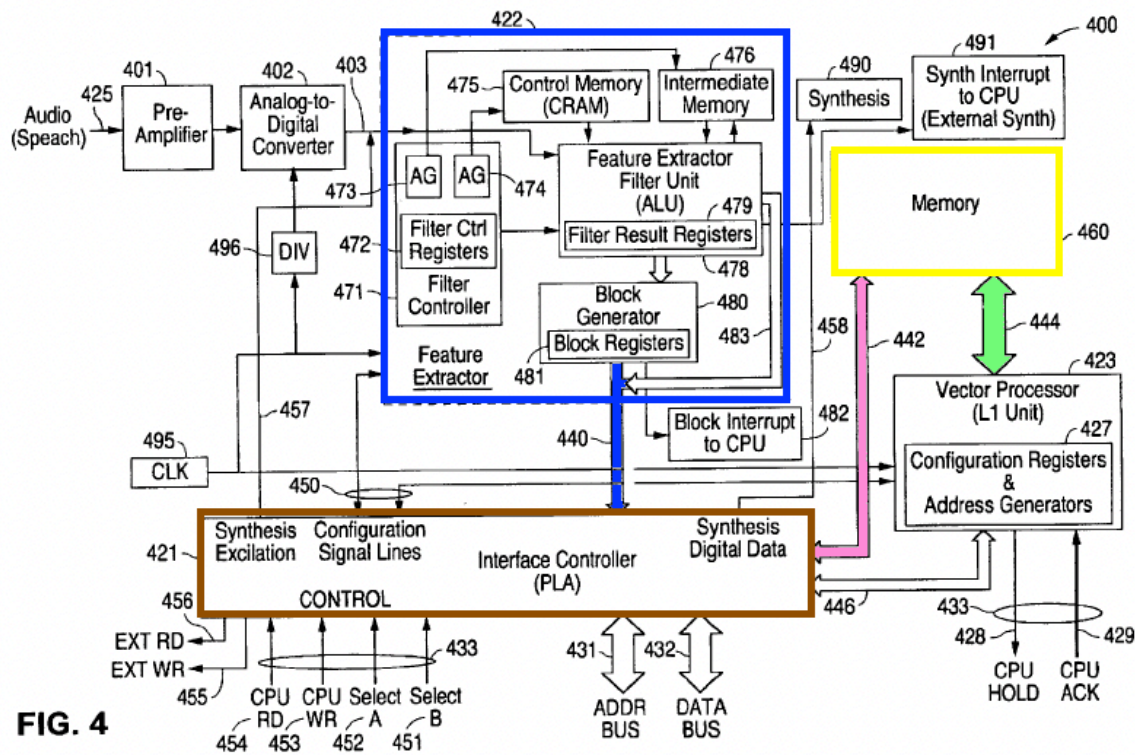
'789 Patent, 9:7 (emphasis added); *see also* '789 Patent, 22:24-26, 10:12-21, 10:37-40, 24:32-36, Fig. 11.

136. Further, in my opinion, classifier processor 341, which may be a DSP, includes inputs and outputs for receiving and transmitting data. *See* ¶ 126, above, discussing Motorola DSP56000. Classifier (including classifier processor 341) and its inputs/outputs also are collectively an interface or such is obvious, at least because the I/O ports serve as an interface for receiving/transmittin data to the classifier processor 341 (which may be a DSP). *Smyth*, 5:12-15, 5:63-67, 6:13-15, Figs 2-3. In my opinion, because classifier 34 receives and outputs data (i.e., feature vectors, probabilities) from and to other electronic components (e.g., feature extractor 33 and sequencer 35, respectively), a POSITA would have understood and found obvious that the collective classifier and at least one of a signal line or communication path between extractor and classifier or the input at classifier is or includes an interface or that such would have been obvious to a skilled artisan.

**b) *Mozer's* Teachings**

137. Regarding *Mozer's* teachings, in my opinion, to the extent Patent Owner argues that *Smyth* does not teach or at least render obvious the claimed “first interface[,]” it would have been obvious to a POSITA to modify *Smyth's* speech recognition system to include *Mozer's* signal lines. In my opinion, *Mozer* teaches a “first interface[,]” as claimed, for receiving at least one feature vector. Specifically,

*Mozer* teaches feature extractor output lines 440, (annotated blue) memory input lines 442, (annotated pink) and memory interface lines 444 (annotated green). I note that I have reproduced annotated Fig. 4 of *Mozer* from the Petition below.



*Mozer*, Fig. 4 (annotated).

138. I note element 442 is shown in *Mozer*, Fig. 4, but I do not see that it is described in the written description. I further note element 342 is referred to as “memory input lines.” *Mozer*, 6:64. In my opinion, a skilled artisan would have understood or at least found obvious that *Mozer*’s element 442, which is located in the same location as memory input lines 342 would have been memory input lines

442. Indeed, a POSITA would have understood or at least found obvious that *Mozer's* numbering scheme changes the first digit and maintains the last two digits for numerous elements. For example, *Mozer* teaches address bus 331 and address bus 431, data bus 332 and data bus 432, vector processor 323 and vector processor 423, feature extractor 322 and feature extractor 422. *Mozer*, Figs. 3-4. In my opinion, any one or all of *Mozer's* signal lines meets the claimed "first interface for receiving at least one feature vector" because said signal lines receive and transmit feature vectors.

139. In my opinion, a POSITA would have been motivated and found it obvious to modify *Smyth's* classifier 34 to include, utilize, or otherwise be associated with *Mozer's* signal lines between various electronic components, including *Smyth's* feature extractor 33, classifier 34, and sequencer 35. I note that *Smyth* already teaches classifier 34 includes classifier processor 341 that receives feature vectors from feature extractor 33, which indicates to a skilled artisan that classifier 34 receives feature vector information. *See* Claim 1(Pre) above. In my opinion, signal lines between various electronic components in computer systems are a routine design element, including as taught by *Mozer*. Further, prior art such as *Kloker* teaches I/O ports for transmitting information between components, such as between DSPs and other components (e.g., a host processor). Therefore, in my opinion, to the extent it is argued that *Smyth* does not teach or at least render obvious signal lines

between components and a claimed “interface,” a POSITA would have been motivated to modify *Smyth* by using a known technique of utilizing signal lines to transmit data between electronic components to improve *Smyth*’s classifier 34, feature extractor 33, and sequencer 35 in the same way as *Mozer* to provide transmission of data (e.g., feature vectors, calculated probabilities). I note that such a modification would have had a reasonable expectation of success because utilizing signal lines in such a manner was known in the prior art and routine. Further, for these reasons, such a modification to *Smyth* would not have required undue experimentation.

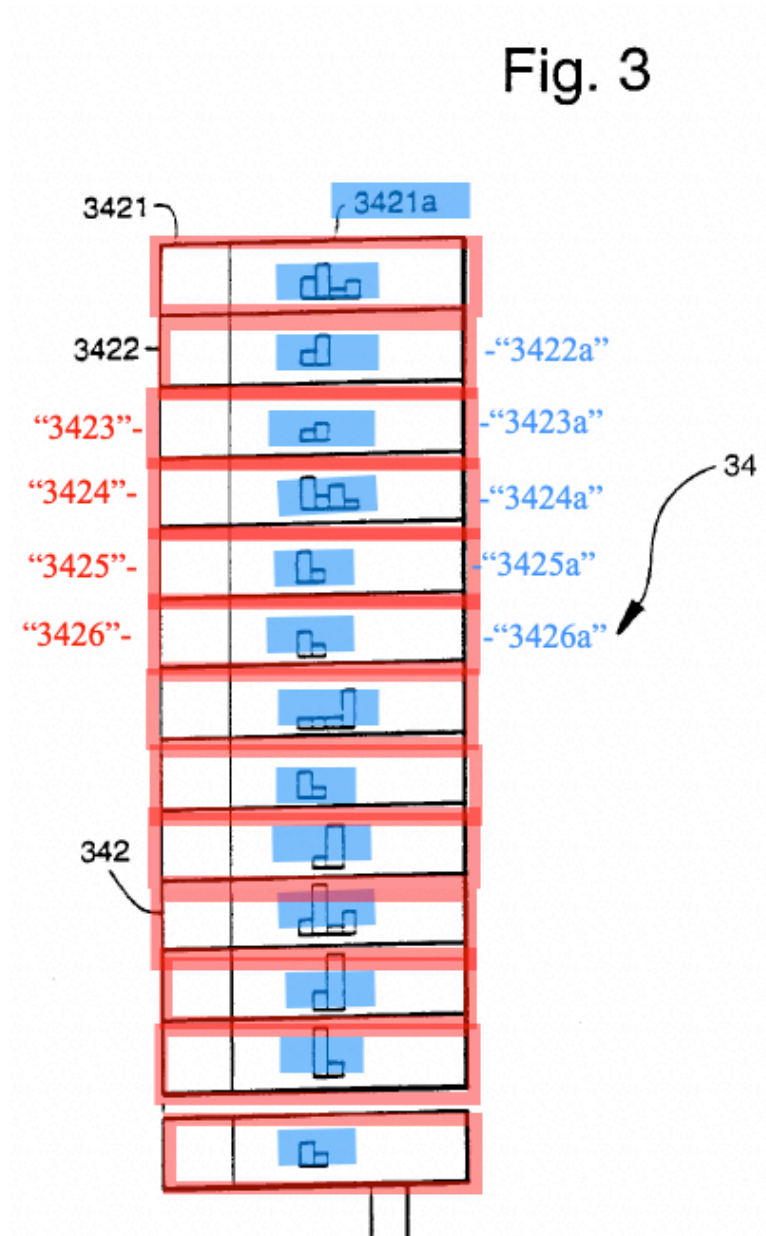
**3. Claim 1(b): “an acoustic model memory for storing an acoustic model defining a plurality of acoustic states;”**

140. The ’789 Patent describes examples of acoustic states, “including *states corresponding to [a] phone or group of phones[.]*” ’789 Patent, 7:29-34, 8:26-31. The ’789 Patent further describes acoustic models include HMM models and “alternative types of acoustic models, rather than HMM models.” ’789 Patent, 38:30-32.

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

141. 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>



<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.

142. 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.”). 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”).

143. 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. In my book, *Schmandt*, I

explained that “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* (Ex. 1010), 18. Therefore, because *Smyth*’s speech states relate to sounds of speech, a POSITA would have understood that such speech states are “acoustic states.”

144. Additionally, a POSITA would have understood that *Smyth*’s speech states meet the claimed “acoustic states” according to the ’789 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 ’789 Patent describes acoustic states as a “phone or group of phones[.]” ’789 *Patent*, 7:29-32 (“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[.]”).

145. *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.

146. 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. I also explained in my book 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 phonemes and phones are synonymous, a POSITA would have understood that *Smyth*’s speech states are “acoustic states” as claimed because *Smyth*’s speech states correspond to a “phone or group of phones.”

**b) “an acoustic model memory for storing an acoustic model defining a plurality of acoustic states”**

147. It is also my opinion that *Smyth* teaches or at least renders obvious an acoustic model and acoustic model memory, as claimed. In my opinion, *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. I note that *Smyth* also teaches classifier 34 receives “the succession of feature vectors and operat[es] on each with a plurality of model states, to generate recognition results[.]” *Smyth*, 5:6-9.

148. In my opinion, it was well known in speech recognition systems to generate templates or models of spoken input, such as taught by *Smyth*. *Smyth*, 1:34-39. For example, *Ney* (Ex. 1053) teaches estimating probabilities of acoustic vectors “during the training phase of the recognition system.” *Ney*, 4. Similarly, *Jiang* (Ex 1004), 11:25-29 (“[I]n order to train the system shown in FIG. 2, training words are spoken into system 60 by the user. The training words comprise a set of training data which is converted to digital samples by A/D converter 64 and then to codewords by feature extraction module 66.”); *Ravishankar* (Ex. 1012), 22, 33-34, 67-71 (discussing lexical trees and words in the vocabulary), 19-20, 27-28 (including discussing training acoustic models in subword units including phonemes).

