

Einführung in die Medieninformatik

Für EMI im 1. Semester beim Bart

Allgemeines

Aufgabenstellungen

Berechenbarkeit (Lösen mathematischer Probleme), Unberechenbarkeit, Churchs These (Gleichwertigkeit aller Maschinenkonzepte), Chomsky-Hierarchie (Zusammenhang Automaten & Sprachgrammatik), Programmkomplexität, Programmverifikation

Medienkompetenz

Medialitätsbewusstsein + Medienwissen, Medienspezifisches Rezeptionsmuster, Medienbezogene Genussfähigkeit (Bedürfnis nach Identifikation und Unterhaltung), Medienbezogene Kritikfähigkeit.
Medien: Mittel zur Speicherung von Information (Physisch oder Format, Text, Video, Audio,)

Audio

Diskretisierung

- analoge Audiosignale verlaufen periodisch mit Frequenz (= Tonhöhe, i.d.R. zwischen 20 und 20 kHz), Amplitude (= Lautstärke)
- Digitalisierung erfolgt über Diskretisierung und Quantisierung:
 1. **Diskretisierung** ist Einteilung der kontinuierlichen Zeitachse in diskrete Punkte an denen abgetastet wird (an denen also die Amplitude des Signals gespeichert wird) (wird angegeben als **Abtastrate** in Samples pro Sekunde)
 2. dabei sollte gelten $f_{\text{abtast}} > 2 \cdot f_{\text{max}}$ ("Abtasttheorem"), sonst Informationsverluste und Aliasing
 3. menschliche Hörschwelle liegt bei 20 kHz; **Achtung:** liegt z.B. Signal mit $f_{\text{max}} = 30$

kHz vor, so sollte dennoch Abtasttheorem eingehalten werden (oder Tiefpassfilter anwenden), da sonst Phantomtöne mit $f < 30$ kHz entstehen (\Rightarrow für uns hörbar)

4. **Quantisierung** ist Speicherung der Amplitudenwerte: dafür steht nur eine begrenzte Bitzahl zur Verfügung, wie präzise man also Amplitudenwerte speichern kann hängt ab von Anzahl verfügbarer bits pro Sample (**Auflösung**) und min/max Frequenz
5. werden min/max Frequenzen für Digitalisierung falsch gewählt, so kommt es zu Clipping
6. da begrenzt viele bit zur Verfügung stehen müssen Amplitudenwerte gerundet werden \Rightarrow Quantisierungsrauschen

PCM

- PCM = Pulse Code Modulation, ein verlustfreies Digitalisierungsverfahren, am einfachsten ist **lineare PCM**
 1. **analoge Filterstufe:** Tiefpass mit bestimmter Grenzfrequenz (es werden also alle höheren Töne rausgefiltert) $\Rightarrow f_{\text{abtast}}$ muss für Einhaltung des Abtasttheorems nicht ganz so hoch gewählt werden
 2. **zeitliche Abtastung** (=Diskretisierung, siehe Oben)
 3. **Quantisierung** (siehe Oben)
 4. **Codierung** (es wird alles binär gespeichert)
- weitere Variante: **Differential PCM** d.h. statt den Amplituden an sich wird hier nur Differenz benachbarter Amplituden gespeichert (\Rightarrow max. Kompressionsfaktor 1:2)
- weitere Variante: **Adaptive DPCM** d.h. DPCM mit Anpassung der Quantisierungsauflösung (je lauter das Signal wird, desto gröber wird quantisiert; siehe auch Quantisierung)

RIFF WAVE ("WAV")

- Containerformat für PCM Daten, diese liegen in Hex vor
- besteht i.d.R. aus folgenden Chunks: RIFF (12 byte), fmt(8 byte+16 byte), data (8 byte + größe der Rohdaten)

Kompression

- Informationen können **redundant** sein (keine neuen Informationen), Informationen können **irrelevant** sein (nicht wahrnehmbar) \Rightarrow wichtig ist was nicht irrelevant und nicht redundant ist
- redundant sind z.B. periodische Wiederholungen; irrelevant sind z.B. Frequenzen > 20 kHz und leise Töne, die durch laute Töne maskiert werden ("**Mithörschwelle**"); laute Signale maskieren sogar kurzzeitig leise Töne die vor/nach dem lauten Signal zu hören wären
- **Entropiecodierung** eliminiert Redundanz (verlustfrei), **Irrelevanz-Codierung/Reduktion** eliminiert Irrelevantes (verlustbehaftet) (z.B. durch löschen von Teilsignalen, Anpassung der Quantisierung), **Dekorrelation** ist Umformung ohne Datenmenge zu ändern
- **MP3** = MPEG Layer 3, beansprucht 20-230 kBit/s (\Rightarrow deutlich kleiner als PCM Rohdaten); Signal wird in Frequenzbänder zerlegt, auf diese werden jeweils angewendet u.A. Entropiecodierung, Irrelevanz-Codierung, nichtlineare Quantisierung, Huffman-Codierung

MIDI

- mit dem **MIDI** Format werden statt Tönen nur Daten zu Instrument, Tonhöhe, Dauer des Anschlags etc. übertragen, woraus ein (evtl. anders klingender!) Ton rekonstruiert wird
- Verwendung z.B. in Musiksoftware und Synthesizern, das Format wird auch für andere Sachen als Audio verwendet

- Problem: evtl. Verzögerungen da serielle Übertragung; heute ist Datenübertragung per USB möglich

Audioverarbeitung

- veränderung der **Hüllkurve** = Veränderung des Lautstärkeverlaufs, so auch Ein-/Ausblenden (fading)
- **Hochpassfilter** multipliziert hohe Frequenzen mit Faktor 1, tiefe mit Faktor 0, dazwischen Faktor zwischen 0 und 1 (\Rightarrow tiefe Frequenzen werden rausgefiltert); umgekehrt für **Tiefpassfilter**
- **Bandpassfilter** analog zu Hoch/Tiefpass aber lässt nur bestimmten Frequenzbereich in der Mitte des Frequenzspektrums durch und filtert höhere und tiefere Frequenzen raus, **Bandsperre** umgekehrt: filtert mittlere Frequenzen raus
- **Dynamikkompression**: laute Stellen werden abgesenkt, leise angehoben
- Pegeländerungen können zu Über/ Untersteuerung führen (höchste Amplituden liegen weit über/ unter Bereich der quantisiert wird (siehe Quantisierung)); führt zu Clipping (Störsignalen) bei Übersteuerung; führt zu Rauschen bei Untersteuerung
- **Resampling**: Signal wird mit anderer Samplefrequenz abgespielt, schneller abgespieltes Signal klingt dann z.B. höher (wie Chipmunks), dieser Effekt kann vermieden werden durch **Time Stretching** (kurzen Ausschnitt wiederholt abspielen)

Klangsynthese

- Anschlagsverhalten wird per Hüllkurve in 4 Schritten nachgebildet: Attack (wie schnell wird der Ton direkt nach dem Anschlag lauter), Decay (wie schnell erreicht er gleichbleibende Lautstärke), Sustain (Lautstärke beim Aushalten), Release (Loslassen)
- **Wave-Table Verfahren**: es wird zum Beispiel Datei für den Laut "E" oder einen Klavierton

gespeichert und bei Bedarf abgespielt, gleiches dann für alle anderen Laute/Töne

- **Additive Synthese**: jede Funktion kann als unendliche Summe von cos/sin dargestellt werden \Rightarrow durch endliches addieren von solchen Funktion kann gewünschtes Signal angenähert werden
- **subtraktive Synthese** nimmt durch Oszillator erzeugtes Signal und verändert dieses durch Filter
- **FM-Synthese** (z.B. bei alten Keyboards) modulieren von Trägerfrequenz durch Modulationsfrequenz, resultierender Klang entspricht keinen akustischen Instrumenten
- Synthesizer erstellen durch GUI oder Skript möglich

Video

Historie

- Zelluloidfilm ab 1888 (Photoemulsion auf Zelluloid): Verschiedene Filmformate, heute **35mm, 24 Bilder / Sekunde**, 1km film ergibt ca. 36,5 min
- Kinemacolor (1906): 32 Bilder / sek, 16 rot, 16 grün mit rotierendem Filter
- Technicolor (1917): Aufspaltung mit Prisma auf 3 Filme
- Agfacolor (1932): als Farbfilter wirkende Körner auf dem Papier
- Edison Phonograph (1877): Nadel schreibt Ton auf Walze mit
- Lichttonverfahren (1922): Lichtdurchlässigkeit proportional zur Amplitude des Tonsignals
- Später: Kombination mit Magnetton (vgl. Tonband, Cassette)
- 4-Kanalton wird als Matrix in die Lichtkanäle moduliert, Dolby 5.1 Surround auf CD oder digital in Filmperforation, 8-Kanal-SDDS komprimiert digital auf Filmaußenrand

