Entwurf und Implementierung eines Eingabe-Frameworks für hybride interaktive Oberflächen

Zwischenvortrag Diplomarbeit

Anton Zeitler
LFE Medieninformatik

Aufgabensteller: Prof. Dr. Heinrich Hußmann
Betreuer: Raphael Wimmer
Datum: 30.6.2009

Ludwig
Maximilians-
Universität
München
LMU

Anforderungen Interaktionsschnittstellen

- Finger/Hand
  - Berührung (z.B. FTIR, kapazitiv)
  - Gestik
- Objekte (z.B. Arm)
- Marker
  - Fiducials
  - Formen bzw. Umrisse
- Geräte (z.B. Handy, Notebook)
  - Steuerung
  - Datenübertragung
- Identifikation (z.B. via Bluetooth, RFID)
- Annäherungssensoren
- Scannen von Dokumenten

[3]

Anforderungen
Technische Seite

Ludwig Maximilians Universität München

- modularer, leicht erweiterbarer Aufbau
- möglichst niedrige Latenz (< 30 ms)
- multiple Arten von Eingabegeräten und Eingabedaten
- Synchronisierung von Eingabe- (und Ausgabe-) Hardware
- Konfiguration und Steuerung der Eingabegeräte (Rückkanal)
- Anpassungs- bzw. Konfigurationsmöglichkeit jedes Arbeitsschritts
- Zugriff auf die Roh- bzw. Ausgabedaten jedes Arbeitsschritts
- Parallelisierung von Arbeitsschritten
- Plattformunabhängigkeit
- Schnittstelle für Skriptsprache

Besondere Herausforderungen

Ludwig Maximilians Universität München

- Kalibrierung gewölbter Flächen
- Entzerren und Zusammenfügen von Kamerabildern (Stitching)
- hohe Flexibilität bezüglich verschiedener Interaktionsformen und Eingabegeräte
- Synchronisierung von modulierten LEDs, Kameras, Projektion und evtl. weiteren Geräten

Aufgabenstellung

Ludwig Maximilians Universität München

- Entwurf der Architektur eines geeigneten Frameworks
- Spezifikation von Schnittstellen, Datenformaten
- Spezifikation möglicher Ausprägungen von Komponenten/Modulen
- funktionsfähige Referenzimplementierung

Softwarearchitektur
Layers (Schichten)