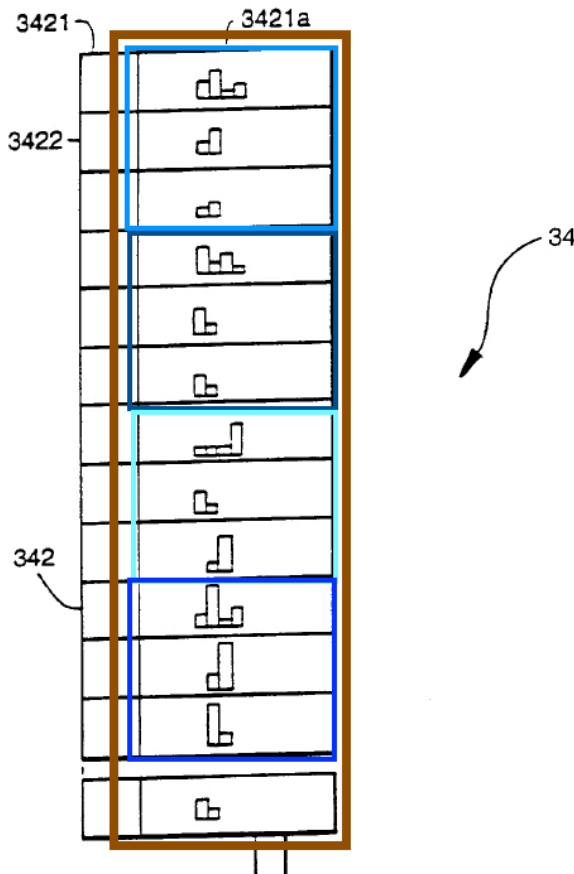
149. I note that a POSITA would have understood a set of speech states such as *Smyth*'s "plurality of model states" to be an acoustic model. *Smyth*, 5:58-62, 5:7-9. As I discussed above, a POSITA would have understood that *Smyth*'s speech states are acoustic states, as claimed. In my opinion, a POSITA would have understood that *Smyth*'s "plurality of model states" is an acoustic model, as claimed, and reference to the states (i.e., speech states) is used for determining the likelihood that a particular feature vector corresponds to one of the speech states. *Smyth*, 5:7-9 (disclosing classifier 34 "receiving the succession of feature vectors and operating on each with a plurality of model states, to generate recognition results[.]" ), 5:58-62 ("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."). As discussed below in my opinions of Claim 1(c), *Smyth* teaches calculating probabilities indicating the likelihood for a feature vector matching each state in state memory 342. *Smyth*, 5:63-6:4. Further, because *Smyth* teaches referencing a plurality of model states, *Smyth* teaches identifying allophones by calculating feature vector probabilities. *Smyth*, 6:15-19.

150. 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

Declaration of Chris Schmandt  
Patent No. 10,839,789

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 recognised 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. Annotated Fig. 3 of *Smyth* below depicts allophones (annotated in several shades of blue) where groups of three speech states are provided for each allophone, and wherein the entire collection of speech states is collectively an acoustic model (annotated brown).

Fig. 3



*Smyth*, Fig. 3 (excerpt, annotated).<sup>4</sup>

151. 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

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

models were “available as off-the-shelf components and are well known in the art[.]”  
*Peters*, 9:45-48.

152. 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 “The 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*, 43 (“Given the sequence of triphones for a word, one can construct an equivalent *word-HMM* by simply concatenating the HMMs for the individual triphones[.]”), 32 (“Word HMMs are built by stringing together phonetic HMM models using NULL transitions[.]”).

153. Additionally, in my opinion, *Smyth* teaches an acoustic model memory. Specifically, state memory 342 stores a plurality of allophones. *Smyth*, 5:60-62. As I discussed above, *Smyth*'s collection of allophones stored in state memory 342 (where each allophone comprises three speech states that are stored in three state fields in said state memory 342) is an acoustic model, as claimed. In my opinion, because *Smyth* teaches state memory 342 stores model speech states (e.g., speech states 3241a, etc., and where three speech states correspond to an allophone) that are representative of acoustic speech utterances, *Smyth* teaches or at least renders obvious the claimed acoustic model memory.

**4. Claim 1(c): “a calculating apparatus for calculating distances indicating a similarity between the at least one feature vector and respective acoustic states of the acoustic model read from the acoustic memory; and”**

154. In my opinion, a POSITA would have understood *Smyth*'s disclosures of “classification processor 34” to refer to “classifier processor 341.” In my opinion, “classification processor 34” should have read “classification/classifier/classifying processor 341,” as element 34 refers to classifier 34 and element 341 refers to the processor of classifier 34. *Smyth*, 5:55-57, 5:12-13, 5:63, 6:1-8. Indeed, *Smyth* refers to classifier processor 341 and classifying processor 341 each several times, such that a POSITA would have understood both terms to be interchangeable with no difference in meaning. *Id.* Specifically, a POSITA would have understood “classifier 34 comprises a classifying processor 341” such that classifier processor 341 would

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

**a) “a calculating apparatus”**

155. In my opinion, *Smyth* teaches the claimed “calculating apparatus for calculating distances[.]” In particular, *Smyth* teaches classifier processor 341 that calculates probabilities.

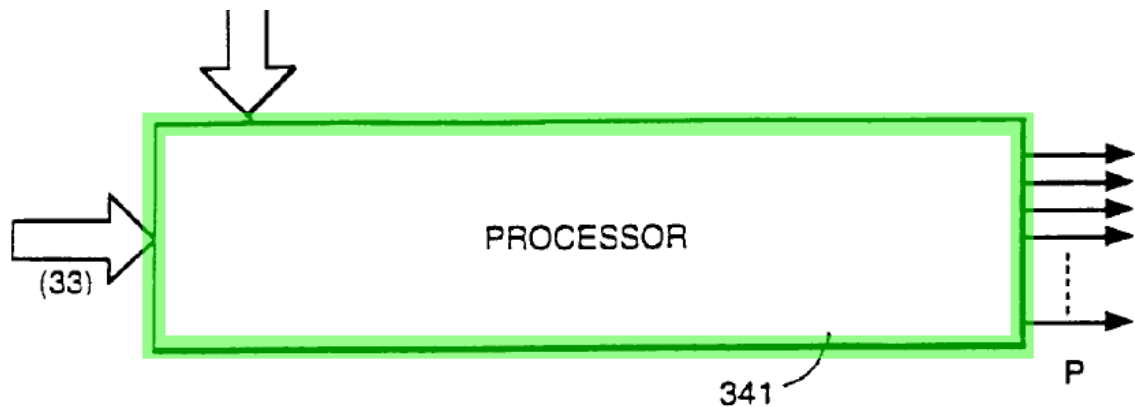
Referring to FIG. 3, in this embodiment, the classifier 34 comprises a classifying processor 341 and a state memory 342.

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.

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*, 5:55-6:4. I also note that *Smyth* refers to classifier processor 341 as a “processor” in annotated Fig. 3, below.

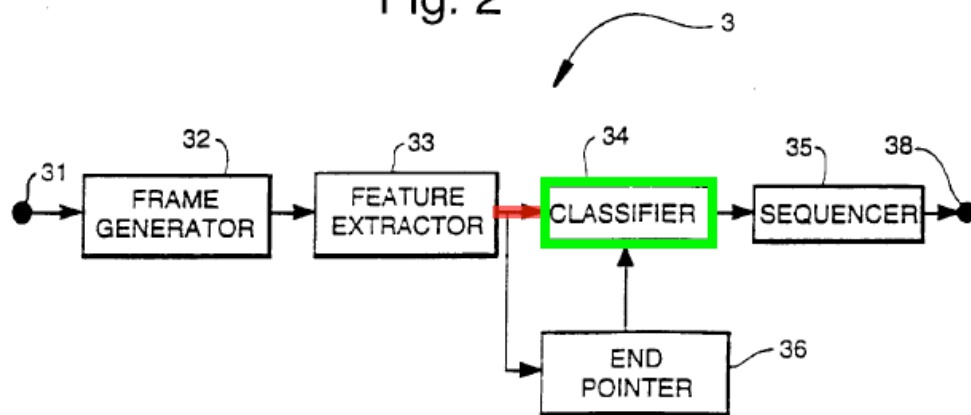


*Smyth*, Fig. 3 (annotated, excerpt). In my opinion, classifier processor 341 is a processor that calculates state probabilities and therefore a skilled artisan would have understood that classifier processor 341 is a calculating apparatus, as claimed.

156. I also note that *Smyth* teaches classifier processor 341 may be a digital signal processor (DSP). *Smyth*, 6:5-6. As I discussed above, it was well known in the art before the priority date of the '789 Patent that DSPs were processing 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 DSP in at least one embodiment, is a calculating apparatus, as claimed.

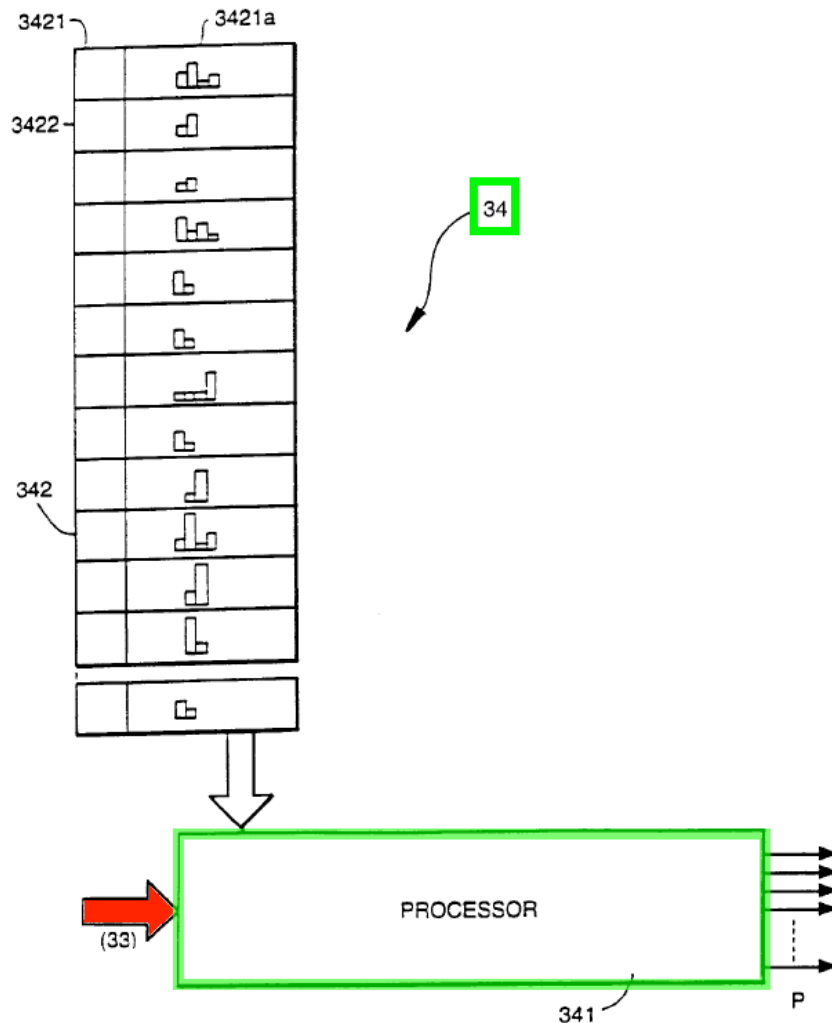
157. It is also my opinion that *Smyth* teaches a feature vector (annotated red) is output from feature extractor 33 and is received at classifier 34 (annotated green).

Fig. 2



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

Fig. 3



*Smyth*, Fig. 3 (annotated). Thus, in my opinion, *Smyth* teaches classifying processor 341 receives a feature vector.

**b) “[a calculating apparatus] for calculating distances indicating a similarity ...”**

158. It is also my opinion that *Smyth* teaches classifying processor 341 performs the claimed “calculating distances indicating a similarity between the at least one feature vector and respective acoustic states of the acoustic model read

from the acoustic model memory[.]” In particular, *Smyth*’s classifier processor 341 outputs a probability (i.e., distance) that the input feature vector corresponds to each state. In my opinion, such probabilities are a “distance” because said probabilities indicate the likelihood between a feature vector and each stored state in state memory 342.

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. Regarding the claimed calculating distances, recall the ’789 Patent describes a distance as “an indication of the probability or likelihood that a feature vector corresponds to a particular state.” ’789 Patent, 4:33-43. *Compare Smyth*, 5:63-6:4 with ’789 Patent, 4:33-43 (“[A] distance is an indication of the probability or likelihood that a feature vector corresponds to a particular state.”), 13:27-29 (“The distance calculator stage of the recognition process computes a probability or likelihood that a feature vector corresponds to a particular state.”), 13:29-33 (“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.”). *Smyth*’s disclosures parallel the ’789 Patent’s description.

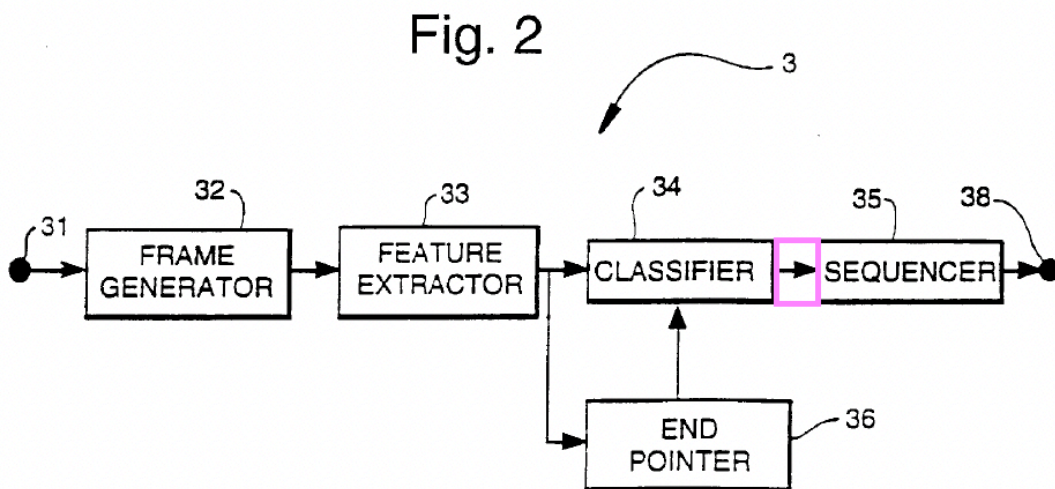
159. In my opinion, because the ’789 Patent describes distances as an indication of probability and because *Smyth* teaches outputting a probability for each state, a POSITA would have understood that *Smyth* teaches the claimed “calculat[e] distances[.]” Indeed, *Smyth* teaches that the plurality of state probabilities that are outputted indicates a likelihood that the inputted feature vector corresponds to each state of the acoustic mode. *Smyth*, 6:1-4. Additionally, each calculated probability (distance) is for an acoustic state of an acoustic model read from the acoustic model memory because *Smyth* teaches the processor 341 “is arranged to read each state field within the memory.” *Smyth*, 5:58–6:4, 1:34-37, 5:9, 5:59, Fig. 2. Each state field stores a speech state, discussed above, and a probability is determined for each state. *Id.* Therefore, *Smyth* teaches the claimed “calculating distances indicating a similarity between the at least one feature vector and respective acoustic states of the acoustic model read from the acoustic memory[.]” *Smyth*, 5:7-9, 5:59, 5:63-6:4, 1:34-37, Fig. 2.