- Interaktive Filme: Mehrere Perspektiven werden aufgenommen und durch den Benutzer gewechselt
- Rendering: Berechnung jedes Bildes durch Rechner (Animationsfilm)
- Kamera (Edison: Kinematograph 1891). Unbelichteter Film wird von Spule in Kamera gerollt, ein Bild belichtet, Blende verdeckt das Bild, Film wird auf zweite Spule gerollt und anschließend entwickelt
- Projektoren für 35mm: Kinetoscope (1892 Edison) mit einem Betrachter, ab 1895 auf Leinwand
- Projektor in modernem Kino: Film auf 3 Tellern gewickelt, rein ins Projektor, raus auf freien Teller
- Digitalprojektor: digitales Bild wird gelesen, Lampe wird extra gekühlt

Formate

- Flüssige Bewegung ab ca. 20 FPS, Film: **24 Hz**
- Flimmern: Bei Dunkelpausen zwischen Bildern, dadurch unangenehmes Flackern
- Flimmerverschmelzungsfrequenz ca. 50 Hz, steigt mit Helligkeit
- Bilder zweifach projiziert, Verdopplung der Dunkelpausen, Flimmerfrequenz 48 Hz
- Computermonitore: hohe Helligkeit, geringer Abstand, hohe Aufmerksamkeit. Frequenz muss mindestens 72 Hz sein
- Keine Dunkelpausen bei TFT-Displays, dadurch kein Flimmern
- Helligkeitsempfindung unterstützen, abhängig vom Zustand des Auges (Auge kann sich an Leuchtdichteunterschiede anpassen, von mondloser Nacht zum sonnenbeschienenen Schneefeld in 30 min). Das angepasste Auge kann ca. 200 Helligkeitswerte unterscheiden
- Fernsehen: Streulicht schränkt Kontrastwahrnehmung ein, Umfeldleuchtdichte sollte ca. 10 % der Spitzenleuchtdichte des Bildschirms betragen

- **Aliasing:** Treppenstufen und Geisterfrequenzen bei Abweichung von abgetastetem und abtastenden Raster
- **Zeitliches Aliasing** (Wagenradeneffekt): Wagenrad scheint sich rückwärts zu drehen bei zu geringer Abtastfrequenz, Stillstand falls genau um 90 Grad verdreht. Vermutlich durch neuronale Adaption verursacht
- **Videosignal (s/w analog): Zeilenweise Abtastung der Helligkeitswerte, serielle Übertragung der Bildpunkte, synchrone Darstellung auf dem Monitor**
- **Progressiv:** Ganzes Bild für Zeile für Zeile übertragen
- **Interlaced:** 2 Halbbilder werden nacheinander übertragen und ergeben zusammen ein Vollbild. Das erste Bild beschreibt nur jede zweite Zeile etc. Ursprünglich zur Reduktion von Bildflimmern entwickelt (50 Hz bei Röhrenmonitoren)
- 100Hz-Röhrenfernseher und LCDs sowie Plasmafernseher zeichnen progressiv, dadurch **Umrechnung** des TV-Signals erforderlich
- **Weave: Gleichzeitige Darstellung der Halbbilder.** Funktioniert bei TV nicht, da die Halbbilder zeitlich versetzt aufgenommen werden, führt zu Kammeneffekt
- **Unschärfe: Weave mit Weichzeichnen**
- **Motion Compensation: Weave**, wobei ungerade Zeilen an gerade Zeilen **angepasst** werden (**aktuell**)
- **Bobbing:** Jedes Halbbild wird durch **interpolation** zu einem Vollbild erweitert, um 50 Hz zu erreichen. Dadurch *wackelt das Bild*, da erste und letzte Zeile jeweils ohne Nachbar sind
- **Blending: Bobbing**, wobei **beide Halbbilder** zum Vollbild ergänzt werden und dann der **Mittelwert** zwischen beiden Bildern genommen wird
- Analoge Formate

- Analoges Fernsehen ohne Zwischenspeicherung arbeitet mit Zeilen (progressive) und Halbzeile (interlaced)
- **BAS:** Helligkeit auf 75 % Signalamplitude skaliert, Austastung durch Zeilenrücklauf (100 % Amplitude für 5 Mikrosekunden), Bildsynchronisation durch volle Amplitude für 10 Mikrosekunden
- **FBAS:** Farbsignale werden überlagert
- **Komponentenvideo:** Getrennte Übermittlung von drei Farbkanälen (RGB, YUV oder YIQ)
- **Y/C-seperiertes Video:** Trennung von Helligkeit und Chrominanzsignalen
- Europäisches Standardsignal **PAL** (Phase Alternate Line System): **interlaced FBAS** in YUV, Auflösung 833x625 Pixel, 720x576 sichtbar, 50 fields = **25 fps**; Bandbreite Y: 5,5Mhz, U/V: 1,8 Mhz
- **NTSC** (National Television Systems Committee): US Standard, **Interlaced FBAS** in YIQ, 700x525 Pixels, **29,97 fps**, ohne Kontrolldaten 485 Zeilen, wovon 480 nutzbar sind
- **Secam** (Sequential Couleur avec Memoire): wie PAL mit Bandbreiten 6/2/2 (Y/U/V)
- NTSC in Amerika, PAL in Europa, Australien, Asien, Secam in Frankreich / Russland
- Digitale Formate
 - Digitale Videos unterstützen weitere Seitenverhältnisse und korrigieren nichtquadratische Pixel
 - Digitales Fernsehen wird über Funktechniken (Antenne, Satellit) und Kabel verbreitet
 - DVB: Digital Video Broadcast, MPEG2-basiert, Senden als Stream, MPEG Digital Storage Command Control (DSM-CC)

für Interaktivität. DVB-C (Cable), DVB-T (Terrestrisch über Antenne), DVB-S (per Satellit), Container nach **Objektkarussell** transportiert **Audio, Video und Daten**. DVB-H für Verteilung über Mobilfunk nicht mehr verbreitet

- **Bildebenen:** schwarzer Hintergrund, Video, Untertitel / Gebärden, HbbTv, Geräteoberfläche (Uhrzeit, Lautstärke). Anwendungen werden über HTTP vom Sender ermittelt.
- **Seitenverhältnis:** TV 4:3 (1.33:1), HDTV 16:9 (1.78:1), 21:9 (2.33:1); Je nach analogem Format werden die einzelnen (digitalen) Pixel entsprechend nichtquadratisch gespeichert *Digitale Videos unterstützen weitere Seitenverhältnisse und korrigieren nichtquadratische Pixel*
- **Digitales Video:** Bilder werden als Sequenz gesendet. Ein Videostream besteht aus Bildern und Teilbildern, sowie Ton. Bilder bestehen aus Pixeln. Pixel haben eine Farbe.

Komprimierung

- Diskretisierung (Abtastung)
 - *Diskretisierung muß Farbe in historischem Schwarzweiss-Verfahren integrieren*
 - **4:1:1** Farbsignal nur bei jedem 4. Pixel, dafür aber in beiden Halbbildern horizontal verschoben abgetastet.
 - **4:2:0** Farbsignal nur bei jedem zweiten Halbbild und nur in halber Auflösung abgetastet
 - **4:2:2** Jedes Pixel wird nach Luminanz (Y) und jedes 2. nach den Rot- und Blaudifferenzen (Cr und Cb) abgetastet. (für professionelles digitales Video)
 - **4:4:4** Jedes Pixel, sowohl in Luminanz wie in den Blau- und Rotdifferenzen wird abgetastet; (Hochqualitatives digitales Format)

- **CODEC:** bezeichnet ein Algorithmenpaar: **Compressor** komprimiert die Videodaten; **Decompressor** stellt Ursprung wieder her; meist sind dabei Informationen verloren gegangen
- Komprimierungstechnik:
 - **Redundanz Reduktion:** Verzicht auf mehrfach vorhandene Information(z.B. räumlich, zeitlich); (verlustfrei)
 - **Irrelevanz Reduktion:** Verzicht auf kaum erkennbare Signalanteile (verlustbehaftet)
- Bei Video besonders sinnvoll, Differenz zwischen zwei Bildern zu berechnen (vgl. **DPCM**)
- **H.261:** Video-Telefonie über ISDN (Bandbreite 64 kbit/s mal Kanal), weniger als 150ms Verzögerung durch Kompression / Dekompression, CIF / QCIF, YCrCb 4 : 2 : 0, Kompression auf 1,5Mbit / s (24x ISDN)
- **H.263:** Weglassen unwichtiger Teile und Verringerung der Bittiefe, halbe Datenrate
- **Intraframes** (JPEG codiert): 16x16 große Blöcke, Unterteilung in 8x8-Blöcke für Helligkeit und zwei Blöcke für Chromakanäle, DCT mit anschließender Huffman-Codierung
- **Predicted Frames:** Unterteilung in 16x16 Blöcke, für jeden Block Suche nach "Best Match" im vorherigen Frame, **Differenzcodierung**
- **MPEG** (Moving Pictures Experts Group): Standard für komprimiertes Speichern von Audio und Video. MPEG-1 (1992, nur progressiv für Videos), MPEG-2 (1993, TV, HDTV), MPEG-3 (HDTV, in MPEG-2 integriert), MPEG-4 (laufende Entwicklung), MPEG-7 (DescriptionInterface), MPEG-21 (Digital Multimedia Framework)
- **Sequence:** Alle Bilder vom s_{header} bis zum s_{end}
- **GOP:** Group of Pictures zwischen 2 GOP-Headern
- **Picture:** Field (Halbbild, d.h. jede zweite Zeile) od. Frame (Vollbild)

