

Umsetzung eines binauralen Lokalisationsmodells als Echtzeit-Anwendung auf CPU und GPU

Mathias Schwyrz, Wolfgang Heß

Harman Becker Automotive Systems, 76307 Karlsbad, E-Mail: Mathias.Schwyrz@harman.com

Einleitung

Binaurale Lokalisationsmodelle schätzen die menschliche auditive Lokalisation. Basierend auf Jeffress [2] wurde von Lindemann [3] ein Kreuzkorrelationsmodell entwickelt, das psychoakustische Phänomene wie beispielsweise laterale Inhibition berücksichtigt. Dieser Modellalgorithmus ist sehr aufwendig in der Berechnung und konnte bisher nur offline simuliert werden, siehe [4]. Mittels GPGPU-Programmierung (General Purpose Computation on Graphics Processing Unit) [1] besteht die Möglichkeit, allgemeine Berechnungen auf die GPU auszulagern. Durch die damit gewonnene Rechenleistung können nun echtzeitfähige Implementierungen des Lokalisationsmodells realisiert werden. Dazu wurden Algorithmen zur parallelisierten Signalverarbeitung auf der GPU entwickelt, sodaß eine parallele Modellberechnung auf CPU und GPU ermöglicht wird.

Datenfusion sowie lernfähige Modellinterpretation bzw. Anpassung an individuelle Nutzer werden durch ein Bayes'sches Netz umgesetzt. Durch den Begriff der Bayes'schen Wahrscheinlichkeit (Unsicherheit, Belief) stellen sie ein geeignetes Instrument zur Interpretation von unsicheren Daten dar. Das Training des Netzes dient zur Anpassung von Lautheitsschätzung und Schätzungen von horizontaler Ausdehnung und Position von auditorischen Objekten auf der Lateralisationsachse an individuelle Nutzer.

System zur Modellberechnung am Rechner

Als funktioneller Rahmen wurde die Algorithmus-Umsetzung als VST-Plugin gewählt, womit ein Audiosignalingrouting zu der PC-Soundkarte mittels Blockverarbeitung erfolgt. NVIDIA liefert mit CUDA (Compute Unified Device Architecture) [1] eine API zur Programmierung der Grafikkarte. Mittels CUDA kann die GPU aus einem auf der CPU erstellten Programm als hochparallelisierter Co-Prozessor verwendet werden. Das Bayes'sche Netz wurde mittels NETICA-API [7] implementiert.

Echtzeit durch maximale Parallelisierung

Für diese Arbeit wurden Algorithmen zur Berechnung einer auditorischen Filterbank mit 24 Frequenzbändern mittels FIR- und IIR-Filter, basierend auf Patterson [6], für CPU und GPU implementiert. Neben der Parallelisierung der Filteralgorithmen wurde die gleichzeitige parallele Berechnungsmöglichkeit von GPU und CPU ausgenutzt. Die GPU rechnet Filterbank, Halbwellengleichrichtung und die Visualisierung der Benutzungsoberfläche, während die CPU das Signalingrouting von und zu der Soundkarte und den Datentransfer zur GPU steuert. Des weiteren werden das Bayes'sche Netz sowie der binaurale Prozessor des Lokalisationsmodells (Kreuzkorrelation, Inhibition, Korre-

lations-Azimuth-Transformation), siehe [4], auf der CPU gerechnet. Dabei wurde durch Multithreading die parallele Berechnung auf einer Multi-Core CPU berücksichtigt. Das binaurale Modell wurde dafür auf zwei Bandbereiche, den Bereich unter 1.5 kHz, in dem die ITD (interaurale Zeitdifferenzen) dominant sind, und den Bereich über 1.5 kHz, in dem die ILD (interaurale Laufzeitdifferenzen) dominant sind, reduziert.

Parallelisierte Algorithmen – GPU-Kernel

Die Umsetzung der Filterbank eröffnet zwei Möglichkeiten zur Parallelisierung: Zum einen die Unabhängigkeit der einzelnen Bandpassfilter und zum anderen die der einzelnen Audiokanäle. Umgesetzt wurde eine IIR-Filterbank nach Abbildung 1, sowie eine FIR-Filterbank nach Abbildung 2. Aufgrund der rekursiven Struktur lassen sich die einzelnen Biquad-Filterstufen nicht weiter zerlegen. Pro auditorischem Frequenzband werden 4 Biquadfilter-Stufen benötigt. Die FIR-Filterbank erlaubt hingegen eine wesentlich ausgeprägtere Parallelisierung, da die einzelnen Faltungsergebnisse unabhängig voneinander berechnet werden können. Allerdings werden hier 4096 Filterkoeffizienten benötigt.