**5. Claim 1(d): “a second interface for sending at least one distance calculated by the calculating apparatus;”**

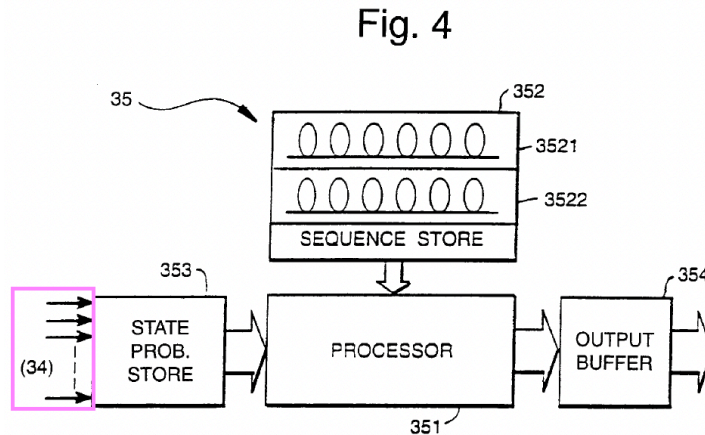
**a) *Smyth*’s Teachings**

160. In my opinion, *Smyth* teaches the claimed “second interface[.]” In particular, *Smyth* teaches classifier 34 and at least one signal line/communication

path between classifier 34 and sequencer 35 or the classifier 34's output. *See* Claim 1(a) above where I opined on the claimed “first interface[.]” *Smyth* teaches classifier 34 outputs “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*, 6:1-4, 5:7-9, 5:63-67. *Smyth* teaches “sequencer 35 which is arranged to receive the classification results form [*sic*] the classifier 34 and to determine the predetermined utterance to which the sequence of classifier output indicates the greatest similarity[.]” *Smyth*, 5:12-15, 6:13-15 (“Also provided is a state probability memory 353 which stores, for each frame processed, the state probabilities output by the classifier 341.”), 6:36-42 (“The parsing processor 351 is arranged to read, at each frame, the state probabilities output by the classifier processor 341[.]”). I have reproduced annotated Fig. 2 from the Petition below. Annotated Fig. 2 depicts the output of classifier 34 (i.e., probabilities) transmitted to sequencer 35.



*Smyth*, Fig. 2 (annotated); *see* Claim 1(c) above. I have also reproduced annotated Fig. 4 from the Petition below. Annotated Fig. 4 depicts state probability memory 353 of sequencer 35 receiving the probabilities output from classifier 34.



*Smyth*, Fig. 4 (annotated). In my opinion, a POSITA would thus have understood that state probability memory 353 of sequencer 35 receives state probabilities (i.e., distances) that are output from classifier 34 and in particular classifier processor 341. Thus, in my opinion, *Smyth* teaches the claimed “sending at least one distance calculated by the calculating apparatus” because *Smyth*’s collective classifier and signal line/communication path between classifier 34 (which includes classifier processor 341) and sequencer 35 (which includes state probability memory 353) transmits state probabilities.

161. It is also my opinion that *Smyth* teaches or at least renders obvious the claimed second interface. Namely, *Smyth* teaches classifier 34 and at least one of the associated signal line or communication path between classifier 34 and sequencer

35 is a “second interface.” A POSITA would have understood that *Smyth*’s collective classifier and signal line/communication path transmit probabilities to sequencer 35 and sequencer 35 includes components such as parsing processor 351 and state probability memory 353 that receive said calculated probabilities. Thus, *Smyth*’s collective classifier and signal line/communication path are or include a second interface, as claimed, or such would have been obvious to a POSITA.

162. I note the ’789 Patent does not explicitly define “interface[.]” but through usage a POSITA would have understood the ’789 Patent to describe standard computer interfaces used for communicating between computer components. The following are some illustrative examples of the ’789 Patent’s references to interfaces.

FIG. 17 shows a block diagram of a distance calculation accelerator 200, used in an embodiment of the invention. **A feature vector is received at input 212 of the accelerator 200.** The feature vector is loaded into feature vector (FV) memory A 208 via a bus 211.

*’789 Patent*, 25:6-10 (emphasis added).

The distance engine may support the pass-through of data from the front-end to the search stage. ... The data passed through may be for any purpose. Examples might include ... data from another input device such as a button or keyboard in a multi-modal **interface**.

*’789 Patent*, 5:34-48 (emphasis added).

The distance engine may implement one or more additional pipeline stages to overcome delays introduced by low bandwidth, high latency, or conflicted bus **interfaces**.

'789 Patent, 5:58-60 (emphasis added).

FIG. 11 shows a Language Model **interface** architecture.

'789 Patent, 9:7 (emphasis added). *See also* '789 Patent, 22:24-26, 10:12-21, 10:37-40, 24:32-36, Fig. 11.

163. It is also my opinion that classifier processor 341 may be a DSP. *Smyth*, 5:12-15, 5:63-67, 6:13-15, Figs. 2-3. *See* my opinions at Claim 1(c) above on the same. Further, *see* my opinions regarding the claimed “first interface[.]” *See* Claim 1(a) above; *Smyth*, 6:5-8, 5:49-53; *Kloker*, 3-4, 14-15.

#### **b) Mozer's Teachings**

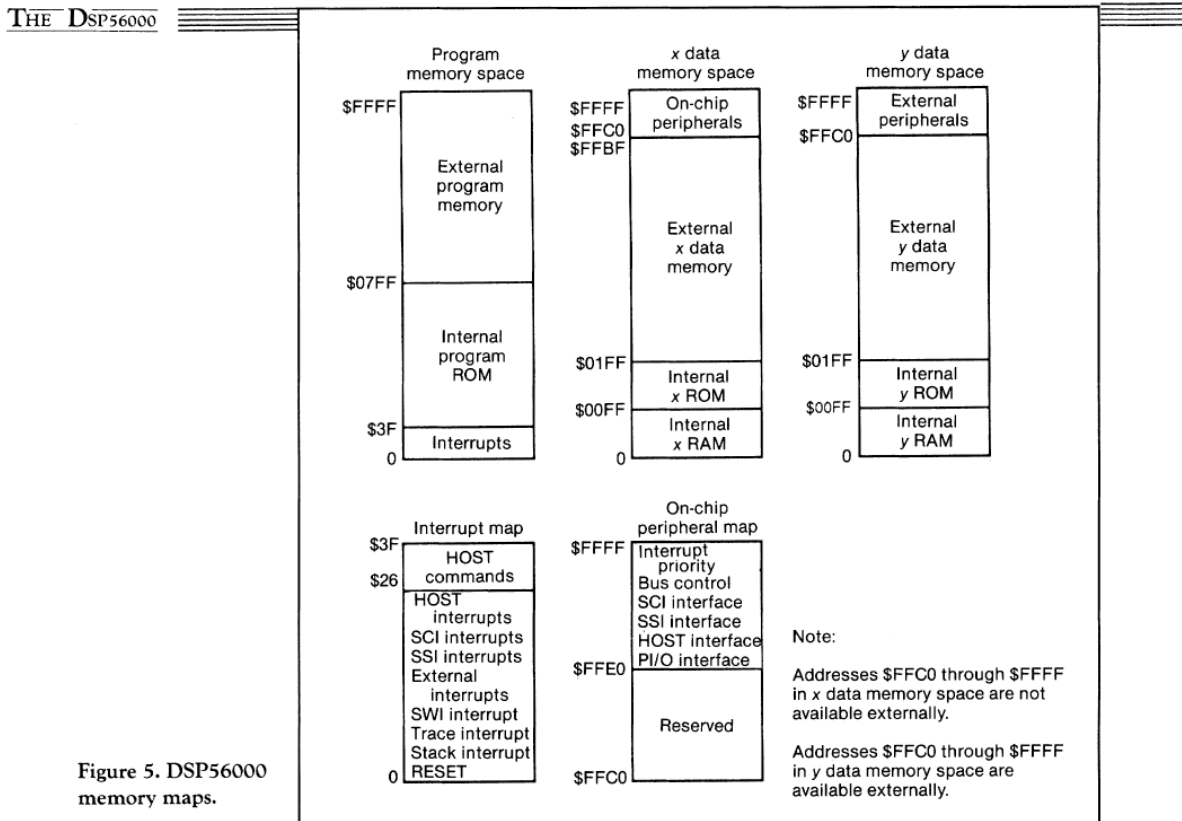
164. I also note that *Mozer* teaches a “second interface,” similar to the mapping above for the “first interface.” *See* Claim 1(a) above; *Mozer*, 5:47-53, 5:60-67, 6:1-3, 7:7-13, 7:38-39. Figs. 2-3. In my opinion, when *Mozer's* system as depicted in Fig. 2 includes an audio recognition peripheral of Fig. 3, bus 230 is used to send the distance calculation from the vector processor to the external CPU. Therefore, in my opinion, *Mozer's* bus 230 is at the least a second interface, as claimed, for sending a distance calculated by the vector processor 223, 323. *Mozer*, Fig. 4. I provided my opinions regarding a POSITA would have been motivated and

found it obvious to combine *Mozer* and *Smyth* above. See Claim 1(a) above. I incorporate my opinions from Claim 1(a) herein.

**6. Claim 1(e): “wherein the calculating apparatus and the acoustic model memory are fabricated on a single integrated circuit.”**

**a) Smyth’s Teachings**

165. As I discussed above for Claim 1(b), *Smyth* teaches the claimed acoustic model memory. See Claim 1(b) above; *Smyth*, 5:1-15, 5:55-57. Also, as I discussed above for Claim 1(c), *Smyth* teaches a classifier processor 341 that, in my opinion, teaches or at least renders obvious the claimed calculating apparatus. See Claim 1(c) above; *Smyth*, 5:49-53, 6:5-8. *Smyth* also teaches classifier processor 341 may be a DSP (e.g., Motorola DSP56000). See Claim 1(c) above. In my opinion, the Motorola DSP56000 provides both memory and a processor fabricated on a single integrated circuit (IC). *Kloker* (Ex. 1009), 1 (discussing the Motorola DSP56000 includes “a core processor, RAM, ROM, peripheral interfaces, and a memory expansion interface on a single, 88-pin chip[.]”

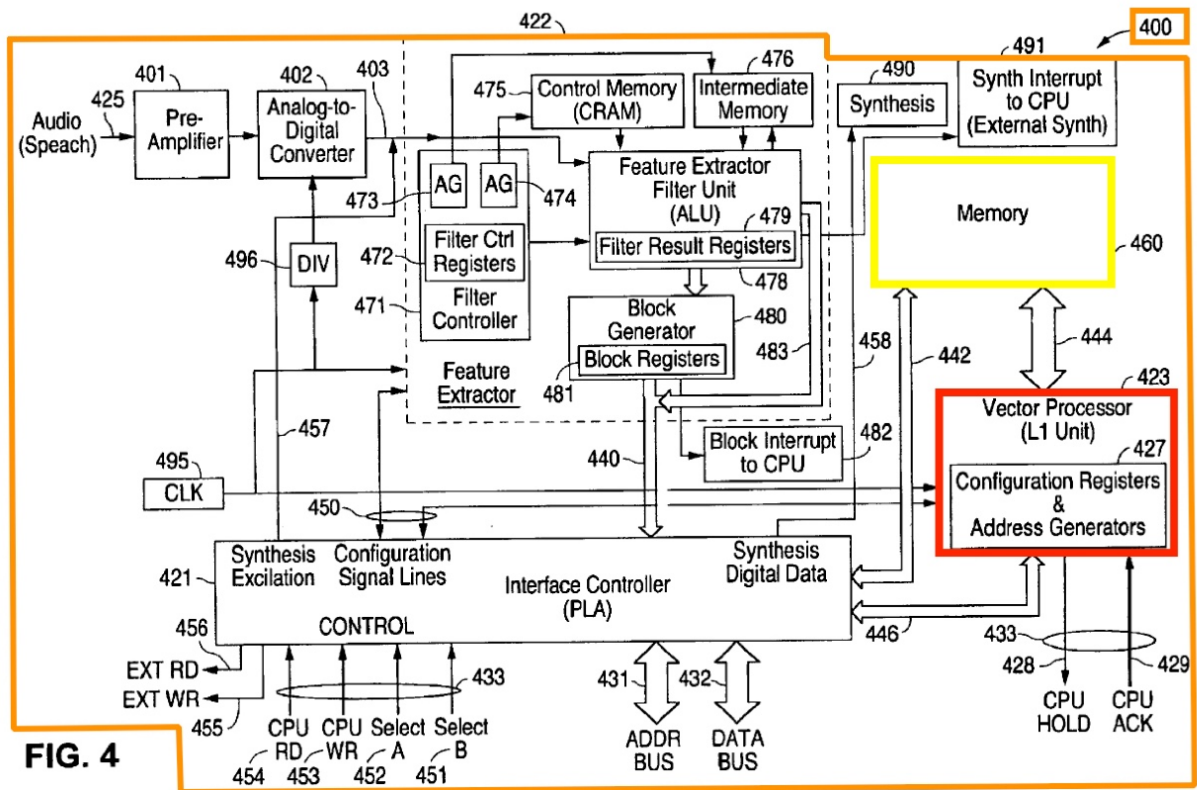


Kloker, 6 (“Figure 5. DSP56000 memory maps”). A POSITA would have understood or at least found obvious that *Smyth* thus teaches a single IC on which both classifier processor 341 and state memory 342 are fabricated. Therefore, in my opinion, for at least these reasons, *Smyth* teaches or at least renders obvious Claim 1(e).