- **Macroblock:** 16 x 16 Pixels
- **Slice:** 16 Zeilen
- **Block:** 8 x 8 Elemente (Luminanz / Chrominanz / DCT-Koeff.)
- **Sample:** Einzelner Helligkeitswert eines Pixels

MPEG

- Bei MPEG-1 werden **B-Frames** eingeführt, die von vorne und hinten interpoliert werden (**Macroblockweise** aus nächstem und vorherigen vorhergesagt; **Differenz** mit DCT und Huffman-Kodierung), I-Frames JPEG-kodiert, ansonsten wie H.261
- MPEG Datenstrom
 - **Audio:** Gliederung in Frames (Folge von Abtastwerten), Audio Access Units (kleinste für sich voll dekodierbare Einheit) und Slots (1 - 4 Byte).
 - **Video** aus 6 Schichten: Sequence Layer, Group of Pictures layer (Abfolge der verschiedenen Bildtypen), Picture layer (kodierte Einzelbilder), Slice Layer (Ebenen, Komponenten) eines Bildes, Macroblock Layer (Makroblöcke), Block Layer (Einzelblöcke)
- MPEG-2 zu MPEG-1: Best-Match-Suche auch nach Fields, Makroblöcke Farb-Subsampling von 4 : 2 : 2, 4 : 4 : 4, die Frame-Größe kann bis zu 16383^2 Pixel betragen, nichtlineare Quantisierungstabelle
- Verschiedene Profile (High Level: FullHD, Main Level: HD), in höheren Profilen **räumliche Skalierbarkeit** (Datenstrom kann in verschiedenen Auflösungen gezeigt werden), **Zeitliche Skalierbarkeit** (Datenstrom adaptiv zu Veränderungen der Bandbreite)
- **MPEG-1** bis HD (CD-ROM, Digital Storage Media); (1,5 Mbps); 352x240, 720x480; Stereo CD

- **MPEG-2** DVD, Broadcast, HDTV, (4,6 Mbps); 176x144, 352x288, 1280x720; Surround Sound
- **MPEG-4** für TV, Virtual Games, Streaming [...]; (20 Kbps - 6 Mbps); 176x144, 352x288, 720x480, 1280x720; Sprache, Musik, CD
- MPEG-4: Kodierung von Szenen von audiovisuellen Objekten, Kombination natürlicher und synthetischer Medien-Objekte, Prinzip unabhängig von Bitrate (low bitrate bis lossless)

YUV

- Farbwahrnehmungsbezogenes Chroma-Subsampling $Y : C_R : C_B$, Y-Abtastung 3 Mhz, C_R und C_B -Abtastraten ev. geringer, Räumliche Position wird angedeutet
- AVI: Containerformat für beliebigen Codec
- DV-Standard bei Camcordern (DCT, D1, D2..., Digital Betacam), CCIR 601 o. ä.
- MJPEG: JPEG-basierte Kompression der Vollbilder, nicht standardisiert
- MPEG: Standard
- Bei Video besonders sinnvoll, Differenz zwischen zwei Bildern zu berechnen (vgl. DPCM)
- H.261: Video-Telefonie über ISDN (Bandbreite 64 kbit/s mal Kanal), weniger als 150ms Verzögerung durch Kompression / Dekompression, CIF / QCIF, YCrCb 4 : 2 : 0, Kompression auf 1,5Mbit / s (24x ISDN)
- H.263: Weglassen unwichtiger Teile und Verringerung der Bittiefe, halbe Datenrate
- Intraframes (JPEG codiert): 16x16 große Blöcke, Unterteilung in 8x8-Blöcke für Helligkeit und zwei Blöcke für Chromakanäle, DCT mit anschließender Huffman-Codierung
- Predicted Frames: Unterteilung in 16x16 Blöcke, für jeden Block Suche nach "Best Match" im vorherigen Frame, Differenzcodierung

- MPEG (Moving Pictures Experts Group): Standard für komprimiertes Speichern von Audio und Video. MPEG-1 (1992, nur progressiv für Videos), MPEG-2 (1993, TV, HDTV), MPEG-3 (HDTV, in MPEG-2 integriert), MPEG-4 (laufende Entwicklung), MPEG-7 (DescriptionInterface), MPEG-21 (Digital Multimedia Framework)
- Sequence: Alle Bilder vom *s_header* bis zum *s_end*
- GOP: Group of Pictures zwischen 2 GOP-Headern
- Macroblock: 16 x 16 Pixels
- Slice: 16 Zeilen
- Block: 8 x 8 Elemente (Luminanz / Chrominanz / DCT-Koeff.)
- Sample: Einzelner Helligkeitswert eines Pixels
- Bei MPEG-1 werden B-Frames eingeführt, die von vorne und hinten interpoliert werden (Macroblockweise mit DCT und Huffman-Kodierung), I-Frames JPEG-kodiert, ansonsten wie H.261
- Audio: Gliederung in Frames (Folge von Abtastwerten), Audio Access Units (kleinste für sich voll dekodierbare Einheit) und Slots (1 - 4 Byte).
- Video aus 6 Schichten: Sequence Layer, Group of Pictures layer (Abfolge der verschiedenen Bildtypen), Picture layer (kodierte Einzelbilder), Slice Layer (Ebenen, Komponenten) eines Bildes, Macroblock Layer (Makroblöcke), Block Layer (Einzelblöcke)
- MPEG-2 zu MPEG-1: Best-Match-Suche auch nach Fields, Makroblöcke Farb-Subsampling von 4 : 2 : 2, 4 : 4 : 4, die Frame-Größe kann bis zu 16383² Pixel betragen, nichtlineare Quantisierungstabelle
- Verschiedene Profile (High Level: FullHD, Main Level: HD), in höheren Profilen räumliche Skalierbarkeit (Datenstrom kann in verschiedenen Auflösungen gezeigt werden), Zeitliche Skalierbarkeit (Datenstrom adaptiv zu Veränderungen)

- MPEG-1 bis HD (1,5 Mbps), MPEG-2 für DVD oder Broadcast oder HDTV: 4,6 Mbps, MPEG-4 20 Kbps - 6 Mbps für TV, Virtual Games, Streaming [...]
- Neue Interaktivität in MPEG-4 gewünscht
- MPEG-4: Kodierung von Szenen von audiovisuellen Objekten, Kombination natürlicher und synthetischer Medien-Objekte, Prinzip unabhängig von Bitrate (low bitrate bis lossless)
- 3D-Szenengraph: Objektbaum
- X3D, VRML, OpenGL: Rendering-Pipeline (Tessellation, Transformation, Sichtbarkeit, Beleuchtung, Projektion, Clipping, Transformation in affine Koordinaten, Viewport mapping, Scan conversion, Verdeckung, Scissoring, Shading / Texturisierung, AntiAliasing)
- VRML: "SVG in 3D" ohne XML, z. B.:

```
#VRML V2.0 utf8
Shape {
  geometry Box{ size 2 2 2}
}
```

Multimodale Systeme

Multimodale Systeme nutzen mehr als einen Sinneskanal zur Wahrnehmung beim Menschen. Dadurch Verbesserung der Intuition und Erhöhung der Robustheit eines interaktiven Systems z. B. durch Einbringung von Lippenbewegungen, Kommunikation kann barrierefreier gestaltet werden.

