Entwicklung und Performance eines Sprachdialogsystems mit begrenzten Hardwareressourcen

Frank Duckhorn, Guntram Strecha, Matthias Wolff, Rüdiger Hoffmann

Institut für Akustik und Sprachkommunikation
frank.duckhorn@ias.et.tu-dresden.de

Kurzfassung: Die Integration von Spracherkennung und -synthese auf Systeme mit begrenzten Hardwareressourcen wird immer häufiger benötigt. Wir haben ein Dialogsystem entwickelt, welches auf einer Kombination aus einem digitalen Signalprozessor (DSP) und einem Field Programmable Gate Array (FPGA) lauffähig ist. Es wurde versucht die Verluste in der Erkennungsleistung sowie der Synthesequalität möglichst gering zu halten. Um Speicherplatz zu sparen, verwenden Erkenner und Synthese die selben, sprecherunabhängigen Hidden-Markov-Modelle. Der Spracherkenner ist phonembasiert und kann beliebige kontextfreie Grammatiken verarbeiten sowie zwischen verschiedenen Grammatiken wechseln. Die Synthese basiert auf der Verkettung von HMM-kodierten Einheiten. Die Einheiten bestehen aus einer Folge von Indizes der Verteilungsdichten der HM-Modelle des Spracherkenners, einer F0- sowie einer Intensitäts-Kontur. Zur Generierung einer Zielstimme werden die sprecherunabhängigen Syntheseparameter auf Merkmalebene konvertiert. Im Rahmen dieser Veröffentlichung werden wir Ergebnisse zur Leistungsfähigkeit, Performancemessung sowie zum Speicherbedarf des Dialogsystems vorstellen. Außerdem soll das entstandene System vorgeführt werden.

1 Projektziel

Dieser Beitrag stellt ein Projekt vor, dessen Ziel es ist, ein Sprachdialogsystem auf einem USB-Stick zu entwickeln. Das System soll zur Steuerung von Mess- und Prüfgeräten eingesetzt werden. Es kommuniziert über USB mit dem zu steuernden Gerät und bezieht möglichst ausschließlich von dort seine elektrische Leistung. Durch ein kabelgebundenes oder ein Bluetooth-Headset werden Sprachbefehle eingegeben sowie Rückmeldungen durch Sprachsynthese ausgegeben.

Im Beitrag [1] der letzten ESSV haben wir bereits einen Zwischenstand sowie Konzepte für die Spracherkennung und -synthese auf einer einfachen Hardware vorgestellt. Hier werden nun die notwendigen Änderungen und Erweiterungen der Konzepte sowie Messungen zur Performance des Systems präsentiert.

Hardwareseitig besteht das System aus einem digitalen Signalprozessor (TMS320C6727+) und einem FPGA (Xilinx LCA Virtex4). Der Signalprozessor verfügt über eine Gleintkommarecheneinheit sowie 512 kB internen und 16 MB externen Speicher. Aufgabe des Signalprozessors ist die Steuerung und Verwaltung des Systemablaufs sowie die Ausführung schlecht parallelisierbarer oder einfacher Algorithmenteile. Der FPGA hingegegen soll aufwendige Algorithmenteile parallel und schnell ausführen. Außerdem steuert er die Signalaufnahme und -ausgabe.

Über eine angeschlossene MirkoSD-Karte werden dem System Parameter sowie die Daten der Modelle für Spracherkennung und -synthese eingespielt und können so auch einfach getauscht werden.

2 Spracherkennung

2.1 Konzept und Merkmalgewinnung

Der verwendete Spracherkenner soll auf dieser Plattform (Signalprozessor und FPGA) in Echtzeit lauffähig sein. Außerdem soll er möglichst robust sein und von verschiedenen Sprechern ohne Adaption benutzt werden können.

Als primäre Merkmale werden die logarithmierten Werte einer MEL-Filterbank verwendet, welche aus dem fourier-transformierten Signal gewonnen werden. Diese werden für die eigentliche Spracherkennung wie auch für die Sprach-/Pausedetektion verwendet. So kann Rechenzeit gespart werden und es müssen nicht zwei verschiedene Algorithmen implementiert werden. Für das eingehende Audiosignal werden also permanent die primären Merkmale berechnet. Diese werden um ihre dynamischen Merkmale erweitert und durch eine Hauptkomponentenanalyse zu einem 24-dimensionalen sekundären Merkmalvektor für die Spracherkennung reduziert. Für die Sprach-/Pausedetektion werden keine dynamischen Merkmale verwendet, um Verzögerungszeiten des Algorithmus zu minimieren. Hier wird nur eine Hauptkomponentenanalyse angewendet. Der entstandene Merkmalvektor wird auf die 10 Merkmale der größten Varianz reduziert.