## b) *Mozer’s Teachings*

166. It is also my opinion that *Mozer* teaches a single integrated circuit on which both a vector processor 423 (i.e., a calculating apparatus) and memory 460 (i.e., acoustic model memory) are fabricated. Specifically, *Mozer* teaches an “audio recognition peripheral integrated circuit 400” that includes “vector processor 423

and/or memory 460.” *Mozer*, 9:22-26, 7:38-39, 7:53-58, 10:7-12, Fig. 4. *Mozer* also teaches loading first and second vectors into memory 460 to calculate a distance between said vectors. *Mozer*, 10:1-26. I have reproduced annotated Fig. 4 of *Mozer* from the Petition below. I note the memory 460 (annotated yellow) and vector processor 423 (annotated red). The audio recognition peripheral integrated circuit 400 (annotated orange).



*Mozer*, Fig. 4 (annotated). Therefore, in my opinion, *Mozer* teaches a single integrated circuit on which both a calculating apparatus and a memory for storing acoustic models are fabricated.

**c) Motivation to Combine *Smyth* in View of *Mozer***

167. In my opinion, a skilled artisan would have been motivated and found it obvious to modify *Smyth* to fabricate both state memory 342 and classifier processor 341 of classifier 34 on a single integrated circuit per *Mozer*'s teachings. I note *Mozer* teaches advantages of providing a vector processor and acoustic memory on a single integrated circuit. *Mozer*, 9:16-18 ("However, the embodiments described herein are illustrative of the advantages of providing a recognition peripheral system on one integrated circuit[.]"). Further, *Mozer* teaches "offload[ing]" calculations to a particular piece of hardware (e.g., vector processor) so a different processor can perform other speech recognition tasks such as word identification using calculated distances from the vector processor. *Mozer*, 10:46-52, 10:1-34. In my opinion, when modifying *Smyth* according to *Mozer*'s teachings a POSITA would have expected to likewise capture *Mozer*'s advantages. Further, a skilled artisan would have understood that such a use of a known technique of fabricating certain components on a single integrated circuit would have improved *Smyth*'s similar speech recognition system (specifically, acoustic coprocessor, *see* Claims 1(Pre)-1(d) above) in the same way as *Mozer*'s speech recognition system. In particular, such a use of a known technique would have offloaded "burdensome" processing to an acoustic processor in the same way as such processing is offloaded to *Mozer*'s audio recognition peripheral integrated circuit 400. I also note that *Smyth*

already teaches performing distance calculations utilizing a DSP (e.g., Motorola DSP56000) where the DSP includes a processor 341 and memory 342. *See* Claim 1(c) above; *Smyth*, 6:1-6. To the extent it is argued that *Smyth* does not teach a single IC for the DSP that performs distance calculations, it would have been obvious to a POSITA and a POSITA would have been motivated to make such a modification.

168. It is also my opinion that such a modification would have had a reasonable expectation of success. As I noted above, *Smyth* and *Mozer* both teach speech recognition systems with similar hardware. For example, *Smyth* teaches classifier 34 is part of recognition processor 3, and classifier 34 comprises state memory 342 and classifier processor 341. *Smyth*, 4:22-31, 4:40-42, 5:1-12, Figs. 1-3. I also note that *Smyth* discloses various hardware arrangements and modifications such as utilizing three separate DSPs or two DSPs and a processor for specific processing tasks. *Smyth*, 5:49-53, 6:5-8, 6:47-51, 10:51-58. Therefore, in my opinion, modifying *Smyth* according to *Mozer's* teachings would have been straightforward and would not have required undue experimentation. Further, as I noted above, it was known in the prior art to fabricate a single integrated circuit that comprised a calculating apparatus and acoustic model memory for use in a speech recognition system. *See* Section V.F discussing *Toyoda* (Ex. 1029). I also note it was known in the prior art to fabricate a single integrated circuit with two or more processors or DSPs and wherein each DSP or processor has its own memory. *See*

Section V.G discussing *Boike* (Ex. 1006) and *Ozcelik* (Ex. 1034). Therefore, for at least these reasons, a POSITA would have been motivated and found it obvious to modify *Smyth* to the extent not expressly taught in *Smyth* to fabricate a single integrated circuit with a classifier processor 341 and state memory 342 thereon. For the reasons I discussed above, such a modification would have been advantageous and would have had a reasonable expectation of success.

**B. Independent Claim 10**

**1. Claim 10[Pre]: “An acoustic coprocessor, comprising:”**

169. See Claim 1(Pre) above.

**2. Claim 10(a): “a first interface for receiving at least one feature vector;”**

170. See Claim 1(a) above.

**3. Claim 10(b): “a calculating apparatus for calculating distances indicating a similarity between the at least one feature vector and at least one acoustic state of an acoustic model read from an acoustic model memory;”**

171. See Claims 1(b)-1(c) above.

**4. Claim 10(c): “a second interface for sending at least one distance calculated by the calculating apparatus.”**

172. See Claim 1(d) above.

**C. Dependent Claim 11**

**1. Claim 11: “The acoustic coprocessor of claim 10, wherein the calculating apparatus and the acoustic model memory are fabricated on a single integrated circuit.”**

173. See Claims 1(e), 10 above.

**D. Dependent Claim 14**

1. ***Claim 14:***” *The acoustic coprocessor of claim 10, wherein the second interface for sending distances is a serial bus.”*

174. *Mozer* teaches using a bus 130 to transfer information between a processing unit 110 and an audio recognition peripheral 120 and that the bus 130 may be a serial bus 130.

As part of an audio recognition operation, the programmable processing unit 110 also transmits control signals and interface information to the audio recognition peripheral 120 over the general purpose bus 130. The general purpose bus 130 may comprise control, address, and data information. The bus 130 may include either distinct control, address, and data lines or parallel lines carrying the combination of control/address/data. Additionally, the bus 130 may be comprised of partitioned combinations thereof such as distinct control lines and combined address/data lines. Of course, a serial bus may also be employed.

*Mozer*, 4:17-28.

175. In my opinion, a POSITA would have been motivated and found it obvious to modify *Smyth* to use a serial bus for the second interface per *Mozer*’s teachings. A POSITA would have understood and found obvious that using a serial bus would have advantageously reduced the trace complexity of an integrated circuit (printed circuit board) because a serial bus would have used one line instead of eight or sixteen parallel lines (copper traces). Thus, a serial bus would have

advantageously required less space on a circuit board. Further, such a modification would have had a reasonable expectation of success because using serial buses to transmit information/data was well known in the prior art. One example is *Poulis*, which teaches it is more efficient in some circumstances to use a serial bus due to the limited number of interconnection lines. *Poulis* (Ex. 1064), 6:64-7:11. Further, I note that such a modification to *Smyth* would have had a reasonable expectation of success because serial buses were well known and commonly used.

**E. Dependent Claim 17**

1. ***Claim 17: “The acoustic coprocessor of claim 10, wherein the acoustic coprocessor is capable of, in response to receiving a feature vector, autonomously calculating the distances for each acoustic state of the acoustic model.”***

176. See Claim 10 above. I note that *Smyth* teaches classifier processor performs a distance calculation. See Claims 1(b)-1(c) above; *Smyth*, 5:63-67, 6:1-4. The classification processor calculates the probabilities *in response* to receiving a feature vector because the “classifier 34 receiv[es] the succession of feature vectors and operat[es] on each with a plurality of model states, to generate recognition results.” *Smyth*, 5:7-9. A POSITA would have understood or at least found obvious that *Smyth*’s classifier processor 341 would have autonomously calculated the distances for each acoustic state of the acoustic model because classifier processor 341 is “arranged” to both read and calculate distances.

**X. OPINIONS REGARDING GROUND 2: CLAIMS 2, 4, 6-9, 12, 16, 18, AND 20-21 ARE OBVIOUS OVER *SMYTH*, *MOZER*, AND *NGUYEN***

**A. Dependent Claim 2**

**1. Claim 2: “The acoustic coprocessor of claim 1, wherein the acoustic model contains Gaussian parameters and the distances are probabilities”**

177. See Claim 1 above. *Smyth* teaches an acoustic model and calculating distances that are probabilities. See Claims 1(b)-1(c). *Nguyen* teaches a speech recognition system that computes a similarity measure between frames of input speech and reference patterns and defines a “local measure of closeness” between acoustic feature vectors. *Nguyen* (Ex. 1047), 2:30-38. *Nguyen* teaches using “Gaussian distribution data or Gaussian mixture density data which serves as the basis for the acoustic model” and “is loaded into memory.” *Nguyen*, 3:35-46, 3:50-54 (“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.”). Further, *Nguyen* teaches that each “Hidden Markov Model includes a collection of probabilities associated with the states themselves and transition amongst the states. Because probability values associated with each state may be more complex than a single value could represent, some systems will represent probability in terms of a Gaussian distribution.” *Nguyen*, 2:49-54, 2:54-67 (*Nguyen* teaches that to “provide a more robust model, a mixture of Gaussian distributions may be used in a blended

manner to represent probability values as shown diagrammatically at 26 and referenced by a mixture index pointer 28. Thus, associated with each state is a mixture index pointer which in turn identifies the Gaussian mixture density data for that state.”), Fig. 1.

178. In my opinion, a POSITA would have been motivated and found it obvious to modify *Smyth* to use *Nguyen’s* Gaussian distribution to represent probability of the acoustic model. I note *Nguyen* expressly motivates such a modification, including because *Nguyen* teaches “probability values associated with each state may be more complex than a single value could represent,” and to “provide a more robust model[.]” *Nguyen*, 2:49-67. Further, a POSITA would have understood and found obvious that Gaussians reflect probability which in turn can reflect variability in human speech. For *Smyth*, each model is fixed, one set of feature values. However, people speak slightly differently, and the Gaussian probability representation better reflects variability. The center of the Gaussian (its “mean”) would correspond to *Smyth’s* stored feature, which is representative of a state, but the width of a Gaussian (its standard deviation) indicates how strictly the heard speech needs to match this mean or typical vector. Further, because a Gaussian can be completely specified by its mean and standard deviation, it is a concise way to represent speech. Since it is already a measure of probability, it is computationally efficient, and subject to well-known algorithms to compute the probability of a given

Gaussian representing a given HMM state. Therefore, a POSITA would have been motivated to make such a modification to *Smyth*. Such a modification would have had a reasonable expectation of success because representing the probability of an acoustic model in terms of Gaussians was known in the art, as taught by *Nguyen*, and was well understood. Such a modification would have been straightforward and would have required only minor software or programming changes to implement.

**B. Dependent Claim 4**

1. ***Claim 4: “The acoustic coprocessor of claim 2, wherein the second interface for sending distances is a serial bus.”***

179. *Smyth* in view of *Mozer* teaches a second interface for sending distances is a serial bus. *See* Claim 14 above. *Mozer*, 4:17-28.

**C. Dependent Claim 6**

1. ***Claim 6: “The acoustic coprocessor of claim 4, wherein the acoustic coprocessor, in response to receiving a feature vector, autonomously calculates the distance for an associated acoustic state of the acoustic model.”***

180. *See* Claims 1(b), 4 above. I note that *Smyth* teaches classifier processor performs a distance calculation. *See* Claims 1(b)-1(c) above; *Smyth*, 5:63-67, 6:1-4. A POSITA would have understood or at least found obvious that *Smyth*'s classifier processor 341 would have autonomously calculated the distances for each acoustic state of the acoustic model because classifier processor 341 is “arranged” to both read and calculate distances.

**D. Dependent Claim 7**

1. ***Claim 7: “The acoustic coprocessor of claim 6, further comprising a result memory, wherein the distance for an associated acoustic state of the acoustic model is stored in the result memory.”***

181. See Claim 6 above. *Nguyen* teaches a result memory to store the output of distance calculations between a feature vector and one or more acoustic models.

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.

*Nguyen*, 3:51-57, 3:64-4:22, 3:11-24, 3:35-50, 2:30-38.