- Pipeline zur Wahrnehmung: Stimulation, sensorischer Kurzzeitspeicher, Wahrnehmung, Entscheidungsfindung, Reaktion, ggf. Gedächtnisspeicherung. Aufmerksamkeit ab Wahrnehmung nötig.
- Beispielsweise Datenhandschuh, Joystick

Stereoskopische Ausgabe

- Shutter-Brillen, Head Mounted Display (Problem: Cyber-Sickness weil Bewegungen unnatürlich)

- Head Mounted Display: Kopfhörer, Positionssensoren liefern Daten über Lage und Blickrichtung
- Autostereoskopische Techniken: z. B. 3D-LCD-Display mit Linsensystem, führt Kopfbewegungen mit Head-Tracking-System nach (Brechwinkel bedingen Abstand), Objekte können vor und hinter dem Display liegen
- Autostereoskopische Displays sind noch nicht ausgereift: Meist nur ein oder zwei Betrachter, nur Raumillusion, feste Betrachtungsposition, "Blick durch Fenster" (nicht sehr realitätsnah), Objekte, die nur von einem Auge gesehen werden, müssen ausgeblendet werden, noch zu reaktionslahm
- Polarisierte Lichtquellen für 3D: Bild wird polarisiert statt durch Farbfilter überlagert, je ein Auge sieht horizontal oder vertikal polarisiertes Licht
- STB: Interaktiver Empfänger für Fernsehen und Internet. Bindeglied zwischen HiFi, TV, PC etc und Internetz. Standards für APIs, Netzprotokolle, Medienformate. Unterschiedliche Hardwarekonzepte, aber meist Tunermodul, Demodulator, Prozessor, MPEG2-Decodierer, Demultiplexer, Backchannel-Controller, Descrambler, Schnittstellen-Controller, Audio DAC, Entschlüsselungssystem (Conditional Access)

Processing

- Variablendeklaration: TYP NAME=WERT (Variablenamen immer klein schreiben)
- Felder: TYP[][] NAME = {WERT1, WERT2, ...}, {WERT_A, WERT_B, ...} (eindimensional oder mehrdimensional möglich); Zuweisung über NAME[1][0]=45; (Elemente von 0 beginnend nummeriert)

Ausgaben

- `print()`; Ausgabe auf der Konsole; `println()` mit anschließendem Zeilenumbruch
- Kommentare mit `//` oder mehrzeilig mit `/*...*/`
- Formen: `line(x1, y1, x2, y2)`; `triangle(x1, y1, x2, y2, x3, y3)`; `rect(x, y, breite, hoehe)`; `ellipse(xM, yM, durchmesserBreite, durchmesserHoehe)`;
- Zeichenmodi: `rectMode(CORNER)`: normal linke obere Ecke als Bezugspunkt; `rectMode(CENTER)`: Mittelpunkt als Bezugspunkt; `rectMode(CORNERS)` zwei EckPUNKTE angeben; (für `ellipseMode` ist `CENTER` standard, `CORNER[S]` arbeitet mit umgebenden Rechteck
- Farben: `fill(0-255)` für Graustufen; `fill(R,G,B)` in 8bit(0-255) erwartet; gilt für alle folgenden gezeichneten Formen; (weiterer Wert bei `fill()` Methode für Transparenz)
- Hintergrund: `size(x,y)` legt Breite, Höhe fest; `background(R,G,B)` Farbe des Hintergrundes festlegen (auch mit Graustufen mgl. s. o.); `background()` übermalt gesamten Zeichenbereich
- Text: `text(string, basisX, basisY)`; [basis=linke untere Ecke]; `textSize(0-∞)` Textgröße in pt; `textAlign()` `CENTER`, `LEFT`, `RIGHT` Ausrichtung des Textes
- String (Bsp. s): `s.length()` gibt Länge des Strings zurück; `s.equals(s2)` prüft Strings auf Inhaltsgleichheit (nicht = verwenden überprüft Speicheradressen); `s.substring(0,5)` gibt Teilstring (start inklusive, ende exklusive) zurück (start bei 0) im Bsp. die ersten 5 Zeichen von s

aktiver Modus

- benötigt `void setup(){}` und `void draw(){}` (Endlosschleife bis Programm abbricht) Methoden
- `framerate()` in 1/s für `draw` Methode im `setup` Teil festlegbar

- Mauseingaben: Variablen **mouseX** und **mouseY** beinhalten aktuelle Position der Maus *pmouseX*, *pmouseY* enthalten Position der Maus in vorheriger draw "Runde"
- Tastatur Eingaben: `void keyPressed(){}` methode beinhaltet Variable `key` mit gedrückter Taste (Unterscheidung über `Switch` Anweisung hilfreich) (ebenso `void mousePressed(){}`) (!!funktioniert nur im aktiven Modus (setup und draw))
- Zufall: `random(10)` gibt zufällige Zahl zw. 0 und 9

Audio

- `import processing.sound.*`; Importiert Soundbibliothek
- `SinOsc osc`; Deklaration einer Oszillator Variable; `osc = new SinOsc(this)`; Speicher von `osc` wird initialisiert; (meist in `setup` ausgeführt)
- `osc.play(freq, amp)`; startet Ausgabe eines Tons (`osc.stop` stoppt Ton)
- `osc.amp(amp)` verändert Amplitude[0,1]; `osc.pan(pan)` verändert links rechts[-1,1]; `osc.freq(freq)` verändert Frequenz[hörbarer Bereich];

Java

Java Programmierung

- Java ist objekt-orientiert (parallele Programmierung ist leicht möglich)
- Java ist dynamisch Speicherbedarf wächst und schrumpft automatisch
- Übersetzen des Quellcode in Bytecode:
 1. Eingabe von Quellcode (erzeugt Quellcode Datei)
 2. Übersetzen in Bytecode (erzeugt separate Bytecode Datei [Prozessor lesbare .class Datei]) (bei Fehler gehe zu 1)

3. bei Erfolg: ausführen des Bytecodes (bei Fehler gehe zu 1)

- Bytecode für alle OS gleich; Unterscheidung im jeweiligen Interpreter
- Hauptbestandteile:
 - Methode: ist eine Folge von Anweisungen; werden aufgerufen (können sich selbst aufrufen(rekursion)); werten Parameter aus; können vererbt werden(nur bei objektorientiert)
 - *main*-Methode: einzigartig in jedem Programm; legt Beginn des Programms fest; wird vom Interpreter als erstes im Bytecode gesucht
 - Klassen: sind Vorlagen für Objekte (mit *new* werden Objekte erzeugt); Objekte haben Attribute(Eigenschaften) und führen Methoden aus
 - Blöcke: fassen Folgen von Anweisungen zusammen; legen den Namensraum für Variablen fest; Verschachtelung mgl.; in geschweiften Klammern
 - Anweisungen
 - Kontrollstrukturen: `If`, `While`, `For(Foreach)`, `Do`, `Switch`, Kontrolltransfer-Anweisungen(`break`: beende Anweisungsfolge (beendet eine oder mehrere verschachtelte Schleifen), `continue`: beendet in einer Schleife sofort einen Durchlauf, `return`: beendet Anweisungsfolge einer Methode, (`throw`: später))
 - Modifiers: *public*, *static*
 - Package: enthält(Quellcode und Bytecode) einer oder mehrere Klassen, die sich einen gemeinsamen Geltungsbereich (Namespace) für Klassen teilen; Klassennamen müssen nur in einem Package eindeutig sein; über *import* einzelne Klassen importierbar
 - Kommentare: `//...`, `/*...*/`

- reservierte Worte: *class*, alle Modifiers,... dürfen nicht für Variablennamen o.Ä. verwendet werden

Android Apps

- Android ist linuxbasiert; Apps laufen auf optimierter Java VM für mobile Prozessoren; Apps kommunizieren miteinander (durch Intents))
- Apps bestehen aus Modulen für Interaktion des Nutzers mit Daten(Activities); beenden sich selbst über ständig prüfende "Abbruchschleife"
- **Entwicklungsprozess:** Setup (IDE wählen, SDK, virtuelle Geräte einrichten); Entwicklung (Quellcodes, Ressourcen, Manifest); Debugging(ADB, logs, Geräte per USB testen); Veröffentlichung(Release, Signatur, App Store)
- Quellcode \Rightarrow Bytecode: Android Package Datei (ausführbare Datei) enthält: Datei, (formatierte)Ressourcen, AndroidManifest.xml (beschreibt: SDK Version, packages, Activities, Icon, Stil)
- Daten \Rightarrow Medien: (von unten) [Bits und Byte] [Sensordaten, Prozessor] [Netzwerk, Protokolle] [Ressourcen, Medien, Codec] [GUI] [App] nur oberen 3 Komponenten in der App sichtbar

Smartphone

- Ausgabetechnik: Display, Lautsprecher, Vibrationsmotor
- Eingabetechnik: Touchscreen, Uhr(am Kernel), Akkumulator, Sensoren(Mikrofon, Kamera, Gyroskop, Beschleunigungssensor, Kompass, Thermometer, Näherungssensor, Lichtsensor, GPS, Barometer,...)
- Kabel-/Funkschnittstellen: USB, Wifi, Bluetooth, NFC(RFID)
- MEMS-Sensor: Bewegung einer Resonanzmasse verändert Kapazität messbar (Beschleunigungen, Kreisbewegungen); Anwendung: Schrittzähler, Anzeigeausrichtung ändern, Bewegung in Räumen