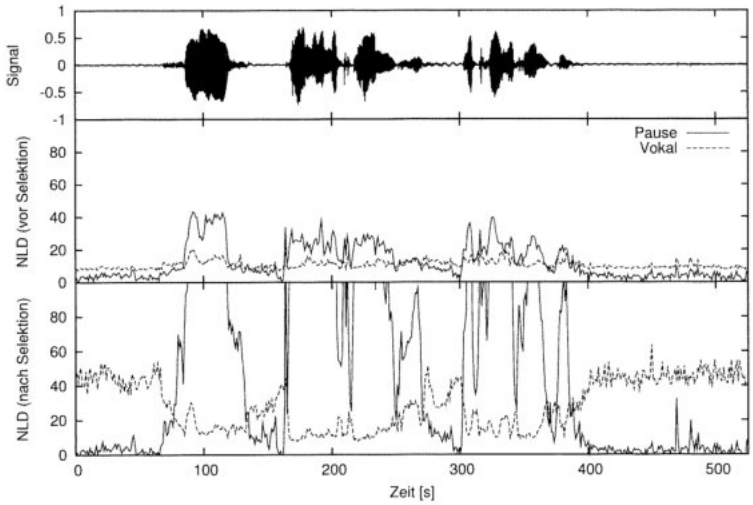


Abbildung 1 - Beispiel zur Sprach-/Pausedetektion mit negativ logarithmieren Wahrscheinlichkeitsdichten (NLD) der Mischverteilungsmodelle für Pause und Vokal vor und nach der Selektion von Normalverteilungen

2.2 Sprach-/Pausedetektion

Die Sprach-/Pausedetektion wird, wie auch die Spracherkennung selbst, mit statistischer Klassifikation umgesetzt. Sie basiert auf Mischverteilungsmodellen (Gaussian-Mixture-Models - GMMs). Da die gleichen Merkmale wie für die Spracherkennung verwendet werden, kann auch

des gleiche Trainingsverfahren benutzt werden. Nur werden hier keine Modelle pro Phonem erzeugt. Es hat sich als sinnvoll erwiesen, ein GMM für Vokale und eines für Pause und Störgeräusche zu trainieren. Die Modelle für die anderen Laute (Nasal-, Plusiv-, und Frikativlaute) wurden verworfen, da sie nicht so robust wie die Vokalmodelle arbeiten. Hier wurden häufiger Pause oder bestimmte Störgeräusche einer dieser Lautgruppen zugeordnet.

Die beiden generierten GMMs bestehen aus jeweils acht Normalverteilungen, wobei jede über eine vollständige Kovarianzmatrix verfügt. Trotz phonemweise gelableten Trainingsdaten wurde nicht verhindert, dass einzelne Normalverteilungen innerhalb des Vokalmodels auf kurze Pausenabschnitte im Trainingsmaterial trainiert wurden. Das konnte durch die Beobachtung der Zuordnung der Merkmalvektoren im Trainingsverlauf bestätigt werden. Umgekehrt tritt das gleiche Phänomen auch für das Pausenmodell auf. Schließlich wurden aus beiden GMMs jeweils vier Normalverteilungen ausgewählt, die für Vokal- beziehungsweise Pausenabschnitte in einer geringen Anzahl von Beispielaufnahmen die besten Wahrscheinlichkeitsdichten erzeugen.

In der Abbildung 1 ist eine Beispielaufnahme sowie die von beiden GMMs erzeugte negativ logarithmierte Wahrscheinlichkeitsdichte über der Zeitachse aufgetragen. Die Wahrscheinlichkeitsdichten sind jeweils vor und nach der Selektion der Modelle angegeben. Man kann sehen, dass die Diskriminativität der Modelle durch die Selektion deutlich steigt. Das Vokalmodell zeigt jetzt erst überhaupt deutlich Unterschiede zwischen Sprach- und Pausenabschnitten und die Unterschiede beim Pausenmodell werden stark vergrößert.