182. In my opinion, a POSITA would have been motivated and found it obvious to modify *Smyth-Mozer* to include one or more result memories as taught by *Nguyen*. Indeed, such a modification would have provided memory storage space for more distances to be calculated and written to memory without having to wait for the state probability memory 353 of sequencer 35 to be available to write distance calculations results thereto. Such a modification would have provided *Smyth*'s classifier 34 with a memory for storing said distance calculations. See Claim 1(c). Such a use of a known technique of using a result memory to store distance calculation results would have advantageously improved *Smyth*'s speech recognition

system in the same way as *Nguyen*'s. I also note that such a modification would have had a reasonable expectation of success because using a memory to store data and in particular distance calculation results was well known as taught by *Nguyen*. A POSITA would have understood that simply adding a memory would have been simple to implement and would not have required undue experimentation to implement. Further, even though a POSITA would have understood that adding an additional result memory to *Smyth* may have had an associated cost increase, a POSITA still would have been motivated and found it obvious to make such a modification because of improved performance gained by adding an additional memory (i.e., result memory).

**E. Dependent Claim 8**

1. ***Claim 8: "The acoustic coprocessor of claim 7, wherein the result memory is configured so that distances calculated from a first feature vector are sent over the second interface for sending while at least one distance calculated from a second feature vector is being stored in the one or more result memories."***

183. *Smyth-Mozer* in view of *Nguyen* teaches a result memory. See Claim 7 above. *Nguyen* teaches computing a similarity measure for each acoustic feature vector in a first group of vectors during step 36 and then performing a word search process in step 44. *Nguyen*, 3:51-4:16. *Nguyen* teaches that while the search process is performed, another similarity measure is computed for a second ("subsequent") group. Per *Nguyen*:

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.

184. In my opinion, a POSITA would have understood and found obvious that *Nguyen* thus teaches or at least renders obvious “wherein the result memory is configured so that distances calculated from a first feature vector are sent over the second interface for sending while at least one distance calculated from a second feature vector is being stored in the one or more result memories,” as claimed. Indeed, *Smyth-Mozer* as modified by *Nguyen* would have transmitted distances calculated from a first feature vector to perform a word identification process while

distances are calculated from a second feature vector and stored in one or more result memories. As I discussed above, a POSITA would have been motivated and found it obvious to modify *Smyth-Mozer* to include a result memory. *See* Claim 7 above. Additionally, in my opinion, a POSITA would have been motivated and found it obvious to transmit results from a result memory to *Smyth's* memory 353 via a second interface to perform word identification processing. *See* Claim 1(d) above. Therefore, in my opinion, *Smyth-Mozer-Nguyen* teaches a result memory that would have been configured to transmit via a “second interface” distances calculated from a first feature vector while storing distances calculated from a second feature vector in one or more result memories.

**F. Dependent Claim 9**

**1. Claim 9(Pre): “A speech recognition system comprising the acoustic coprocessor of claim 7, the speech recognition system further comprising at least one CPU, wherein:”**

185. *Smyth* teaches using a processor or DSP for parsing processor 351. For example, *Smyth* teaches “parsing processor 351 may, for example, be a microprocessor such as the Intel™ i-486™ microprocessor or the Motorola™ 68000 microprocessor, or may alternatively be a DSP device.” *Smyth*, 6:47-50. I note that the '789 Patent describes processor and CPU as interchangeable. “We are also using the term ‘CPU’ and ‘processor’ interchangeably.” '789 Patent, 30:1-2. Therefore, in my opinion, a POSITA would have understood or at least found obvious that

implementing parsing processor 351 as a processor would have also taught implementing parsing processor 351 as a CPU according to the description of the '789 Patent. Therefore, in my opinion, *Smyth-Mozer-Nguyen* teaches Claim 9(Pre). *Smyth*, 6:10-12, 6:36-51.

**2. Claim 9(a): “the CPU executes software to perform or initiate a speech recognition word search using the distances calculated by the acoustic coprocessor in response to receiving the first feature vector, and”**

186. 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.

187. *Smyth* teaches a sequencer 35 that comprises parsing processor 351 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. In particular, *Smyth* teaches parsing processor 351 (of sequencer 35) reads probabilities (i.e., distances) output by classifying 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, 6:43-61. *Smyth* further teaches that each state sequence to be recognized corresponds to a “word, phrase or other speech sequence to be recognised” 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 performing or initiating a speech recognition word search using the distances calculated by the acoustic coprocessor in response to receiving the first feature vector, as claimed.

3. ***Claim 9(b): “a time period for the speech recognition word search overlaps a time period for the acoustic coprocessor to calculate the distance in response to receiving the second feature vector.”***

188. See Claims 1(c), 9(a) above. *Smyth-Mozer* in view of *Nguyen* teaches a result memory. See Claim 7 above.

189. 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 phoneme 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.

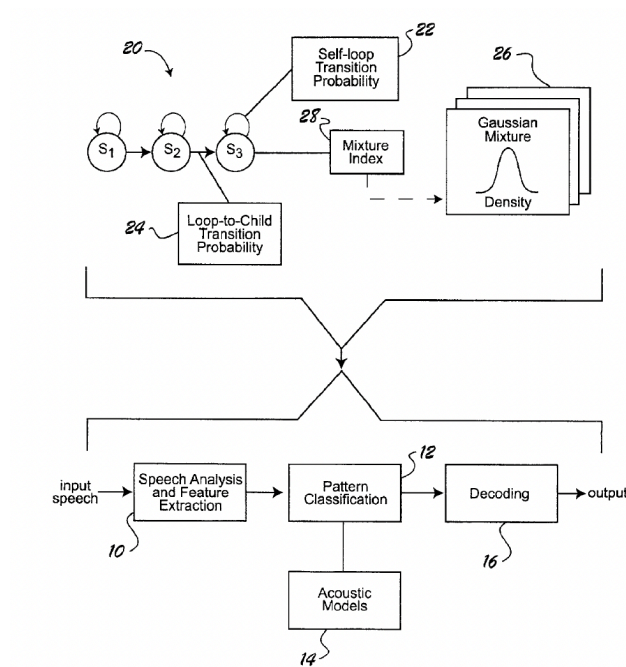


FIG. 1

*Nguyen*, Fig. 1.

190. 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.

191. 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).

192. *Nguyen* teaches computing a similarity measure for each acoustic feature vector in a first group of vectors during step 36 and then performing a word search process in step 44. *Nguyen*, 2:30-38, 3:51-4:16. *Nguyen* teaches that while the search process is performed, another similarity measure is computed for a second (“subsequent”) group, and that this process is performed iteratively “for each of the acoustic feature vectors that represents the input speech.” Per *Nguyen*:

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), *Smyth-Mozer-Nguyen* teaches “a time period for the speech recognition word search overlaps a time period for the acoustic coprocessor to calculate the distance in response to receiving the second feature vector,” as claimed.

193. In my opinion, a POSITA would have been motivated and found it obvious to modify *Smyth-Mozer* according to *Nguyen*’s teachings. Specifically, modifying *Smyth-Mozer* 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.

194. Additionally, modifying *Smyth-Mozer* to use its multiple DSPs/processors to perform *Nguyen's* contemporaneous distance calculation and searching steps would have advantageously increased *Smyth-Mozer's* system performance by performing both steps in parallel. *Smyth-Mozer* teaches an embodiment where three DSPs are utilized (i.e., one DSP for frame generator 32 and feature extractor 33, a second DSP for classifying processor 341, and a third DSP for parsing processor 351) and another embodiment where two DSPs and a processor are utilized (one DSP for frame generator 32 and feature extractor 33, a second DSP for classifying processor 341, and a separate processor for parsing processor 351). *Smyth*, 5:55-57, 6:5-8, 6:43-51. In my opinion, a POSITA would have been motivated to modify *Smyth-Mozer's* multiple DSPs/processors to use *Nguyen's* contemporaneous processing in either embodiment to cause either *Smyth-Mozer's* second and third DSPs/processors to perform tasks in parallel or for all three of *Smyth-Mozer's* DSPs/processors 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.

195. A POSITA would have understood that modifying *Smyth-Mozer* 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 *Smyth-Mozer* 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, *Smyth-Mozer* would have been modified to generate feature vectors in groups, batch the results, and transmit the group of acoustic feature vectors to classifying 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 *Smyth-Mozer* teaches utilizing multiple DSPs/processors 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.

196. Additionally, a POSITA would have had a reasonable expectation of success when modifying *Smyth-Mozer* to use *Nguyen's* “contemporaneous”

processing technique. *Smyth-Mozer* itself already teaches embodiments with three separate DSPs/processors, such that modifying *Smyth-Mozer* to have the distinct DSPs/processor to 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.

**G. Dependent Claim 12**

1. ***Claim 12: “The acoustic coprocessor of claim 10, wherein the acoustic model contains Gaussian parameters and the distances are probabilities.”***

197. See Claims 2, 10 above.

**H. Dependent Claim 16**

1. ***Claim 16 [Pre]: “A speech recognition system comprising the acoustic coprocessor of claim 10, the speech recognition system further comprising at least one CPU, wherein:”***

198. See Claims 9(Pre), 10 above.

2. ***Claim 16(a): “the CPU executes software to perform or initiate a speech recognition word search using one or more distances calculated by the acoustic coprocessor in response to receiving a first feature vector, and”***

199. See Claim 9(a) above.

3. ***Claim 16(b): “a time period for the speech recognition word search overlaps with a time period for the acoustic coprocessor to calculate one or more distances in response to receiving a second feature vector.”***

200. See Claim 9(b) above.

**I. Dependent Claim 18**

1. *Claim 18: “The acoustic coprocessor of claim 17, further comprising one or more result memories configured to store the distances for each acoustic state of the acoustic model.”*

201. See Claims 7, 10, 17 above. As I discussed above, *Smyth-Mozer* as modified by *Nguyen* teaches a result memory. See Claim 7 above. Also, as I discussed above, *Smyth*’s output probabilities meet the claimed distances. See Claims 1(c)-1(d).

**J. Dependent Claim 20**

1. *Claim 20: “The acoustic coprocessor of claim 18, wherein the calculating apparatus is capable of computing distances using acoustic models representing Gaussian probability distributions.”*

202. See Claims 2, 18 above.

**K. Dependent Claim 21**

1. *Claim 21: “The acoustic coprocessor of claim 20, wherein the acoustic model uses a neural network to calculate at least one of the distances.”*

203. See Claim 20 above. *Smyth-Mozer-Nguyen* teaches claim 21. Namely, *Mozer* teaches using a neural network for speech recognition and that using a neural network for speech recognition was “well known by those skilled in the art.” *Mozer*, 10:35-52, 5:53-60; see also my opinions above regarding *Jelinek* (Ex. 1020) teaching neural networks used in speech recognition. See Section V.D (discussing *Jelinek*).

It is important to note that the present invention is not limited to the above embodiment utilizing the vector processor for carrying out a

distance calculation between two vectors. For example, if the audio recognition algorithm running on the external processor was a neural network recognition algorithm, the vector operation executed by a vector processor may include, for example, loading pattern values and neural network weights into the memory 460. The vector operation in that case would be to perform a multiply-accumulate operation on the pattern values and neural network weights as is well known by those skilled in the art.

*Mozer*, 10:35-52.

204. In my opinion, a POSITA would have been motivated and found it obvious to modify *Smyth* to use a neural network to calculate distances as claimed, as taught by *Mozer*. Modifying *Smyth* to utilize a neural network for distance calculations would have advantageously been use of a known technique to improve *Smyth*'s similar system for speech recognition in the same way as *Mozer*'s speech recognition system. I note that *Jelinek* teaches that utilizing a neural network to "estimate a posteriori probabilities" and would have achieved state of the art results in a speech recognition system. *See* Section V.D (discussing *Jelinek*). In my opinion, such a modification would have had a reasonable expectation of success because numerous techniques/algorithms for performing distance calculations were known in the prior art, including using a neural network in the manner taught by *Mozer* to perform such distance calculations. *Mozer*, 10:35-52. Additionally, as I noted above, such was well known in the prior art as taught by *Jelinek*, and therefore a POSITA

would have had a reasonable expectation of success when making such a modification.

**XI. OPINIONS REGARDING GROUND 3: CLAIM 13 IS OBVIOUS OVER SMYTH, MOZER, AND HOUSER**

**A. Dependent Claim 13**

**1. Claim 13:” The acoustic coprocessor of claim 10, wherein the acoustic model memory is a FLASH memory.”**

205. As I opined above, *Smyth* teaches an acoustic model memory as claimed. *See* Claim 1(b) above. *Smyth* does not expressly refer to a memory type used to implement state memory 342. *Houser* (Ex. 1040) teaches using a flash memory to store phonemic or template vocabulary data for use in a speech recognition system. *Houser*, 15:33, 15:46-48. A POSITA would have been motivated and found it obvious to implement *Smyth*’s memory 342 using FLASH memory because FLASH memory was advantageously fast and non-volatile. A POSITA would have had a reasonable expectation of success performing a simple substitution by using FLASH memory as taught by *Houser* instead of *Smyth*’s unspecified memory type. Such a use of a known technique of implementing an acoustic model memory as FLASH memory would have advantageously improved *Smyth*’s acoustic model memory in the same way as *Houser*’s acoustic model memory. A POSITA would have had a reasonable expectation of success performing such a modification because FLASH memory was well known and because *Smyth* does not specify a particular memory type, such that a POSITA would have

understood FLASH memory would have been an acceptable and logical choice, especially since such is expressly taught by *Houser*.