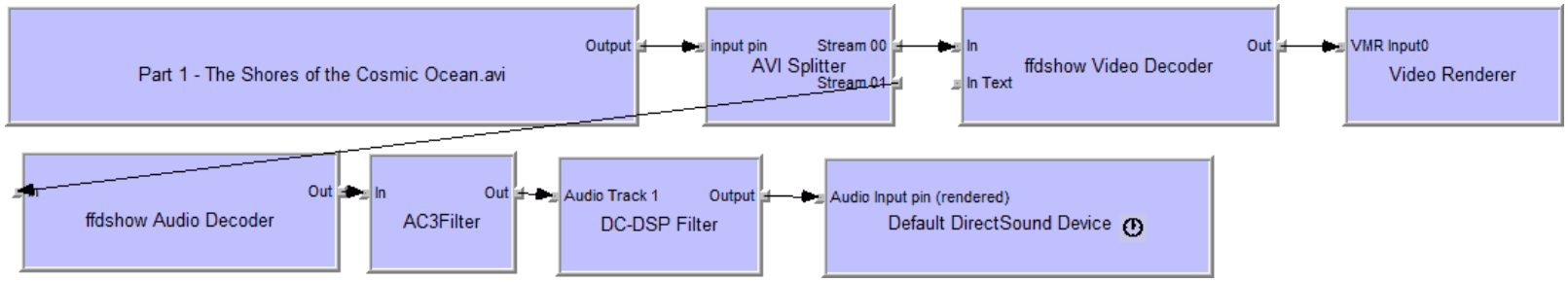
Ludwig Maximilians Universität München

- Aufteilung eines Problems in Abstraktionsschichten
- jede Schicht erfüllt spezifische Aufgaben, von unten nach oben stärker abstrahiert
- Beispiel: ISO/OSI-Modell
- Vorteile:
  - Reduzierung bzw. Aufteilung der Komplexität des Problems
  - Austauschbarkeit und Wiederverwendbarkeit von Schichten
  - Schnittstellen sind standardisierbar
  - auf jede Schicht kann eine andere Anwendung aufsetzen
- Nachteile:
  - zusätzlicher Kommunikationsaufwand zwischen Schichten
  - nach Festlegung der Schichten sind schichtübergreifende Anpassungen aufwendig
  - oft individuelle Schnittstellen zwischen verschiedenen Schichten

Softwarearchitektur
Pipes and Filters

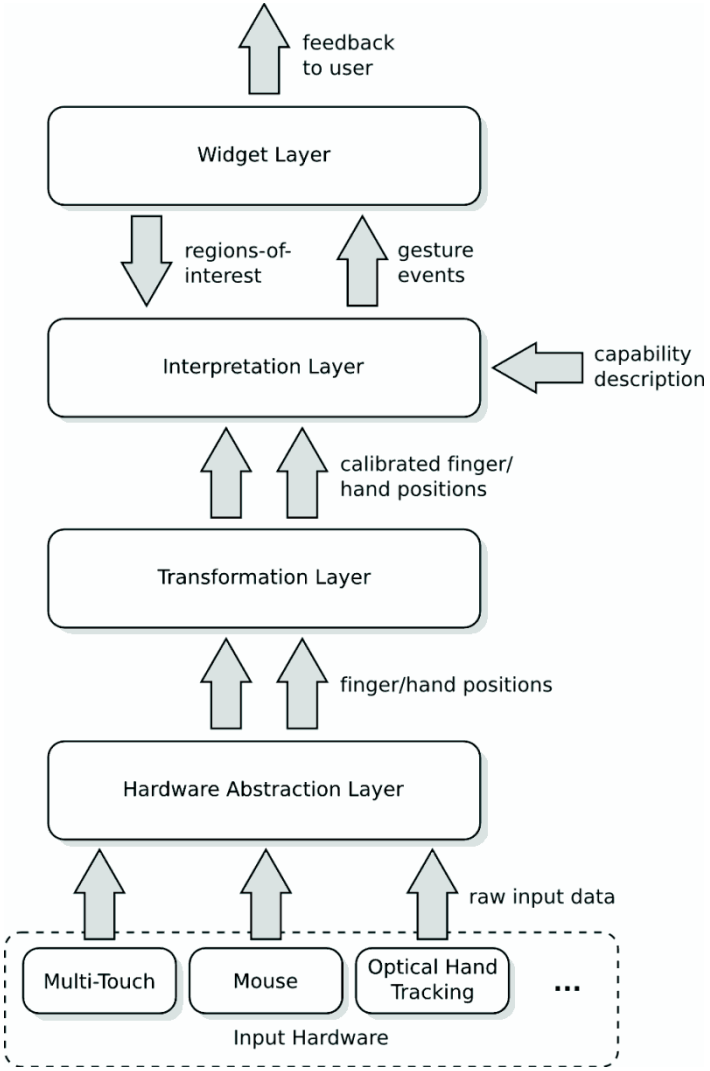
Ludwig Maximilians Universität München

- Verarbeitung von Datenströmen in einer Operationskette (Pipeline)
- ein Filter führt eine spezifische Operation auf den Daten aus
- Kanäle (Pipes) transportieren einheitliches Datenformat zwischen Filtern
- Beispiel: DirectShow
- Vorteile:
  - einheitliche Schnittstelle und einheitliches Datenformat
  - einfache Konfiguration verschiedener Pipelines mit beliebigen Operationsfolgen
  - Aufteilung von Datenströmen auf mehrere Filter möglich (Parallelisierung)
- Nachteile:
  - Synchronisierung zwischen Filtern notwendig
  - möglicherweise Datenkonvertierung für Pipes notwendig
  - Fehlerbehandlung kann aufwendig sein