Mit diesen modifizierten Modellen kann nun in Echtzeit zu jedem Merkmalvektor eine Entscheidung für Sprache oder Pause getroffen werden. Leider ist die Folge dieser Entscheidungen oft von kleineren Lücken unterbrochen. Deswegen wurde ein endlicher Automat eingesetzt, der Parameter der Sprach-/Pausefolge, wie sie in der Tabelle 2.2 aufgelistet sind, erzwingt und damit die Folge glättet. Die minimale Pauselänge ist bewußt recht groß gewählt um kleiner Pausen zwischen Wörtern innerhalb eines Kommandos sicher überbrücken zu können. Durch diese Einstellungen benötigt der Automat eine Entscheidungsverzögerung von 20 Merkmalvektoren. Dies entspricht 200 Millisekunden.

Parametername	Anzahl der Merkmalvektoren
Minimale Sprachlänge	4
Minimale Pausenlänge	20
Maximale Sprachlänge	600
Sprachvorlauf	10
Sprachnachlauf	10

Tabelle 1 - Parameter für den endlichen Automaten der Sprach-/Pausedetektion

2.3 Spracherkennung mit endlichen Automaten

Die Spracherkennung wurde mit endlichen, gewichteten Automaten (Finite-State-Transducer) umgesetzt. Dieser Aufbau wird in den letzten Jahren im beliebter (siehe [2, 3]). Der Vorteil ist, dass alle verwendeten Modelle (Phonemmodelle in Form von Hidden-Markov-Modelle, Lexikon und Grammatik) als Automat dargestellt werden können. Diese einzelnen Automaten können mittels Komposition zu einem einzigen Automaten vereinigt werden. So besteht die Aufgabe der Spracherkennung auf der Hardware nur in der Dekodierung der Merkmalvektorfolge durch diesen einen Automat, das Erkennungsnetzwerk. Die Merkmalextraktion, wie sie

im Abschnitt 2.1 Überblicksweise beschrieben wurde, wird zusammen mit der Berechnung der Wahrscheinlichkeitsdichten der Mischverteilungsmodelle auf dem FPGA ausgeführt. Auf dem Signalprozessor läuft dann eine zeitvariante A* Suche, die den besten Pfad im Erkennungsnetzwerk bestimmt und damit das Erkennungsergebnis.

Der Spracherkenner kann reguläre Sprachen verarbeiten, die in Form von Grammatiken in das Erkennungsnetzwerk eingehen. Durch Umschalten zwischen verschiedenen abgespeicherten Erkennungsnetzwerken kann ein Sprachdialog realisiert werden, zum Beispiel für das Nachfragen bei Datenlöschungen.

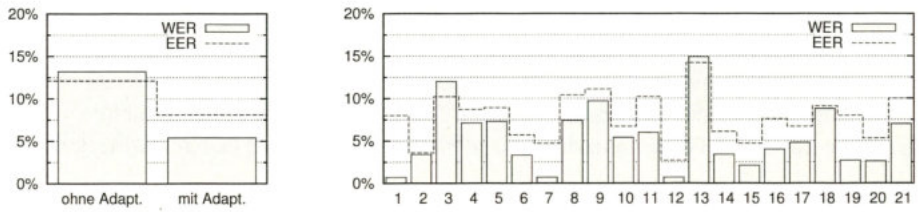
2.4 Performance

Die Merkmalextraktion sowie die Dichteberechnung für die Spracherkennung wie auch für die Sprach-/Pausedetektion bis hin zu den negativ logarithmierten Wahrscheinlichkeitsdichten der Mischverteilungsmodelle wird im FPGA in Echtzeit berechnet. Genauer benötigt der FPGA für ein Fenster ca. 8 ms bei einer Fortsetzrate der Fenster von 10 ms. Damit stehen diese Wahrscheinlichkeitsdichten vor Ablauf des jeweils nächsten Fensters dem Signalprozessor zur Verfügung.

Diese Wahrscheinlichkeitsdichten werden dem DSP übergeben und dort im Speicher gepuffert. Falls die Sprach-/Pausedetektion einen vollständigen Sprachabschnitt erkannt hat, wird mit den zugehörigen gepufferten Wahrscheinlichkeitsdichten die Suche gestartet. Diese bestimmt im aktiven Erkennungsnetzwerk den wahrscheinlichsten Pfad und damit das wahrscheinlichste Kommando. Für die Pufferung der Wahrscheinlichkeitsdichten sind bei 128 Mischverteilungsdichten und sechs Sekunden maximaler Sprachlänge 300 kB Speicher notwendig. Der Suchalgorithmus selbst benötigt dann je nach Kommandotyp zwischen 10 und 50 Prozent der Signallänge um das Erkennungsergebnis zu bestimmen.