**XII. OPINIONS REGARDING GROUND 4: CLAIMS 22-24 ARE OBVIOUS OVER *SMYTH*, *MOZER*, *NGUYEN*, AND *VENSKO***

**A. Dependent Claim 22**

1. *Claim 22: “The acoustic coprocessor of claim 21, wherein the acoustic coprocessor is capable of generating an interrupt in response to a completion of calculation of a set of distances.”*

206. See Claim 21 above. In my opinion, *Vensko* teaches performing a distance calculation between an assigned template and a character distance vector. Specifically, *Vensko* (Ex. 1038) teaches template processors that perform a dynamic programming algorithm (DPA). *Vensko*, 6:22-33 (below), 10:10-19.

The master processor, via the circuit means even numbers 180-186 and the remote bus 170, after the beginning of an utterance has been detected, causes the character distance vector to be transferred to each of the template processor for comparison. The master processor via an interrupt commands the template processors to begin processing the character distance vectors on a frame basis. When each template processor has run the DPA on each of the assigned templates for the latest character distance vector and obtained the results, they generate an interrupt to the master processor 160 informing it that the results are ready for transfer.

*Vensko*, 6:22-33.

207. In my opinion, a POSITA would have been motivated and found it obvious to modify the *Smyth* combination's classifier 34 to be capable of generating an interrupt in response to a completion of calculation of a set of distances, as claimed, per *Vensko*'s teachings. Modifying the *Smyth* combination in this way would have advantageously allowed *Smyth*'s classifier 34 (see my discussion of Claim 1(Pre) for the mapping of "acoustic coprocessor") to generate an interrupt to inform the sequencer 35 (comprising parsing processor 351, state sequence memory 352, and state probability memory 353) that a set of distances have been calculated. In my opinion, generating an interrupt in this manner would have been a more efficient way to transmit data to sequencer 35. Such a modification to the *Smyth* combination would have had a reasonable expectation of success, requiring only small software programming changes, and would have been straightforward. Further, generating interrupts in this manner was known in the art, with *Vensko* as one example. I also discussed interrupts being known in the art above. See Section V.H.

**B. Dependent Claim 23**

1. ***Claim 23: "The acoustic coprocessor of claim 22, wherein the acoustic model memory includes a plurality of storage locations within a static random access memory (SRAM)."***

208. See Claim 22 above. In my opinion, *Mozzer* teaches memory 460, which stores a "second vector representing a template" may be an SRAM, such as a 2K

SRAM or 4K SRAM. *Mozer*, 10:17-21, 9:22-28. In my opinion, a POSITA would have understood that 2K or 4K would have designated the memory size of the SRAM, and that the SRAM of either size would have had a plurality of storage locations. Indeed, a POSITA would have understood and found obvious that computer memory such as SRAM would have had a plurality of storage locations (e.g., sectors, cells, and so on). For example, *Hag* (Ex. 1048) teaches “a 4K SRAM could comprise a 64x64 array of cells[,]” and “an SRAM 10 includes an array 12 of memory cells[,]”. *Hag* (Ex. 1048), 1:54-55, 1:21-28 (“two-dimensional array of memory cells[,]” “[e]ach cell is capable of storing one bit of information[,]”) 4:21-22.

209. It is also my opinion that a POSITA would have been motivated and found it obvious to modify the *Smyth* combination to implement *Smyth*’s state memory 342 using an SRAM per *Mozer*’s teachings. Indeed, a POSITA would have understood and found obvious that SRAM was both fast and consumed low power. For example, *Hag* (Ex. 1048) teaches that “[a]mong the fastest integrated circuit memories are static random access memories (SRAMs), so named because they avoid the requirement of a refresh cycle used in the generally denser, but slower, dynamic random access memories (DRAMs). SRAMs typically include a two-dimensional array of memory cells arranged in rows and columns. Each cell is capable of storing one bit of information.” *Hag*, 1:21-28. In another example, *Yang*

462 (Ex. 1063) also teaches such. *Yang* 462, 1:14-24 (“Static RAMs (SRAM) are fast, low power and easier to use in the microprocessor system ... they require no refresh circuit.”). Additionally, *Akamatsu* teaches the same. *Akamatsu* (Ex. 1062), 12:32-35 (“Fast, low-power SRAM chips can be realized easily, in other words high-speed SRAMs driven by batteries in portable equipment can be provided.”). Therefore, a POSITA would have been motivated and found it obvious to implement *Smyth*’s state memory 342 using an SRAM. Such a use of a known technique would have advantageously improved *Smyth*’s memory in the same way as *Mozer*’s. Such a modification would have had a reasonable expectation of success because *Mozer* expressly teaches using an SRAM to store vector models for performing distance calculations. Such a modification would have been straightforward and would not have required undue experimentation because SRAM was well known in the prior art.

**C. Dependent Claim 24**

- 1. Claim 24: “The acoustic coprocessor of claim 23, wherein the SRAM and the calculating apparatus are fabricated on a single integrated circuit.”**

210. See Claims 1(e), 23 above. *Smyth-Mozer* teaches SRAM memory 460 and vector processor 423 are fabricated on a single integrated circuit (i.e., *Mozer*’s IC 400). *Mozer*, 7:53-55. I provided a motivation to combine *Smyth* and *Mozer* at Claim 1(e) above.

**XIII. OPINIONS REGARDING GROUND 5: CLAIMS 25-28 AND 37 ARE OBVIOUS OVER *SMYTH, MOZER, NGUYEN, VENSKO, AND COMEAU***

**A. Dependent Claim 25**

1. *Claim 25: "The acoustic coprocessor of claim 22, further comprising a status register, wherein the acoustic coprocessor is capable of setting one or more bits in the status register to values that indicate the completion of calculation of the set of distances."*

211. See Claim 22 above. In my opinion, *Comeau* (Ex. 1044) teaches a coprocessor mailbox/FIFO status register 102 (i.e., status register). Register 102 notifies the host core processor 40 when the host FIFO 76 that is used to transmit data from the coprocessor core 48 to the host processor core 40 is full, empty, etc.

Per *Comeau*:

One or more registers 90 are provided which are memory mapped to the host processor core 40 for storing mailbox/FIFO status information. Similarly, one or more registers 102 are provided which are memory mapped to the coprocessor core 48 for storing mailbox/FIFO status information. The following state information may be maintained in the mailbox/FIFO status registers 90, 102 for the respective FIFOs 74, 76:

1. full
2. empty
3. overflow error
4. underflow error

The HPI 46 is capable of generating interrupts to the host processor core 40 on selected events related to FIFO operation. For example, the host processor core 40 can be interrupted when:

- 1.—host FIFO 76 is not empty or has reached a RX interrupt threshold;
- 2.—coprocessor FIFO 74 is not full or has reached a TX interrupt threshold;
- 3.—host FIFO 76 underflows (caused by reading an empty FIFO) or coprocessor FIFO 74 overflows (caused by writing to a full FIFO). In this example, the underflow and overflow events are reported through the mailbox/FIFO error interrupt described below in the context of the host interrupt status and host interrupt control registers, and the host mailbox/FIFO status register. Data written to a full FIFO is discarded. Data read from an empty FIFO is undetermined.

*Comeau*, [0153]-[0161], [0151]-[0152], [0168], [0181]-[0182].

212. *Comeau* further teaches coprocessor mailbox/FIFO status register 102 includes a `c_hfifo_empty` bit to indicate Host FIFO 76 is empty and a `c_hfifo_full` bit to indicate Host FIFO 76 is full, as depicted in Table 5 below.

TABLE 5

| Coprocessor Mailbox/FIFO Status Register |                            |        |             |                       |
|------------------------------------------|----------------------------|--------|-------------|-----------------------|
| Bits                                     | Name                       | Access | Reset State | Description           |
| 15:12                                    | —                          | R      | 0           | Reserved              |
| 11                                       | <code>c_hfifo_empty</code> | R      | 1           | Host FIFO 76 is empty |

TABLE 5-continued

| <u>Coprocessor Mailbox/FIFO Status Register</u> |               |        |             |                                                                                                                                                                              |
|-------------------------------------------------|---------------|--------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bits                                            | Name          | Access | Reset State | Description                                                                                                                                                                  |
| 10                                              | c_hfifo_full  | R      | 0           | Host FIFO 76 is full. A write while this bit is asserted causes an overflow error (i.e., keep writing FIFO until this bit asserts. Can also be configured for an interrupt). |
| 9                                               | c_hfifo_uflow | R/W    | 0           | Asserted when the host performs a read on an empty host FIFO 76 (raises FIFO error interrupt) . Cleared by writing a 1 to this bit.                                          |
| 8                                               | c_hfifo_oflow | R/W    | 0           | Asserted when the coprocessor processor writes to the host FIFO 76 when the host FIFO 76 is full. Raises FIFO error interrupt. Cleared by a 1 written to this bit.           |

*Comeau*, [0342].

213. In my opinion, when modifying the *Smyth* combination per *Comeau*'s teachings, *Smyth*'s acoustic coprocessor has completed its distance calculation, and the acoustic coprocessor makes the data available to parsing processor 351 to perform word identification using the calculated distances. When modifying the *Smyth* combination per *Comeau*'s teachings, the coprocessor would have notified parsing processor 351 of sequencer 35 that distance calculations were done via the

FIFO (First In First Out memory - coprocessor puts data in, host reads the data out).

As long as the parsing processor 351 is keeping up with the distance calculations output by classifier 34, the FIFO is usually empty, as the distance(s) have been read out by the host, and the `c_hfifo_empty` bit will be 1. When the coprocessor finishes a calculation and puts data in the FIFO, this bit will change to 0 (FIFO is no longer empty, i.e., data is available).

214. Thus, the FIFO is initially empty, and so the bit in the status register is 1. Then, the first frame is processed and distance results are put in the FIFO. The coprocessor then changes the “empty” bit to be 0, which indicates that the FIFO is not empty. A POSITA would have understood or at least found obvious that this indicates the calculation is complete. Then, sequencer 35 (comprising parsing processor 351) reads the FIFO, which causes the empty bit to again be set to 1, as data has been read out of the FIFO. The process repeats with the next frame processed, distance data placed in the FIFO, and the empty bit is changed to 0 again.

215. In my opinion, a POSITA would have been motivated and found it obvious to modify the *Smyth* combination to include a status register such as taught by *Comeau*. Such a use of a known technique of setting a bit in a status register would have advantageously allowed parsing processor 351 to be informed when more distance calculations were available for performing word identification. Such a modification would also have advantageously allowed the modified *Smyth* to

provide a “flexible mechanism for status information to be shared with the host [processor] 40.” *Comeau*, [0201]. Therefore, such a use of a known technique of employing a status register would have advantageously improved the modified *Smyth*’s speech recognition system in the same way as *Comeau*’s system. Further, such a modification would have had a reasonable expectation of success because *Comeau* also teaches a speech recognition system. *Comeau* teaches a “speech recognition manager to synchronize the management of the resources between the host processor and the coprocessor. Each of these resource manager/observables comprises a software component on the coprocessor and a peer component on the host.” *Comeau*, [0274].

216. Further, a POSITA would have had a reasonable expectation of success when making such a modification because status registers were well known in the prior art and implementing a status register in the *Smyth* combination would have been straightforward and would not have required undue experimentation.

**B. Dependent Claim 26**

1. ***Claim 26: “The acoustic coprocessor of claim 25, wherein the acoustic coprocessor further comprises a direct memory access (DMA) controller capable of copying distances from the one or more result memories to a buffer memory.”***

217. See Claim 25 above. As I opined above, *Smyth-Mozer* in view of *Nguyen* teaches a result memory. See Claim 7 above. In my opinion, a POSITA would have understood or at least found obvious that *Smyth*’s state probability

memory 353 was a buffer memory, as claimed. The '789 Patent describes a buffer memory as a memory storing distance calculation results. '789 *Patent*, 6:9-23, 34:13-31. *Smyth*'s memory 353 both receives and stores state probability distance calculation results from classifier 34 and parsing processor 351 retrieves said distances to perform a word search process. *Smyth*, 6:10-42, Figs. 3-4.

218. Further, I note that the *Smyth* combination in view of *Vensko* would have taught a DMA controller, as claimed. *Vensko* teaches "a plurality of processors doing template comparisons of speech data. Each processor has an associated memory shared with the other processors by direct memory access (DMA) through a shared data bus." *Vensko*, Abstract, 7:48-53. I note that *Vensko* also teaches that "DMA transfers may be used to transfer character distance vector data from end processor 110 to each of the template processors 130, 140 ... 150; to transfer template data among the template processors 130, 140, ... 150; or to transfer template processor results data to the master processor." *Vensko*, 10:3-9, 10:19-31, Figs. 4, 5, 7.