Bestehende Ansätze libtisch

- Framework für beliebige multi-touch-Oberflächen
- Architektur: Layers (C++)
- UDP-Schnittstelle zwischen Schichten
- Vorteile:
  - kompakt und flexibel
  - leicht austauschbare Schichten
  - plattformunabhängig
- Nachteile:
  - nur Basisfunktionalität, kein Konzept für komplexe Anforderungen
- Autor: Florian Echtler (TUM)

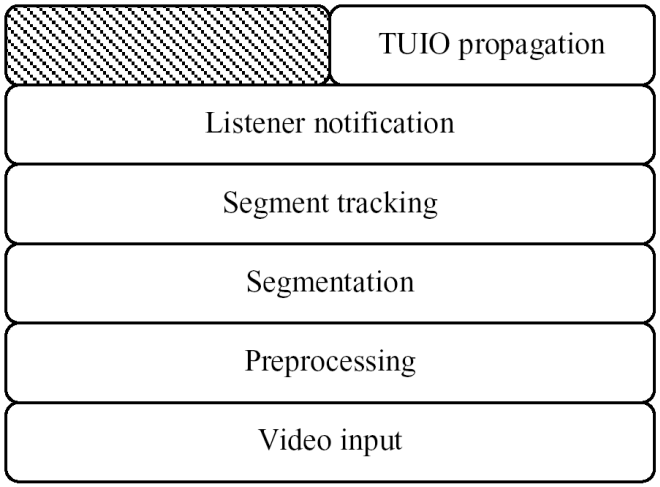


[4]

Bestehende Ansätze touchlib

Ludwig Maximilians Universität München

- Bibliothek für IR-basierte multi-touch-Oberflächen
- Behandelt überwiegend die Eingabe und leistet Vorverarbeitung
- Architektur: Layers/Pipes and Filters (C++)
- Vorteile:
  - bewährt und stabil
- Nachteile:
  - spezialisiert auf Blob-Erkennung, Finger und Hände
  - hohe Latenz (~65 ms bzw. ~33 ms)
  - plattformabhängig (Windows)
- Autor: NUI Group/White Noise Audio



[5]

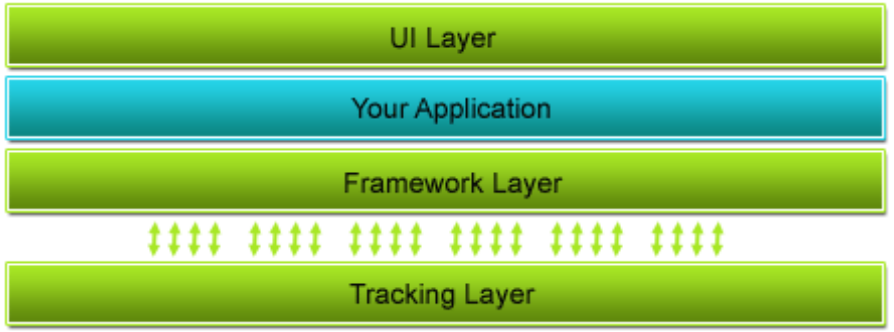
Bestehende Ansätze tBeta/CCV

Ludwig Maximilians Universität München

- neu entwickelte Bibliothek für IR-basierte multi-touch-Oberflächen
- ebenfalls Schwerpunkt auf Eingabe und Vorverarbeitung
- Architektur: Layers (C++)
- Vorteile:
  - flexible Kalibrierungsalgorithmen
  - Unterstützung für viele Kameramodelle
  - nutzt GPU für rechenintensive Aufgaben
  - plattformunabhängig
- Nachteile:
  - wie Touchlib Spezialisierung auf Finger und Hände
- Autor: NUI Group