- Näherungssensor: Reflektion von Infrarotlicht auf kurzen Distanzen(z.B. beim am Ohr halten); Anwendung: Display ausschalten bei Telefonat, Gestensteuerung
- Magnetsensor: Messung des Erdmagnetfeldes im 3-dimensionalen Raum (Hall-Effekt); Anwendung: VR Brillen, präzisere Lagebestimmung

Technologien zur Barrierefreiheit von Apps

- 4 Prüfkriterien:
 - Wahrnehmbarkeit (z.B. statt Pixeln Bildbeschreibung, Formeln; Untertitel)
 - Bedienbarkeit (z.B. Gesteneingabe führt zu Link, Tastaturunterstützung)
 - Verständlichkeit (z.B. Gebärdensprache, Leichte Sprache)
 - Robustheit (trotz Javascript, Einsatz von ARIA)
 - Prüfwerkzeuge: WAVE Toolbar für Bedienbarkeit und Wahrnehmbarkeit von Webseiten; Juicy Studio für Wahrnehmbarkeit von Farbkontrasten
- Sehbehinderung(Mangel an Überblick): Vergrößerungsmöglichkeiten, hoher Kontrast, Farbumkehr, Farbkorrektur
- blinde Menschen(Pixelbarriere): Talkback, Tastatureingabe, Gestensteuerung
- motorische Behinderung(fehlende Bedienbarkeit): Reaktionszeit berühren/halten, Sprachsteuerung
- Hörbehinderung(multimediale Barriere): Untertitel, Interpretation von Geräuschen
- Talkback(Screenreader): filtert darzustellende Schriftzeichen, Eingaben und Widgets; verbalisiert Änderung oder Ausgaben; ermöglicht Erkunden des Bildschirms (klick \Rightarrow Sprachausgabe, doppelklick \Rightarrow auswählen)

- Untertitel: Schriftform für akustische Inhalte, folgen den Szenen in Videos; Erstellt durch Spracherkennungssoftware, Sprachdolmetscher; werden in WebVTT gespeichert

App Layouts

- Apps bestehen aus Activities, diese aus Widgets(für input/output), welche durch Layout-Elemente angeordnet werden
- Hauptsächlicher Unterschied: keine Fenster sondern Activities, Vereinigung mehrerer einzelner Hardware Komponenten in einem Gerät \Rightarrow neue Sensordatenutzungsmöglichkeiten; Vergleich per WIMP(Window, Icon, Menu, Pointer)
- Toolkits in Guis: Label, Menü (Auswahl weniger Optionen und Selektion), Button, Radiobutton(=1 auswählen), Toggle Button(on/off), Checkbutton(≥ 0 auswählen), Textfield, Scrollbar, Liste (Auswahl einer aus vielen Optionen), Tree(Auswahl aus verschachtelten Optionen), Toast(temporäre Meldung unten am Bildschirm)
- GUI: Views(Interaktionsobjekte, widgets) Module der GUI für Interaktion mit der Activity; Activities füllen Bildschirm aus; App kann mehrere Activities enthalten; Steuerung der Anordnung der Widgets über Layoutmanager(View Group[keine Interaktion möglich])
- ViewGroup Layouts: **Relative Layout**(Nachfolger halten ihre *relative* Position ein), **Linear Layout**(steuert Position der Nachfolger); Anordnung der Komponenten(Views und Layouts) des GUI **hierarchisch**
- Edit Text: Eingeben von Text und Zahlen über virtuelle Tastatur; anfügen, löschen und einfügen an der Schreibmarke möglich;(Detail: über Substrings von bereits eingegebener Zeichenkette Zeichen dazwischen einfügen)
- Gestensteuerung: Eine Geste ist jegliche Bewegung der Gliedmaßen des Körpers oder eine Pose; von Widgets unterstützt, Navigation im Textfeld über Wischgesten mgl.

- Widget-Eigenschaften können über Properties-Fenster oder über XML Datei festgelegt werden
- Eingaben in ein Widget werden durch Android vorverarbeitet, zusammengefasst und führen zum Aufruf einer Verarbeitungsmethode (Java), welche die Ausgabe berechnen

MCI

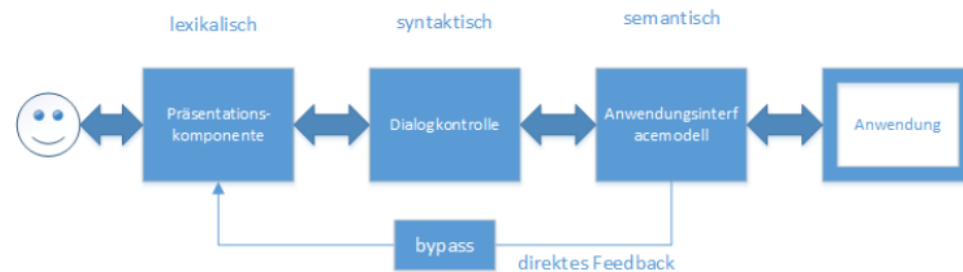
Universelles Design für...

- **Seherschädigungen:** Eingabe über (Braille)tastatur, Gesten; Ausgabe über Sprachsynthese-Screenreader oder Braille-Screenreader, beides wird z.B. von IOS und Android unterstützt, Braille-Screenreader z.T. auch für taubblinde Menschen nützlich
- **Farbenblindheit:** relativ häufig können Menschen z.B. Rot und Grün nicht unterscheiden \Rightarrow nicht nur Farben verwenden um Informationen zu vermitteln
- **Hörbehinderungen und gehörlose Menschen:** heutzutage kann z.B. Google Pixel automatisch Untertitel für alle Inhalte auf dem Smartphone erstellen, auch Ausgabe in Gebärdensprache z.B. im Fernsehen möglich, Eingabe per Tastatur/Gesten
- in IOS 14 Geräuscherkennung um akustische Signale zu visualisieren, Gebärdensprache ist aber noch immer nicht maschinell übersetzbar, Schrift für gehörlose Menschen schwer nutzbar
- **Menschen mit Körperbehinderungen:** Interaktion durch Alternative Augmentative Communication (AAC): Saug-Blaseschalter, Zwinkern, Eyetracking, Kopfgesten; außerdem Spracheingabe
- **Menschen mit geistigen Behinderungen:** z.B. Große Symbole ("BLISS"); Sprachein-/Ausgabe für Menschen mit Dyslexie (Leseschwäche), einfache Sprache
- **Ältere Menschen:** Sprachein-/Ausgabe, hoher Kontrast, große Icons

- **Menschen aus anderen Kulturen:** einfache Sprache
- **ohne Sprache:** eingabe über Gesten/Gebärden; Ausgabe per Display/Symbole

Interaktion

- klassisches Model ist **EVA**, dort werden aber Benutzer nicht berücksichtigt
- **Seeheim Modell** für GUIs, berücksichtigt wird darin, dass Anwendungen z.T. auch ohne Eingabe Präsentation erzeugen (z.B. Crash Report)



Interaktionstechniken

- **Menüs:** funktionieren z.B. auch rein textbasiert, sollten nicht mehr als 7 ± 2 Verschachtelungen enthalten, "Breadcrumbs" zur Orientierung (z.B. Auswahl | Auszahlung | Betrag)
- **Frage/Antwort:** z.B. Abfragesprache wie SQL (in Menüs z.B. bei Online Registrierungen: Frage Bundesland, Antwort Sachsen; Frage PLZ...); Alexa, Siri, können aber nicht immer hilfreich antworten
- **GUIs mit WIMP** = Windows Icons Menus Pointer (\Rightarrow Fenster ermöglichen parallele Bearbeitung von Aufgaben)
- **Formular/Dialog:** modale Dialoge erzwingen Abschluss des Dialogschritts (z.B. Druckermenü); nichtmodale Dialoge erlauben parallele Bearbeitung von Aufgaben (z.B. Rechtschreibprüfung)

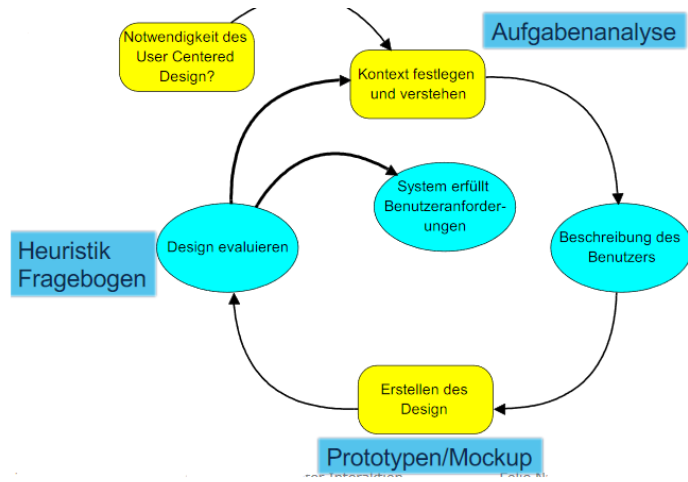
- **Drag & Drop:** z.B. für Fensterverwaltung und verschieben von Dateien
- **3D Interaktion:** VR/AR, Motion Tracking, Smartphone GPS
- Kommandozeile, Zeigen und Gesten, Multitouch

Anforderungsanalyse

- eine Methode um kleinere Interaktionsschritte festzulegen ist **Card Sorting**: es wird ermittelt wie Benutzer Dinge benennen, gruppieren und die Gruppen benennen

- unterschieden wird zwischen Mindestanforderungen und Zusatzfunktionen
- **geschlossenes Card Sorting** = vorgegebene Gruppen (Nachteil: weniger Entscheidungsfreiheit für Benutzer), **offenes Card Sorting** = ohne vorgegebenen Gruppen (Nachteil: schwierigere Auswertung (oft manuell))
- unterschieden werden muss zwischen Anforderung und Umsetzung: nicht "App braucht Play/Pause Knopf" sondern App hat Anforderung: Lied starten/stoppen, mögliche Umsetzung: Button
- **Ethische Fragen** Werden die Daten anonym oder zumindest pseudonymisiert gespeichert? Können/Sollen z.B. behinderte Menschen oder Kinder teilnehmen? Haben Probanden Einverständnis und Freiwilligkeit erklärt, kennen sie den Zweck der Studie?