219. In my opinion, a POSITA would have been motivated and found it obvious to modify the *Smyth* combination to include a DMA controller because DMA controllers allow memory transfers without requiring the processor to read and write each of the values being transferred. Rather, hardware such as the DMA controller would have performed this task with minimal load on the processor. Such

a use of a known technique would thus have improved *Smyth*'s similar speech recognition system in the same way as *Vensko*'s. It is also my opinion that a POSITA would have had a reasonable expectation of success when making such a modification to the *Smyth* combination because DMA controllers were well known and understood in the prior art, and modifying *Smyth* to include a DMA controller would have been straightforward. To the extent adding a DMA controller would have added additional cost to *Smyth*, a POSITA would still have been motivated and found it obvious to include a DMA controller to take advantage of reducing processor load on *Smyth*'s DSPs/processors to improve speech recognition system efficiency.

**C. Dependent Claim 27**

- 1. Claim 27: “The acoustic coprocessor of claim 26, wherein the DMA controller is capable of copying distances in a plurality of sequential bursts.”**

220. See Claim 26 above. A POSITA would have understood that a DMA controller such as taught by *Vensko* would have been capable of copying distances in a plurality of sequential bursts, as claimed. *Vensko* teaches performing DMA transfers, for example. *Vensko*, 11:36-43, 9:56-10:31. In my opinion, a POSITA would have understood and found obvious that doing DMA as big transfers or small bursts is entirely a question of software programming. A DMA controller is loaded with a source address, a destination address, and the number of bytes to transfer. The

DMA controller then transfers those bytes. Based on the DMA controller's programming, the DMA controller could transfer, for example, 500 bytes at once or transfer 500 bytes in 10 transfers of 50 bytes. A POSITA would have understood and found obvious that choosing the sizes of the number of bytes to transfer is simply a design decision based on observed bus traffic. Therefore, in my opinion, the *Smyth* combination as modified by *Vensko* teaches this claim.

221. Further, a POSITA would have understood or at least found obvious that performing sequential bursts for copying distances was known in the art, as taught by *Carmon*.

In the present example, the DMA controller arbiter will first find the packet request that asks for 256 words to be moved for the speech recognition task. It will load up the indicated control words from the packet request to see how many words to move and will load the modulo addressing control boundaries. It will then read the host system memory address and the DSP memory address and arbitrate for the host memory bus. Once a grant to the host memory bus is received, the DMA controller and processor will burst a number of bytes (16 bytes in this embodiment) of data from the host system processor memory to its own internal buffer. It will then arbitrate for the DSP bus and each time it receives a grant, it will write another word into the DSP memory according to the DSP beginning memory address which it read from the DMA packet control request. After all 16 bytes have been written, it will go back to arbitrate for the host system memory bus again. The process will continue until all 256 words have been moved. Then the

DMA machine will read the next packet in the DMA packet transfer request list.

*Carmon* (Ex. 1049), 7:21-40.

The DMA handler 11 breaks down the packet byte count indicated in the DMA packet control words into a number of bursts of data. Depending upon the direction of transfer, as indicated by the direction of transfer bit in the DMA packet request, the DMA handler 11 repetitively initiates either the host system bus master controller 12 or the DSP bus master controller 13 to cause arbitration for the appropriate buses and transfer of the appropriate size burst of data to or from the beginning addresses as noted in the DMA request packet read from the DSP memory. When the affected DMA arbiter and controller completes the DMA transfer, it signals the DMA handler controller 11 which signals either a new burst or packet or, due to keeping count, when it realizes it has exhausted the requirement of the packet by count request and acknowledges its completion back to the packet controller 10. The packet controller 10 will then read in another set of packet control words for another DMA operation and the process will continue until the DSP's DMA packet transfer control list partition has been traversed and a "wait state" packet has been encountered.

*Carmon*, 14:5-24.

DMA transfers take place utilizing the burst count (bc). The bc is equal to the burst size except when the remaining or beginning packet count is less than the burst size. The packet byte count is repetitively broken down into bursts until it has been exhausted.

*Carmon*, 15:28-32.

222. Further, *Wilcox* expressly teaches a DMA controller transmitting data in a plurality of bursts and that such was done “in a known manner.”

**Both of these transfers of data are generally made under the control of direct memory access (DMA) controllers in a known manner.** A DMA data transfer takes place independently of the program being executed by the processor (either host or local), and can occur at the full burst throughput rate of the host system bus. DMA transfers are generally initiated by providing the DMA controller with: (1) a starting address of the location in the host processor memory containing the data to be sent to the disk drive system (termed the source address); (2) either the amount of data to be transferred, or the ending address the data in the host processor memory (the end address); and (3) the I/O address of the disk drive controller to which it is desired to send the data (the destination address). The DMA controller then requests the host processor to grant it access to the system bus. **When access is granted, the DMA controller enables a burst of data of a predetermined size to be transferred by accessing the host processor memory.** More specifically, the DMA controller conditions the host memory to place data on the system bus, by placing a memory address on the system bus, conditioning the host memory to place data from the addressed location on the system bus, and simultaneously conditioning the disk drive system to store the data from the system bus in it's [sic] local memory. **When this transfer is complete, the address is incremented, and the DMA cycle repeated until all the desired data has been transferred, all in a known manner.**

*Wilcox* (Ex. 1065), 1:50-2:9.

In general, a block of data is transferred between the memory 56 and the external interface 58 as a series of smaller DMA burst transfers. Each DMA burst transfer transfers a predetermined amount of data. DMA burst transfers are repeated until the entire data block has been transferred.

*Wilcox*, 4:65-5:3.

223. Additionally, *Pomerantz* teaches a DMA controller transmits data in a plurality of bursts.

According to one aspect of the present invention, a method is provided in which a device, in response to a read request issued by a host, transfers data to the host through a series of direct memory access (DMA) data in bursts. The host is allowed to interrupt the read command and determine whether a first portion has been transferred correctly from the device based upon an error code calculation transmitted from the device to the host during the termination phase of a data in burst.

*Pomerantz* (Ex. 1066), 2:37-45.

224. Therefore, a POSITA would have understood and found obvious that DMA controllers, such as taught by *Vensko*, were capable of copying distances in a plurality of sequential bursts based on design preferences, including observed bus traffic. As I noted above, a POSITA would have been motivated and found it obvious to modify the *Smyth* combination to include a DMA controller per *Vensko's*

teachings and would have had a reasonable expectation of success for such a modification. *See* Claim 26 above.

**D. Dependent Claim 28**

1. *Claim 28[Pre]: “A speech recognition system comprising the acoustic coprocessor of claim 26, wherein:”*

225. *See* Claims 26, 1(Pre), 10(Pre) above.

2. *Claim 28(a): “the speech recognition system further comprises at least one processor configured to execute software to perform or initiate a speech recognition word search using one or more distances calculated by the acoustic coprocessor in response to receiving a first feature vector, and”*

226. *See* Claims 16(Pre), 16(a) above.

3. *Claim 28(b): “a time period for the speech recognition word search overlaps with a time period for the acoustic coprocessor to calculate one or more distances in response to receiving a second feature vector.”*

227. *See* Claim 16(b) above.

**E. Dependent Claim 37**

1. *Claim 37: “A speech recognition system comprising the acoustic coprocessor of claim 27, the speech recognition system further comprising at least one microphone.”*

228. *See* Claim 27. *Smyth* teaches a microphone. *Smyth*, 1:24-32, 4:22-24, 4:38-40, 4:46-67, Fig. 1.

**XIV. OPINIONS REGARDING GROUND 6: CLAIM 29 IS OBVIOUS OVER  
SMYTH, MOZER, NGUYEN, VENSKO, COMEAU, AND BOIKE**

**A. Dependent Claim 29**

1. *Claim 29: “The speech recognition system of claim 28, wherein the acoustic coprocessor and the at least one processor are fabricated on a single integrated circuit.”*

229. See Claim 28 above. I note that *Smyth* teaches that classifier processor 341 may be a DSP and parsing processor 351 may be either a DSP or a microprocessor. *Smyth*, 6:5-8, 6:45-51. I note that *Kloker* teaches the DSP56000 expressly taught by *Smyth* “can be used as a very fast microprocessor in any application that requires high-speed calculations or real-time response.” *Kloker* (Ex. 1009), 1. Further, using DSPs allowed for flexibility, consistency, and higher integration, as taught by *Mariani*, and I note *Mariani* also discloses the Motorola DSP56000. *Mariani* (Ex. 1013), 9-10.

230. 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

Declaration of Chris Schmandt  
Patent No. 10,839,789

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).

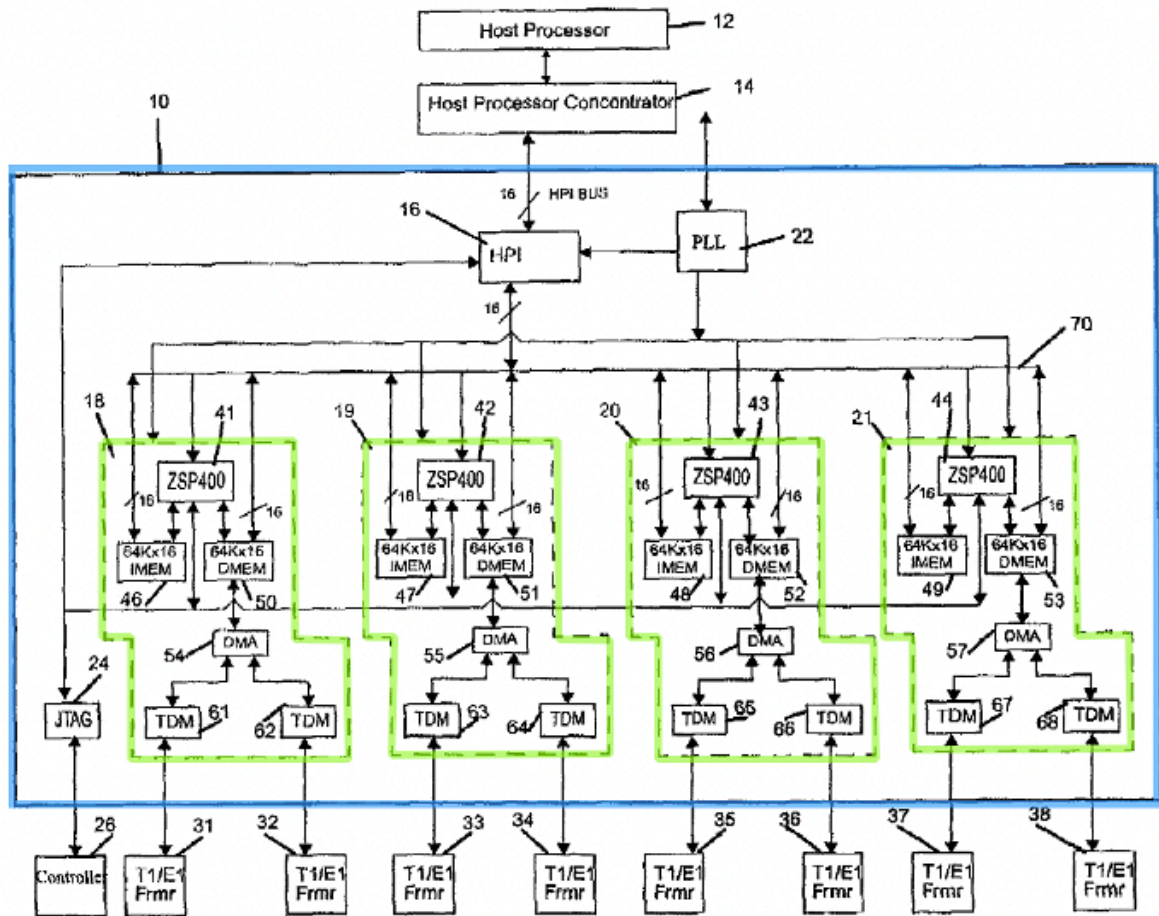


FIGURE 1

*Boike*, Fig. 1 (annotated).

231. 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.”).

232. In my opinion, a POSITA would have been motivated and found it obvious to modify the *Smyth* combination to include its acoustic coprocessor and at least one processor 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 *Smyth* combination 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/processors 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 *Smyth*. Further, a POSITA would have understood that the use of a technique of placing multiple DSPs/processors on a single IC would have advantageously improved the modified *Smyth*’s system in the same way as *Boike*’s system.

233. 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/processors 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[.]”

**XV. OPINIONS REGARDING GROUND 7: CLAIM 35 IS OBVIOUS OVER  
*SMYTH, MOZER, NGUYEN, VENSKO, COMEAU, AND WHITTAKER***

**A. Dependent Claim 35**

1. *Claim 35: “The acoustic coprocessor of claim 26, wherein the acoustic model includes parameters represented with integer data.”*

234. See Claim 26 above. I note that the ’789 Patent describes integers only in the context of lexical trees. ’789 Patent, 18:10-16 (“Each lexical tree comprises

lexical tree elements, and each element has three fields. For now, the fields will be integers and will be converted to `std_logic_vector` later.”), 18:20-54.

235. In my opinion, *Whittaker* (Ex. 1041) also teaches converting an array of integers into a tree structure in the same way as described in the '789 Patent. *Whittaker*, Abstract, 2:20-28, 2:33-44, 3:8-15, 3:27-32 (“the method for compressing according to the invention converts each ordered array of integers into a tree structure[.]”).