Um die Erkennungsleistung des Spracherkenners zu bestimmen wurden zwei verschiedene Grammatiken benutzt. Die erste Grammatik besteht aus 15 Kommandos mit maximal einer dreistelligen Zahl mit einer Nachkommastelle und einer optionalen Einheit. Die Zweite besteht 12 Kommandos mit bis zu zwei solcher Zahlen. Die Erkennungsnetzwerke umfassen 1385 beziehungsweise 1379 Zustände und 2938 beziehungsweise 2980 Übergänge. Für jede der beiden Grammatiken existieren ungefähr zwei Stunden an Audiodaten. Die Daten der jeweils anderen Grammatik wurden als OOV Daten benutzt um die Leistung der Rückweisung zuverlässig zu bestimmen.

Durch Adaption auf das Headset und die Grammatik (nicht auf den Sprecher!) wurde versucht die Erkennungsleistung zu steigern.



(a) Gesamt

(b) pro Sprecher (mit Adapt.)

Abbildung 2 - Kommandofehlerrate (WER) und Equal-Error-Rate (EER) der Rückweisung

In der Abbildung 2 sind die Fehlerrate der Kommandoerkennung sowie die Equal-Error-Rate der Rückweisung dargestellt. Man kann sehen, dass die Adaption auf Headset und Grammatik die Erkennungsleistung deutlich steigert. Auch die Qualität der Rückweisung kann verbessert werden. Aus der Darstellung der Leistung pro Sprecher lässt sich ablesen, dass diese für einen Großteil Sprecher ähnliche Werte erreicht. Trotzdem gibt es Ausreißer. Einige Sprecher werden sehr gut erkannt, aber auch einige sehr schlecht. Ein Grund für diese Streuung konnte noch nicht gefunden werden. Außerdem sieht man, dass die Leistung der Rückweisung im wesentlichen der Erkennungsleistung folgt.

3 Sprachsynthese

Die Anforderungen an die Sprachsynthese hinsichtlich der Implementierung auf der Zielplattform sind ein geringer Speicher- und Rechenbedarf. Das Einsparen von zusätzlichem Speicher für die Synthese verringert einerseits die Kosten und andererseits die Stromaufnahme des USB-Gerätes, damit eine externe Stromversorgung vermieden werden kann. Um die Anforderungen zu Erfüllen, verwenden wir eine HMM-basierte Sprachsynthese, dessen HM-Modelle identisch mit denen des Spracherkenners sind. Der Spracherkenner ist sprecherunabhängig, d. h. die verwendeten Modelle sind unter Verwendung von Sprachdaten einer sehr großen Anzahl von Sprechern trainiert. Das hat zur Folge, dass die Synthese mit diesen Modellen eine gemittelte Stimme erzeugt. Um die vorgesehene Zielstimme zu erzeugen, ist es notwendig eine Stimmenkonvertierung durchzuführen.

Für unser Dialogsystem zur Steuerung eines Messgerätes ist die Menge der auszugebenden Sprachprompts begrenzt. Neben feststehenden Statusinformationen müssen dreistellige Messwerte mit Nachkommastellen und Messwert synthetisiert werden können. Um das zu realisieren, basiert die Sprachsynthese auf der Verkettung von Einheiten. Die Größe dieser Einheiten liegt im Silben- oder Wortbereich. Sie bestehen aus einer Folge von Indizes der Verteilungsdichten der HM-Modelle sowie einer Grundfrequenz- und Intensitätskontur und haben dementsprechend eine sehr geringen Speicherbedarf. Das Zufügen der Grundfrequenzkontur zu den Einheiten erspart die Implementierung der Algorithmen zur Prosodiegenerierung und des damit verbundenen Speicher- und Rechenbedarfs.