Usability Engineering



- wer ist Zielgruppe (expertise/eingeschränkt...)?, was ist der Nutzungskontext (online/offline, abgelenkt/konzentriert...)? welche Anforderungen ergeben sich daraus? wie setzt man diese um?
- **Lo-Fi Prototypen** = Low Fidelity d.h. Skizzen, Storyboard, Wireframe (Gittermodell von z.B. Website, halt alles ohne Schriftart, Farben, Bildern), Papier (auch für haptische Oberflächen wie Braille), **Middle**: Mock-ups (Programm mit Papier und Pappe basteln), Wizard-of-Oz (Interaktion wird durch Menschen "hinter der Bühne" nachgebildet)
- **Hi-Fi Prototypen** = High Fidelity d.h. Powerpoint, HTML, Programm grob entwickeln so dass Interaktionstechniken implementiert sind
- Prototypen verzichten oft auf optimale Effizienz und Zuverlässigkeit, sie haben oft Pseudodaten, vereinfachte Algorithmen, nur eine Zielplattform
- **Effektivität** (von Effekt): tut es das was es soll? **Effizienz**: ist es Ressourcensparen? (auch nicht nur bzgl. Zeit); **Zufriedenstellung**

Evalutaion

- = Bewertung eines interaktiven Systems(von Experten oder Benutzer)

- Ziel: Feststellung der Qualität der Benutzungsoberfläche
- Qualitätsmerkmale: Effizienz(in angemessener Zeit), Effektivität(es funktioniert), Zufriedenheit
- Methoden: Benutzer einzeln Befragen; Benutzer mit Fragebogen befragen
- Heuristik nach Nievergelt: Wo bin ich?, Was kann ich tun?, Wie kam ich hierhin?, Wo kann ich hin?

Heuristik

- = Verfahren, um Probleme zu lösen bei begrenztem Wissen und wenig Zeit
- Heuristik zur Barrierefreiheit: **Wahrnehmbarkeit**, Bedienbarkeit, Verstehbarkeit, Robustheit
- Barrierefreiheit: **Umfang**, in dem Produkte, Systeme,... durch Menschen mit weitesten Benutzererfordernissen genutzt werden können um Ziele zu erreichen
- Figur-Grund-Wahrnehmung(selektive Interpretation und visuelle Unterscheidung) wird beeinflusst durch Wahrnehmung des Farbkontrasts

Farbe

- Zäpfchen reagieren auf untersch. Wellenlängen (R,G,B), Stäbchen reagieren bei wenig Licht (schwarzweiß)
- Farbkontraste: Wahrnehmung beeinflusst durch **Farbton, Helligkeit, Sättigung**; Definition von Kontrasten in Farbmodelle
- RGB Farbmodell: rot, grün, blau; sRGB-Intervall[0,1]; 8bit-Intervall[0,255] oder in hex[000000,FFFFFF]; $R_{sRGB} = \frac{R_{8bit}}{255}$ entwickelt für **additives** Farbmodell(Röhrenmonitore); Umrechnung in anderen Farbraum für andere Technologien(LED, DLP)
- Der Farbkontrast von zwei Farben ist dann ausreichend, wenn ihr Helligkeitswert das Verhältnis von $t=3$ (empirische Festlegung) oder besser aufweist $t = \frac{L_1 + 0,05}{L_2 + 0,05}$;

- $t=4,5$ für normale und kleine Schrift (14 pt) für Level AA nach WCAG 2.0
- $t=3$ für große oder fette Schrift (18pt bzw. 14 pt fett) für Level AA...
- $t=7$ für normale und kleine Schrift (14 pt) für Level AAA nach WCAG 2.0

- Farbraum: zeigt welche Farben ausgenutzt werden können; für sRGB Weiß[0.3127,0.3290]
- Farbkontraste: Berechnung der Luminanz im sRGB Farbraum

Evaluation mit Benutzern

- Methoden: Gebrauchstauglichkeit (Design, Äußere Benutzung, Zeit-/Fehlermessung), Benutzerstudien (beobachten/analysieren, der Ausführungen der Benutzer), Benchmarking(Kriterien: Objektivität, Validität, Reliabilität[Test geeignet für Prüfkriterium?])
- Messgrößen(objektiv): Aufzeichnungen, Graphen, Zählungen(quantitativ, thematisch), Muster erkennen(qualitativ)
- Diagramme berichten über Daten und erfinden keine zusätzlichen Daten (Punkte nicht verbinden, wenn keine Zwischenwerte existieren)
- deskriptive Statistik: **Boxplot** zur Visualisierung(Minimum, 1.Quartil, Median(Zentralwert der Messdaten), 3.Quartil, Maximum)

Fragebogen

- Hypothesen formulieren, Variablen und Merkmale dieser klären, zu erhebende Messwerte klären
- Anfangs allgemeine Fragen zur Person usw.; offene Fragen vorteilhaft aber schwer auszuwerten; negierte Aussagen zur Lügnererkennung, klare Aussagen, keine doppelte Verneinung
- Likert Skala: bestehend aus bipolaren Behauptungen, Rating(Skala 1-5 als optimal erwiesen, da Mittelwert existiert)

- Aussagekräftigkeit: mehr als 80% stimmen völlig zu(Ceiling-Effekt), mehr als 80% lehnen ab(Floor-Effekt)
- erfüllen Gütekriterien: **Objektivität** (Ergebnisse unabhängig von Versuchsleiter), **Reliabilität** (gleiche Ergebnisse für denselben Gegenstand), **Validität** (misst was er messen soll)
- NASA Task Load Index(TLX): Bestimmung des kognitiven Workload (mentale belastung); erhebung nach ausführung der Aufgabe
- TLX Fragenkatalog: ermittelt subjektive mentale Belastung (workload) bei Bearbeitung einer Aufgabe oder im Umgang mit einem System
 - Mental Demand (MD): How mentally demanding was the task?
 - Physical Demand (PD): How physically demanding was the task?
 - Temporal Demand (TD): How hurried or rushed was the pace of the task?
 - Performance (OP): How successful were you in accomplishing what you were asked to do?
 - Effort (EF): How hard did you have to work to accomplish your level of performance?
 - Frustration (FR): How insecure, discouraged, irritated, stressed, and annoyed were you?
- anschließend Gewichtung der Fragen und Asuwertung; TLX kein perfektes System

Text

Formeln und Umrechnungen

- **Entropie** $H = -\sum_i p_i \cdot \log_2(p_i)$ (durchschnittlicher Informationsgehalt)
- **mittlere Wortlänge** $L = \sum_i p_i \cdot N_i$
- **Redundanz** $R = L - H$, wobei $H \leq L$ (trägt nicht direkt zu zu Information bei; nützlich z.B. für Fehlerkorrektur)

- Basiseinheit bit, 8 bit = 1 byte, 1024 byte = 1 Kib, 1024 Kib = 1 Mib etc.

Kodierungsverfahren

- mit n bit können 2^n zeichen codiert werden (dabei ist aber nich automatisch Fano-Bedingung erfüllt)
- **Fano Bedingung** (auch Präfixeigenschaft): kein Zeichen bildet den Anfang eines anderen Zeichens (z.B. Morse Code N= -•, C= - • -• \Rightarrow Fano Bedingung nicht erfüllt)
- **Fano Codierung**: "WSKn absteigend sortieren, dann immer in 2 Gruppen teilen"; siehe C-Programm
- **Huffman Codierung**: "immer letzten beiden WSKn zusammenfassen, Codebaum erstellen" siehe people.ok.ubc.ca/ylucet/DS/Huffman.html sind WSKn gegeben dann z.B. für (Zeichen X mit WSK $\frac{65}{100}$) in Python Konsole eingeben: `print(65*"X")`
- Es gilt: Huffman ist **effizienter**, Fano höchstens genau so effizient! (messbar über R und L)
- **Zweck**: Verschlüsselung, Redundanzvermeidung, Digitalisierung analoger Signale, Schutz etc.