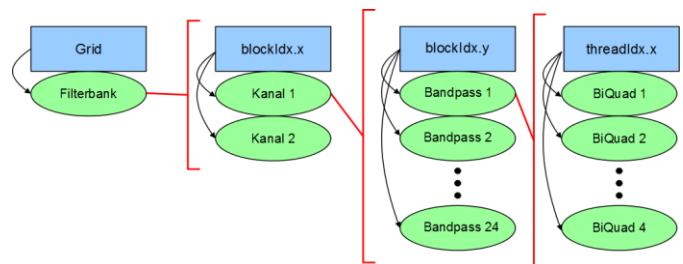


Abb. 1: Zerlegung einer IIR-Biquad-Filterbank für mehrere Kanäle zur Implementierung als CUDA-Kernel.

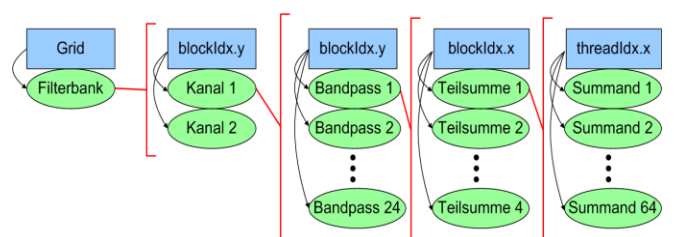


Abb. 2: Zerlegung einer FIR-Filterbank im Zeitbereich für mehrere Kanäle zur Implementierung als CUDA-Kernel.

Die erzielte Performance ist in Tabelle 1 aufgelistet.

Berechnungs-Dauer in ms	GTX 260+ (CUDA)	P4 3.6 GHz (C-Code)	P4 3.6 GHz (IPP-Code)
IIR 2 Kanäle	1.3	8.0	0.6
FIR 2 Kanäle	8.5	>> 100	13.0
IIR 8 Kanäle	4.0	32.1	2.2
FIR 8 Kanäle	28.0	>> 100	53.0

Tab. 1: Performance Betrachtungen CPU-GPU.

Zwar muss der Signalfluss innerhalb der Grafikkarte durch die CPU-Anwendung gesteuert und beim Verschieben der Daten mit dieser synchronisiert werden, die Ausführung der Kernel-Funktion erfolgt jedoch unabhängig und ermöglicht so eine parallele Verwendung der CPU. Abb. 3 zeigt den Prozessablauf innerhalb der VST-Blocksignalverarbeitung.

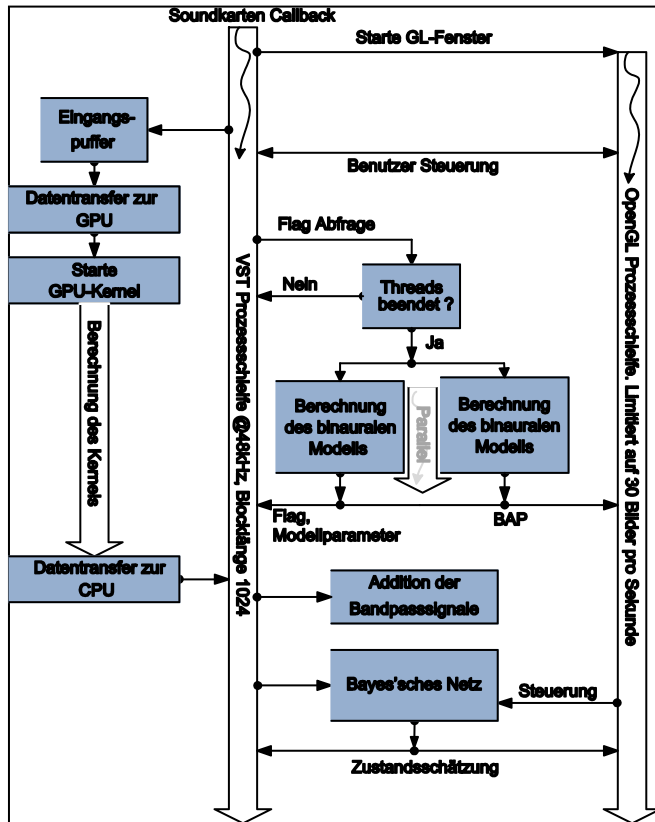


Abb. 3: Ablaufdiagramm der Gesamt-Anwendung. Den Kern bildet die Ausführung des VST-Host Threads, getrieben durch die Blockverarbeitung des Soundkartenpuffers.