3.1 Generierung der Einheiten

Die Generierung der Einheiten erfolgt automatisiert. Die vom Zielsprecher gesprochenen Trägersätze enthalten alle für die Synthese notwendigen Einheiten. Diese Trägersätze werden von dem Spracherkenner unter Verwendung einer entsprechenden Grammatik in die Einheiten zerlegt. Der Erkenner erzeugt ebenfalls die Folge der Indizes der Verteilungsdichten sowie eine Ausgabewahrscheinlichkeit für jede Einheit. Diese Ausgabewahrscheinlichkeiten dienen zur Selektion der Einheit mit größter Erkennungssicherheit, falls alternative Einheiten in den Trägersätzen vorhanden sind. Zur Bestimmung der Grundfrequenzkontur der Einheiten werden die Trägersätze mit einem Periodenmarkierer analysiert und bei Bedarf manuell korrigiert.

3.2 Akustische Synthese

Für die akustische Synthese werden die Steuerkommandos des Gerätes bzw. die auszugebenden Messwerte in die Folge der Einheiten übersetzt. Diese Einheitenfolge wird anschließend verkettet und man erhält die Folge der Indizes für die vollständige Äußerung sowie deren

Grundfrequenz- und Intensitätskontur. Anhand der Indexfolge werden die Folgen der synthetisierbaren Parameter aus den Mittelwerten der entsprechenden Verteilungsdichten der HM-Modelle bestimmt. Diese Parameter sind noch sprecherunabhängig. Zur Stimmenkonvertierung werden die sprecherabhängigen Syntheseparameter $\tilde{\mathbf{P}}_\mathrm{D}$ aus den sprecherunabhängigen Parametern $\mathbf{P}_\mathrm{I}$ mittels der linearen Transformation nach Gleichung (1a) geschätzt [5].

$$\tilde{\mathbf{P}}_\mathrm{D} = \mathbf{P}_\mathrm{I}\mathbf{C}_\mathrm{D} \tag{1a}$$

Die Transformationsmatrix $\mathbf{C}_\mathrm{D}$ schätzen wir aus den Folgen der Verteilungsdichten $\mathbf{P}_\mathrm{I}$, welche sich entsprechend der Indexfolgen aller Einheiten des Inventars ergeben und den dazugehörigen originalen Merkmalvektoren $\mathbf{P}_\mathrm{D}$ der Trägerwörter:

$$\mathbf{C}_\mathrm{D} = \left(\mathbf{P}_\mathrm{I}{}^\mathrm{T}\mathbf{P}_\mathrm{I}\right)^{-1}\mathbf{P}_\mathrm{I}{}^\mathrm{T}\mathbf{P}_\mathrm{D} \tag{1b}$$

Diese Transformation konvertiert einerseits die Stimme und andererseits den Typ der Syntheseparameter. Die sprecherunabhängigen Parameter sind Mel-Filter-Spektren, die durch die Transformation in Line-Cepstrum-Quefrency (LCQ) Parameter [4] konvertiert werden. Nach der Konvertierung wird die Folge der Parametervektoren komponentenweise mit einem Tiefpass erster Ordnung geglättet. Ein Cepstrum-Synthesefilter, mit einem aus der Grundfrequenzkontur berechneten Anregungssignal, dient abschließend zur Synthese des Sprachsignals.

Literatur

[1] DUCKHORN, F., G. STRECHA, M. WOLFF und R. HOFFMANN: *Ein Sprachdialogsystem mit begrenzten Hardwareressourcen*. In: *Proc. Konferenz Elektronische Sprachsignalverarbeitung (ESSV)*, Bd. 32 d. Reihe *Studientexte zur Sprachkommunikation*, S. 88–93, Dresden, Germany, Sep. 2009. TUDPress.

[2] MOHRI, M., F. PEREIRA und M. RILEY: *Weighted finite-state transducers in speech recognition*. Computer Speech & Language, 16(1):69–88, 2002.

[3] MOHRI, M., F. PEREIRA und M. RILEY: *Speech recognition with weighted finite-state transducers*. In: BENESTY, Y. J. und M. SONDHI (Hrsg.): *Handbook of Speech Processing*, S. 559–582. Springer, 2008.

[4] STRECHA, G., M. EICHNER und R. HOFFMANN: *Line Cepstral Quefrencies and Their Use for Acoustic Inventory Coding*. In: *Proc. of Interspeech*, S. 2873–2876, Antwerpen, Aug. 2007.

[5] STRECHA, G., M. WOLFF, F. DUCKHORN, S. WITTENBERG und C. TSCHÖPE: *The HMM Synthesis Algorithm of an Embedded Unified Speech Recognizer and Synthesizer*. In: *Proc. of Interspeech*, S. 1763–1766, Brighton, Sep. 2009.