Zeichensatz

- **Braille**: gibt es für fast alle Sprache, allerdings hoher Pflatzbedarf
- **ASCII**: 128 Zeichen mit je 7 bit, Zeichen 0 bis 31 sind Steuerzeichen (wie unser Enter...); Darstellung oft als Hexadezimal
- **Latin**: 8 bit Zeichensatz, erste 128 Zeichen identisch zu ASCII, es gibt verschiedene Varianten je nach Region (Latin-6 Nordisch, Latin-5 türkisch etc.)
- **Unicode**: internationaler Standard, Anzahl codierbarer Zeichen variiert nach Version,in diesem Dokument z.B. UTF-8 \Rightarrow "A" wird zb als 41 (Hex) codiert (detaillierte Codierungsregeln siehe Folie 20 Medien-Text)

Schrift

- Schriftgrößeinheiten für DTP Standard (Achtung: bei anderen Standards ist z.B. 1 point anders definiert): 1 pica = 12 points = $12 \cdot 0,3527$ mm
- **Serifen** (die kleinen "Verzierungen" an den Enden aller Buchstaben) helfen bei Lesbarkeit
- bei **Rasterschrift** bestehen Zeichen aus Pixeln
- bei **Vektorschrift** bestehen Zeichen aus Kurven (\Rightarrow wird beim reinzoomen nicht verpixelt)

Formatierung

- **Kerning/Unterschneidung**: Abstand zwischen zwei Buchstaben wird vom Textverarbeitungsprogramm manchmal gesondert angepasst, z.B. bei parallelen Diagonalen bei Text "AV", Diagonalen vom A und V werden näher aneinander gerückt
- neben **Blocksatz** gibt es **Flattersatz** (Links-/Rechtsbündig) und **Mittelachsensatz** ("zentriert")

Formate und Software

- eine Eigenschaft von Editoren ist **WYSIWYG** ("what you see is what you get"; z.B. bei Word, z.B. nicht bei Latex)
- **Rich-Text**: z.B. in MS Word; nicht programmierbar
- **HTML**: Integration verschiedenster Medien möglich, erweiterbar durch Plugins, "Aussehen" eines Dokumentes wird getrennt vom allen anderen Sachen im "Style sheet" festgelegt, Dynamisch durch Scripting
- **Latex**: programmierbar, gut für Formeln, Latex ist nur die Interface-Sprache für das Textsatzsystem TeX
- **PDF**: kann nur mit spezieller Software bearbeitet werden, Integration verschiedenster Medien möglich, Layout-treu ("wird immer gleich wiedergegeben")

XML(eXtensible Mark-up Language)

- Medien in Dokumenten definieren sich durch **Inhalt** (Semantik), **Struktur** (Syntax) und **Präsentation** (lexikalische Information)
- Für Arbeiten mit HTML bedient man sich der Sprache, Werkzeugen und Arbeitsmethoden (Layouten o. ä.). HTML basiert auf XML, das als **Metasprache** Text beschreibt und strukturiert (Baumstruktur).
- Dokumente bestehen aus Tags
`<person>Der Bart</person>` oder für leere Elemente `
`; Elementname (person) legt das Tag als Ganzes fest. Metazeichen von XML sind entsprechend zunächst `<`, `>` und `/`.
- Elemente können Elemente enthalten (Schachtelung); Elemente können Text enthalten; Groß- und Kleinschreibung bei Elementnamen beachten; Das öffnende Tag muss genauso lauten wie das Schließende
- In XML-Dokumenten immer XML-Deklaration am Dokumentenbeginn: `<?xml version="1.0"?>`. Folgen kann dann ein **Stylesheet** und eine **DTD** sowie die **Daten**.
- **Wohlgeformtes XML** ist syntaktisch korrekt (s. o., mit korrekten Schachtelungen, richtig verwendeten Metazeichen, genau ein Wurzelement, etc); **Valides XML** erfüllt die angegebene DTD; mittels Parser prüfbar
- **Syntaxdiagramme**: Es gibt ein (nichtterminales) Startsymbol; Symbole in Kreisen sind terminal; Symbole in (abgerundeten) Rechtecken sind nichtterminal, beziehen sich auf andere Symbole; Ausdruck ist "wohlgeformt", wenn ein Durchlauf durch die Diagramme möglich ist
- **Wichtige Begriffe**: Elemente, Knoten (Tags), Vorgänger, Nachfolger, Nachbarknoten, Wurzel, Attribute, Text
- hierarchischer Aufbau in XML Bäumen, Reihenfolge wichtig

- XML-Tags müssen mit Buchstaben oder `'_'` anfangen, danach alphanum. Zeichen mgl.; dürfen keine Metazeichen (`{}`, `"/` etc.) oder Leerzeichen enthalten
- **Entitäten** in XML "definieren Kurzformen für Standardtext oder Sonderzeichen" z. B. `<`, `>`, `"` für das `<` `>` "-Zeichen; (`&...;`) sind "Fluchtsymbole" in XML. Dadurch können alle Metazeichen in XML auch verwendet werden (`&` wird durch `&` gesetzt)
- `<![CDATA[<svg></svg>]]` ermöglicht Textabschnitt ohne Fluchtsymbole (alle Metazeichen zulässig außer `]]`)
- In der XML-Deklaration kann **Attribut** `encoding="iso8859-1"` und `standalone="no"` (gibt an, dass die DTD im Dokument selbst drin ist) verwendet werden
- Kommentare: `<!-- Kommentar -->`
- Element **Attribute**: ist ein Paar "*Attributname=Attributwert*", kann an Elementname angehängt werden; mit Anführungszeichen begrenzt; mehrere pro Element möglich
- **Processing instructions** weisen auf Verarbeitungsschritte hin;

DTD

- DTD ist selbst kein XML, definiert Struktur, gibt Aufbau der XML-Daten mit Regeln (legt Reihenfolge der Elemente fest) an, beschreiben nicht die Semantik der Daten
- Beispieldefinition:

```
<!DOCTYPE person [  
<!ELEMENT person (fname, profession)*>  
<!ELEMENT profession (#PCDATA)>  
<!ELEMENT fname (#PCDATA)>  
>  
<person>  
<fname>Tim</fname>  
<profession>Troll</profession>  
<profession>Toll</profession>  
</person>
```

- `<!ELEMENT (...)>` definiert ein Element und in Klammern die erwarteten Kindelemente, in fester Reihenfolge. Man kann `?` für ein-oder-keinmal anfügen, `*` für beliebig oft (oder gar nicht), `+` für ein oder mehr; Alternativen möglich mit `((a, b) | (c, d))`; `#PCDATA` für beliebig viel normalen Text; `EMPTY` (statt der Kindelementliste \Rightarrow besitzt keine) wenn Element immer leer sein soll; `ANY` für komplett egal
- **Textzeichen mit Elementen mischen**:
`<!ELEMENT paragraph (#PCDATA | name | profession | footnote | emphasize | date)*>`
(kann durch `*` beliebig oft permutiert werden)
- **Attributtypen** sind z. B. `CDATA` (Text), `ID` (eindeutiger XML-Name), `NMTOKEN` (wohlgeformter XML-Name), `IDREF` (Name eines anderen im XML vorkommenden Elements [für eindeutige sichere Verknüpfungen]), `IDREFS` (Mehrere Namen von anderen XML-Tags im Dokument, getrennt durch Leerzeichen)
- **Attribute** für Elemente sind `#REQUIRED` (muss vorhanden sein), `#FIXED` (muss immer einen bestimmten Wert haben, der danach angegeben wird), `#IMPLIED` (optional). Alternativ kann eine beliebige Zeichenkette geschrieben werden, die den Standardwert angibt.
- `<!ATTLIST name first CDATA #REQUIRED>` fügt `Attribute(first)` zu einem `Element(name)` hinzu
- XML-Namen gehören einem Namensraum, d.h. einer Syntaxbeschreibung an; durch Präfix können identische Elemente (XML-Namen) verschiedener Namensräume unterschieden werden; Alle Elemente verschiedener DTDs müssen unterschieden werden; (wird bei Elementen durch allgemeingültiges Attribut `xmlns` deklariert) Möglichkeit SVG oder HTML Code in XML zu integrieren
- XML für Dokumente: Text Encoding Initiative (TEI) mit TDT für Theater o. ä.
- DocBook / OpenEBook: XML für Bücher wie Informatikbücher

- MusicXML: zum Musikstücke digitalisieren
- DTDs können keine Zahlen oder Maximallängen vorgeben, keine Modularisierung, DTDs sind keine XML-Dokumente