Bestehende Ansätze
Xenakis

- Musikinstrument mit berührungsempfindlicher Oberfläche
- beinhaltet offenes Multi-touch-Framework TWING
- Architektur: Layers (C#)
- Vorteile:
  - gut strukturierter Ansatz für Multi-touch-Events
- Nachteile:
  - Eingabe eher peripher behandelt
  - plattformabhängig (Windows)
- Autor: Arbeitsgruppe der Universität Augsburg



[6]

Weitere bestehende Ansätze

- libavg (C++)
  - Schnittstelle zu Kameras und integrierter Tracker
  - überwiegend auf schnelle Grafikausgabe spezialisiert
- reacTIVision (C++)
  - spezialisiert auf fiducial markers
- Bespoke (Windows, .NET)
- BBTouch/OpenTouch (Mac, C++)
- Touché (Mac, C++)
- TouchKit (C++)
- OpenFTIR (kein Code verfügbar, weiterentwickelt als EquisFTIR)

Innovation

- multiple Arten von Eingabegeräten, nicht beschränkt auf Kameras
- multiple Arten von Eingabedaten, nicht beschränkt auf Bilder bzw. Blobs oder Marker
- Rückkanal zur Kommunikation mit der Eingabehardware
- Entzerren und Zusammenfügen von Kamerabildern (Stitching)
- Zugriff auf die Roh- bzw. Ausgabedaten jedes Arbeitsschritts
- Parallelisierung von Arbeitsschritten

Kommunikationsprotokolle

- Datenaustausch zwischen und innerhalb von Anwendungen
- textbasierte Protokolle
- OSC (OpenSound Control)
  - für Audio-Synthesizer entworfen, aber allgemein einsetzbar
  - sehr universell und daher recht komplex
  - Packets, Messages, Address Patterns, Type Tag Strings, Arguments, Bundles, ...
- TUIO
  - basiert auf OSC, aber vereinfachte Teilmenge
  - ausgelegt für multi-touch-Oberflächen
  - vordefinierte Profile mit zulässigen Befehlen je nach Anwendung
- libtisch
  - noch einfacher als TUIO
  - eigene Syntaxdefinition
  - Beispiel

    region 139869984 0 4 262.5 142.5 0 297.5 142.5 0 297.5 177.5 0 262.5 177.5 0 1 move 1 1 Motion 0 0 0 0 2 0 0 0 10000 0 0

Aktueller Stand

Ludwig Maximilians Universität München

- Anforderungen sind definiert
  - soll mit libtisch integriert werden (Framework ersetzt untere Schichten)
- Beginn des Architekturentwurfs
- Recherche nach wiederverwendbaren Elementen
- Entwurf und Implementierung in den nächsten 2-3 Monaten

Zielsetzung

- Spezifikation und Dokumentation der neuen Software-Architektur
- Kombination und Erweiterung bestehender Ansätze und Lösungen
  - mächtiger und flexibler
- funktionsfähige Referenzimplementierung

Quellenverzeichnis

- [1] https://torfkopp.medien.ifi.lmu.de/
- [2] Johannes Schöning, Peter Brandl, Florian Daiber, Florian Echtler, Otmar Hilliges, Jonathan Hook, Markus Löchtefeld, Nima Motamedi, Laurence Muller, Patrick Olivier, Tim Roth, Ulrich von Zadow
  Multi-Touch Surfaces: A Technical Guide
  Technical Report TUM-I0833, Technical Reports of the Technical University of Munich, October 2008
- [3] http://reactivision.sourceforge.net/
- [4] F. Echtler, G. Klinker
  A Multitouch Software Architecture
  NordiCHI 2008: Using Bridges, 18-22 October, Lund, Sweden
- [5] Michiel Hakvoort
  A Unifying Input Framework for Multi-Touch Tables
  10thTwente Student Conference on IT, Enschede, January 23rd, 2009
- [6] http://xenakis.origo.ethz.ch/wiki/documentation