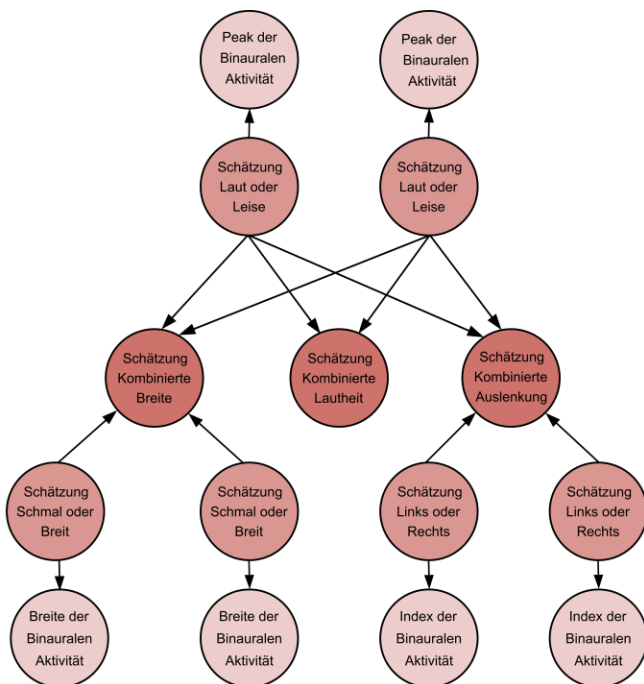


Abb. 4: Bayes'sches Netz zur Interpretation der binauralen Modellberechnung zur Abbildung auf Begriffe der menschlichen auditiven Wahrnehmung

Abbildung 4 zeigt das entwickelte Bays'sche Netz zur Interpretation der Modellparameter. Hierbei werden - jeweils für beide Bandbereiche getrennt - die visualisierten Modellparameter auf Begriffe der Wahrnehmung abgebildet und anschließend zusammengeführt. Zum Training des Systems durch individuelle Nutzer wird ein Audiosignal über Kopfhörer dargeboten. Dank der Echtzeitfähigkeit können die Nutzer eine einfache Anpassung des Systems an die eigene Wahrnehmung vornehmen. Hierfür wurde ein grafisches Interface erstellt, das die Ergebnisse der Lokalisations- und Lautheitsschätzung durch das Modell visuell darstellt und gleichzeitig durch die Einstellung durch Schieberegler (Slider) eine Möglichkeit bietet, die Schätzung zu verändern und direkt an die Wahrnehmung anzupassen. Diese Anpassung wird sofort sichtbar, sodass ein unverzögerter Abgleich zwischen der subjektiven auditorischen Wahrnehmung sowie der visuellen Darstellung dieser Wahrnehmung durch das System erfolgen kann.

Zusammenfassung und Ausblick

In dieser Arbeit wird eine echtzeitfähige Umsetzung eines binauralen Lokalisationsmodells zur instrumentellen Schätzung von Lokalisation und Lautheit von Hörereignissen vorgestellt. Durch ein neuentwickeltes grafisches Interface werden die Modelldaten aufbereitet dargestellt und Nutzern die Möglichkeit einer direkten Anpassung der Schätzung durch das Modell an die individuelle Wahrnehmung geboten. Die Modellinterpretation sowie das Lernen aus den Nutzereingaben erfolgt durch ein Bayes'sches Netz. Durch die Auslagerung von Signalverarbeitungsalgorithmen auf die GPU wurde es möglich, Modellberechnung und -visualisierung echtzeitfähig zu machen. Mit entsprechenden Testszenarien können auf diese Weise relativ einfach Erkenntnisse zur auditorischen Wahrnehmung des Menschen gewonnen werden, und es kann damit zu Weiterentwicklungen entsprechender Modelle beigetragen werden.

Zukünftigen Arbeiten können von der immer größer werdenden Leistungsfähigkeit von Grafikkarten profitieren. So ist die Entwicklung einer 24 Frequenzbänder umfassenden Modellumsetzung auf der Grafikkarte - unter Beibehaltung der Echtzeitfähigkeit - möglich.

Literatur

- [1] NVIDIA: CUDA C-Programmierungsumgebung (2009), <http://developer.download.nvidia.com>
- [2] Jeffress, L.A. (1948): "A place theory of sound localization", J. Comparative and Physiological Psychology, Vol. 41, 35-39
- [3] Lindemann, W. (1986): "Extension of a binaural cross-correlation model by contralateral inhibition. I. Simulation of lateralization for stationary signals", J. Acoust. Soc. Am., Vol. 80, 1608-1622
- [4] Hess, Wolfgang (2006): *Time-variant binaural-activity characteristics as indicator of auditory spatial attributes*, Dissertation, Ruhr-Universität Bochum
- [5] Schwyrz, M (2010): *Objekterkennung in der Lokalisationsebene mittels eines probabilistischen Modells*, Diplomarbeit, Uni Karlsruhe
- [6] Patterson, R. D., M. H. Allerhand, and C. Giguère (1995): "Time-domain modeling of peripheral auditory processing: A modular architecture and a software platform", J. Acoust. Soc. Am., Vol. 98 (4), 1890-1894
- [7] NorsysSoftwareCorp.: Netica. <http://www.norsys.com/>