236. In my opinion, it would have been obvious to a POSITA and a POSITA would have been motivated to modify the *Smyth* combination per *Whittaker's* teachings. For example, *Whittaker* teaches “obtain[ing] an 86% reduction in the size of the language model with no increase in the word error rate.” *Whittaker*, 2:30-32. A POSITA would have understood that modifying the *Smyth* combination would have captured *Whittaker's* advantages. Further, such a modification would have been use of a known technique of representing acoustic model parameters with integer data and would have advantageously improved the *Smyth* combination's system in the same way as *Whittaker's*. Such a modification would have had a reasonable expectation of success because such technique was known in the prior art and would have required only minor changes in software or programming to achieve. Thus, such a modification would have been straightforward and would not have required undue experimentation.

**XVI. OPINIONS REGARDING GROUND 8: CLAIMS 38-39 ARE OBVIOUS OVER *SMYTH*, *MOZER*, *NGUYEN*, *VENSKO*, *COMEAU*, AND *ICHIKAWA***

**A. Dependent Claim 38**

- 1. Claim 38: “A speech recognition system comprising the acoustic coprocessor of claim 27, the speech recognition system further comprising a plurality of microphones.”**

237. See Claims 27, 37. *Smyth* teaches a microphone. *Smyth*, 1:24-32, 4:22-24, 4:38-40, 4:46-67, Fig. 1. *Ichikawa* (Ex. 1042) teaches using a plurality of microphones in a speech recognition system and that using a microphone array of a plurality of microphones advantageously eliminates noise. *Ichikawa*, 1:8-10, 1:28-29, 1:40-42, 3:63-66, 8:66-67, 20:65-67, 22:11-22, Figs. 17, 18.

238. A POSITA would have been motivated and found it obvious to modify the *Smyth* combination to utilize a plurality of microphones as taught by *Ichikawa* to achieve such noise reduction. Such a use of a known technique of employing a plurality of microphones would have advantageously improved the *Smyth* combination in the same way as *Ichikawa*. Such a modification would have had a reasonable expectation of success because such was known in the art and using two or more microphones would have been a straightforward and simple modification to implement. To the extent any cost increase would have occurred from such a modification, a POSITA would still have been motivated to make such a modification and would have understood or at least found obvious that such modification would have improved speech recognition results and therefore such

potential cost increase would have been offset by improvements to the speech recognition system.

**B. Dependent Claim 39**

1. *Claim 39: “The speech recognition system of claim 38, further comprising the buffer memory.”*

239. *See Claims 26, 38 above.*

**XVII. OPINIONS REGARDING GROUND 9: CLAIMS 42-44 ARE OBVIOUS OVER *SMYTH*, *MOZER*, *NGUYEN*, *VENSKO*, *COMEAU*, *ICHIKAWA*, AND *SESHADRI***

**A. Dependent Claim 42**

1. *Claim 42: “A laptop or tablet personal computer comprising the speech recognition system of claim 39.”*

240. *See Claim 39 above.* First, I note that for all of claims 42-45 that it was well known in the prior art that speech recognition systems were run in an assortment of form factors and devices. *Jiang* teaches such. *Jiang*, 4:60-65. Using specific form factors and devices as I opine in Claims 42-45 would have advantageously captured additional market share, and thus a POSITA would have been motivated and found it obvious to modify the modified *Smyth* combination to utilize each of the specific forms/devices of Claims 42-45. Further, such combinations would have had a reasonable expectation of success because incorporating speech recognition systems into numerous devices and forms was well known. *See Claims 42-45.*

241. *Seshadri* (Ex. 1043) teaches implementing a speech recognition system using a laptop computer. *Seshadri*, 3:60-65.

242. Modifying the *Smyth* combination to be implemented using a laptop computer would have advantageously captured additional market share. Therefore, a POSITA would have been motivated and found it obvious to implement the *Smyth* combination using a laptop computer per *Seshadri*'s teachings. Such a modification would have had a reasonable expectation of success because implementing speech recognition systems on laptop computers was known in the art as taught by *Seshadri*.

**B. Dependent Claim 43**

**1. Claim 43: "A mobile phone comprising the speech recognition system of claim 39."**

243. See Claim 39 above. First, I note that for all of claims 42-45 that it was well known in the prior art that speech recognition systems were run in an assortment of form factors and devices. *Jiang* teaches such. *Jiang*, 4:60-65. Using specific form factors and devices as I opine in Claims 42-45 would have advantageously captured additional market share, and thus a POSITA would have been motivated and found it obvious to modify the modified *Smyth* combination to utilize each of the specific forms/devices of Claims 42-45. Further, such combinations would have had a reasonable expectation of success because incorporating speech recognition systems into numerous devices and forms was well known. See Claims 42-45.

244. *Seshadri* (Ex. 1043) teaches implementing a speech recognition system using a mobile phone. *Seshadri*, 3:22-24, 3:28-31, 3:34-36, 3:55-57, 3:60-65.

245. Modifying the *Smyth* combination to be implemented using a mobile phone would have advantageously captured additional market share. Therefore, a POSITA would have been motivated and found it obvious to implement the *Smyth* combination using a mobile phone per *Seshadri's* teachings. Such a modification would have had a reasonable expectation of success because implementing speech recognition systems on mobile phones was known in the art as taught by *Seshadri*, and therefore the minor hardware modifications needed were already known in the art.

**C. Dependent Claim 44**

**1. Claim 44: “An electronic entertainment product comprising the speech recognition system of claim 39.”**

246. See Claim 39 above. First, I note that for all of claims 42-45 that it was well known in the prior art that speech recognition systems were run in an assortment of form factors and devices. *Jiang* teaches such. *Jiang*, 4:60-65. Using specific form factors and devices as I opine in Claims 42-45 would have advantageously captured additional market share, and thus a POSITA would have been motivated and found it obvious to modify the modified *Smyth* combination to utilize each of the specific forms/devices of Claims 42-45. Further, such combinations would have had a reasonable expectation of success because incorporating speech recognition systems into numerous devices and forms was well known. See Claims 42-45.

247. *Seshadri* (Ex. 1043) teaches “the speech recognition system using audio signals” “can be configured on a variety of electronic device[s], either mobile or stationary. For instance, the improved speech recognition system of the invention can be configured on at least but not limited to a laptop computer, a PDA, an audio/video recording device, a home computer, a game console, a remote controller, or other comparable device.” *Seshadri*, 3:54-65. In my opinion, a POSITA would have understood or at least found obvious that any of a home computer, game console, audio/video recording device, and/or PDA would have been an “electronic entertainment product,” as claimed, because all such devices are electronic and used for entertainment (e.g., consuming various media, playing games).

248. Modifying the *Smyth* combination to be implemented using an electronic entertainment product would have advantageously captured additional market share. Therefore, a POSITA would have been motivated and found it obvious to implement the *Smyth* combination using an electronic entertainment product. Such a modification would have had a reasonable expectation of success because implementing speech recognition systems on electronic entertainment products was known in the art as taught by *Seshadri*.

**XVIII. OPINIONS REGARDING GROUND 10: CLAIM 45 IS OBVIOUS OVER *SMYTH*, *MOZER*, *NGUYEN*, *VENSKO*, *COMEAU*, *ICHIKAWA*, AND *STEINBISS***

**A. Dependent Claim 45**

**1. Claim 45: “An appliance comprising the speech recognition system of claim 39.”**

249. See Claim 39 above. First, I note that for all of claims 42-45 that it was well known in the prior art that speech recognition systems were run in an assortment of form factors and devices. *Jiang* teaches such. *Jiang*, 4:60-65. Using specific form factors and devices as I opine in Claims 42-45 would have advantageously captured additional market share, and thus a POSITA would have been motivated and found it obvious to modify the modified *Smyth* combination to utilize each of the specific forms/devices of Claims 42-45. Further, such combinations would have had a reasonable expectation of success because incorporating speech recognition systems into numerous devices and forms was well known. See Claims 42-45.

250. I note the '789 Patent describes “appliance” only once outside of the claims. '789 Patent, 37:58 (“domestic appliances”).

251. *Steinbiss* (Ex. 1045) teaches implementing a speech recognition system using an appliance, such as television sets, video records, hi-fi music systems, kitchen appliances or other appliances with speech control. Per *Steinbiss*:

Other areas of application include the control of electronic consumer appliances such as television sets, video recorders, hi-fi music systems, kitchen appliances or other appliances with speech control, in which the

various appliance functions can be acted upon by means of speech, e.g. volume control, the choice of a television channel or radio station or the adjusting of the motor speed of a kitchen appliance.

*Steinbiss*, [0001]-[0002]. Therefore, in my opinion, the *Steinbiss* teaches a “domestic appliance” with speech recognition functionality.

252. Modifying the *Smyth* combination to be implemented using an appliance would have advantageously captured additional market share. Therefore, a POSITA would have been motivated and found it obvious to implement the *Smyth* combination using an appliance. Such a modification would have had a reasonable expectation of success because implementing speech recognition systems on appliances was known in the art as taught by *Steinbiss*.

## **XIX. CONCLUSION**

253. For the aforementioned reasons, in my opinion, to a POSITA, the claims of the '789 Patent are rendered obvious by the prior art.

254. 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.

Declaration of Chris Schmandt  
Patent No. 10,839,789

Date: November 15, 2022

Respectfully submitted:

By:

A handwritten signature in black ink, appearing to read "Chris Schmandt", written over a horizontal line.

Christopher Schmandt

## **Christopher Schmandt**

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

### **Education**

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

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

### **Professional Experience -- MIT**

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

Architecture Machine Group, Research Associate, 1980-1984

Architecture Machine Group, Research Assistant, 1979-1980

Architecture Machine Group, Graphics Programmer, 1977-1979

Departmental Undergraduate Research Opportunities Program Coordinator, 1984-2018

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

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

### **Sponsored Research Activities**

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

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

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

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

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

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

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

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

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

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

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

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

**Professional Experience -- Intellectual Property**

Co-inventor on U.S. Patents

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

Co-inventor on U.S. Patent Applications

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

- 20030158903 "Calendar bar interface for electronic mail interaction"

## Publications

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

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

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

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

*Exploring Interactions and Perceptions of Kinetic Wearables.* DIS 2017. (with Cindy Hsin-Liu Kao, D. Ajilo, O. 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

*Understanding Speech Without Recognizing Words*, Proceedings, American Voice Input/Output Society Conference, AVIOS, 1987.

*A Robust Parser and Dialog Generator for a Conversational Office System* (with B. Arons and C. Simmons), Proceedings, American Voice Input/Output Society Conference, AVIOS, Palo Alto, CA, 1987.

*Integrated Messages and Network Services for a Personal Workstation*, IEEE Workshop on Telematics and Message Handling Systems, IEEE, 1986.

*Voice Interaction in an Integrated Office and Telecommunications Environment*, Proceedings, American Voice Input/Output Society Conference, AVIOS, San Francisco, CA, 1985.

*Voice Communication with Computers*, book chapter in *Advances in Human-Computer Interaction*, H. R. Hartson ed., 1985.

*Speech Synthesis Gives Voiced Access to an Electronic Mail System*, *Speech Technology*, Vol. 2, No. 3, Aug/Sept 1984.

*A Conversational Telephone Messaging System* (with B. Arons), *IEEE Transactions on Consumer Electronics*, IEEE, Vol CE-30, August 1984.

*Phone Slave: A Graphical Telecommunications Interface* (with B. Arons), Proceedings, Society for Information Display International Symposium, SID, San Francisco, CA, June 1984.

*Input/Display Registration in a Stereoscopic Workstation*, *Displays*, April 1984.

*Remote Access to Voice and Text Messages*, Proceedings, American Voice Input/Output Society Conference, AVIOS, Washington D.C., 1984.

*Fuzzy Fonts: Analog Models Improve Digital Text Quality*, Proceedings, National Computer Graphics Association Conference, Chicago, IL, 1983.

*Greyscale Fonts Designed From Video Signal Analysis*, Society of Applied Learning Technology, Houston, TX, 1983.

*Spatial Input/Display Correspondence in a Stereoscopic Computer Graphic Work Station*, Proceedings, ACM/SIGGRAPH, Detroit, MI, 1983.

*A Programmable Virtual Vocabulary Speech Processing Peripheral* (with W. Bender), Proceedings, American Voice Input/Output Society Conference on Voice Data Entry Systems Applications, AVIOS, 1983.

*The Intelligent Voice Interactive Interface* (with E. A. Hulteen), Proceedings, Human Factors in Computer Systems, National Bureau of Standards/ACM, Gaithersburg, MD, 1982.

*Interactive Three-Dimensional Computer Space*, Proceedings, SPIE Conference on Processing and Display of Three-Dimensional Data, SPIE, San Diego, CA, 1982, Vol. 367.

*Speech Communications, a Systems' Approach*, Proceedings, American Voice Input/Output Society Conference on Entry Systems Applications, 1982.

*Voice Interaction: Putting Intelligence into the Interface*, Proceedings, IEEE International Conference on Cybernetics and Society, IEEE, Seattle, WA, 1982.

*The Intelligent Ear: A Graphical Interface to Digital Audio*, Proceedings, IEEE International Conference on Cybernetics and Society, IEEE, Atlanta, GA, 1981.

*Soft Typography*, Information Processing 1980, IFIPS, S. Lavington ed., North-Holland Publishing Co., 1980.

---