XSLT

- Daten von der Präsentation trennen, Transformation von XML-Struktur in neue Struktur, Formatierung der neuen Struktur
- Einbindung eines XSLT-Stylesheets:

```
<?xml-stylesheet type="text/xsl"
href="helloworld.xsl"?>
<?xml version="1.0" encoding
="iso-8859-1"?>
<greeting>Hallo Welt!</greeting>
```

Beispiel des XSLT-Stylesheets:

```
<xsl:stylesheet version="1.0"
xmlns:xsl="...">
<xsl:template match="/">
<html><head>
<title>Wort zum Tag
</title></head>
<body><p>
<xsl:value-of select="greeting"/>
</p></body></html>
</xsl:template>
</xsl:stylesheet>
```

- Verarbeitung basiert auf Templates, auch ECMA-Script möglich
- `<xsl:apply-templates>` führt die Templates, bei Angabe von `select` nur auf entsprechende Elemente, `<xsl:value-of select="elemname">` holt den Wert eines XML-Tags und fügt ihn an dieser Stelle ins Dokument ein, `<xsl:for-each select="a">` gibt für jedes folgende Element den definierten Text aus, `<xsl:template match="people">` gibt für alle `people`-Elemente im XML den definierten Text aus

- Mit dem Attribut `mode` für `apply-templates` und `template` kann Status definiert werden, z. B.
- `<xsl:variable name="width" select="50"/>` legt Variable `width=50` fest
- `<xsl:if test="position()=1"></xsl>`
- `<xsl:for-each select="book"></xsl>`
- `<xsl:sort select="author"></xsl>`
- XPATH wird in XSLT benutzt, um Tags anzusprechen (`match`-Attribut). ist die Wurzel, Tag-namen werden als Nachfolger interpretiert, der zugehörige Tag eines Attributes kann mit `@born` angesprochen werden
- In XPATH kann gerechnet werden (`div`, `+`, `-`, `*`, `mod`, `position()`, `first()`, `count()`, `round()`, `substring()`)

Bild

SVG

- EncapsulatedPostScript, Bildinformationen in speziellem Code, programmierbar, keine Alphakanäle, nur PostScript-Geräte
- Scalable Vektor Graphics (SVG) in XML für Bilder, Text, Animation, Formen (bspw. mit Inkscape malbar)
- Pfade: `<path d=""></path>`, Möglich anzugeben: M x y (Zeichenstift ohne zu zeichnen bewegen), L x y (Linie zum Ziel malen), Q x1 y1 x2 y2 (Quadratische Bezier-Kurve), C x1 y1 x2 y2 x3 y3 (Kubische Bezier-Kurve), Z (Pfad schließen), H x (Horizontale Linie), V y (Vertikale Linie), A ??? (Ellipsenbogen), S x y (Quadratische Bezierkurve fortsetzen, vorherigen Kontrollpunkt spiegeln) - Groß geschriebenes Zeichen steht für absolute, kleines für relative Koordinaten. Pfade beginnen immer mit `moveto` (M) und können mit Z enden.
- Primitive: circle-Element mit Attributen `cx`, `cy` und `r` (CenterX und CenterY), ellipse mit `cx`, `cy`, `rx` und `ry`, rect mit `x`, `y`, `width`, `height`, line mit

`x`, `y`, `width`, `height`, polygon / polyline mit points (`x1 y1 x2 y2...`)

- Text: text-Element mit `x`, `y`, `size`
- Image: image-Element mit `x`, `y`, `width`, `height`, `xlink:href`
- Einfärben von Formen mit `fill`-Attribut und einer Farbe in Hex (`#FF0000` z. B.) oder Name.
- Rand malen mit `stroke`-Attribut (enthält Farbcode) und `stroke-width`-Attribut (Randbreite in Pixeln)
- transform-Attribut gibt Rotation, Transformation, Skalierung oder Stauchung an: z. B. `"translate(20, 150) scale(0.5) rotate(30,255,230) skewX(25) skewY(25)"` - Bei Rotate dienen der zweite und dritte Parameter dabei als Rotationszentrum
- Clipping: Beschränken eines zu zeichnenden Elements auf den Bereich, den ein anderes Element definiert
- Filter in SVG: "Fotorealismus" durch Unschärfe, gezielte Einbringung von Bildstörungen o. ä.
- Blurring: Unschärfefilter
- Events in SVG mit JavaScript definieren (svg onload-Event, onclick-Event etc.)
- `<ellipse id="elli" cx="100" cy="100" rx="48" ry="90" style="fill:green;" /><animate xlink:href="#elli" attributeType="XML" attributeName="rx" begin="0s" dur="6s" from="48" to="198" fill="freeze" />`
- In `<defs>...</defs>` können SVG-Elemente definiert werden, die nicht gezeichnet werden, sondern mit `<use xlink:href="#id">` später wieder benutzt werden

Licht

- Licht ist eine elektromagnetische, transversale Welle (schwingt parallel zur Ausbreitungsrichtung)
- Lichtgeschwindigkeit im Vakuum $c = 299792458 \frac{km}{s}$ ist vom Medium, in dem sich das Licht bewegt, abhängig (meist Luft), $c = \lambda * f$ (f: Frequenz des Lichts, λ : Mediumskonstante)
- Frequenz ist Maß für Energie $E = hf = \frac{hc}{\lambda}$ (h ist das Plancksche Wirkungsquantum mit ca. $h = 6.63 * 10^{-34} Js$), d. h. je höher die Frequenz, desto energiereicher ist das Licht.
- Frequenz des sichtbaren Lichts beträgt ca. 400 nm - 780 nm, davor liegen die Frequenzbereiche von UV-Licht, Röntgen- und Gammastrahlung, darüber die von infrarotem Licht, UKW-Radio etc
- Je höher die Frequenz, desto geringer die Wellenlänge (antiproportional)
- Lichtemission: Durch Gasentladung (z. B. Xenon Lampen), Photolumineszens (Phosphor "wandelt" einfallendes Licht in andere Wellenlänge um)
- Chemische Reaktion: Verbrennung (Flamme strahlt Licht)
- Phosphor: Elektronenbeschuss bringt Phosphor zum Leuchten
- Absorption: Bspw. beim Durchlaufen einer farbigen Folie wird ein Teil des Lichtspektrums absorbiert
- Reflexion: Ein Teil des Lichts wird beim Auftreffen auf eine Oberfläche reflektiert
- Additive Farbmischung: Überlagerung von Licht mehrerer Emissionsquellen, Spektren werden addiert
- Subtraktive Farbmischung: Einfallendes Licht fällt auf Filter (Absorption / Reflexion) ("Multiplikation mit Filterfunktion")
- Gedruckte Farben erscheinen je nach Beleuchtung unterschiedlich (z. B. Geld wird rötlich bei roter Beleuchtung...)

- Farbtemperatur in Kelvin: "Wärme" des Lichtes, je höher desto weißer - Glühlampe ca. 2800K, 6500K: Tageslicht
- Mensch "sieht" Lichtintensität, visuelle Wirkung, Konstrast, Formen, Bewegung
- Sehen beeinflusst Wachsamkeit, Wohlbefinden, innere Uhr [...]
- Spezielle Rezeptoren, spezieller Nervenpfad im Hirn
- Retina im Auge mit Photorezeptoren besetzt, Fovea im Zentrum des Sehfeldes, "blinder Fleck" des Auges ca. 1,75mm im Durchmesser (Gehirn "vervollständigt" das Bild)
- Farbwahrnehmung in der Retina erfolgt durch Zapfen und Stäbchen (Zapfen: Farbwahrnehmung, Stäbchen: Schwarz/Weiß)

Farbmodelle

- RGB-Farbraum: 256 Farbstufen pro Rot, Grün und Blau, also ca. 16 Millionen Farben
- CIE 1932: xyz-Kanäle, y: Wahrgenommene Helligkeit, x: rot/grün, z: blau/gelb-Unterschied
- Farbwahrnehmung: Helligkeit (Luminance), Farbton und Sättigung (Saturation) hauptsächlich
- Hirn teilt wahrgenommene Farben in weiß, grau, schwarz und verschiedene Farben
- 128 Farbtöne (Hues), 130 Farbsättigungen (Saturation Levels), 16 (blau) - 26 (gelbe) Helligkeitswerte, also ca. 7 Millionen wahrnehmbare Farbnuancen
- Farben in Text: Hervorhebung, sparsam (auffällige) Farben einsetzen
- Räumliches Sehen durch Augenabstand (Fusion, Disparität) zum Erkennen von relativer Größe, Perspektive, Beleuchtung
- Hirn bildet 3D-Modell (Fusion): Unterdrückung unvereinbarer Regionen, bei Überlagerung wechselt unvereinbares linkes / rechts Bild

- Räumliches Sehen: Fokus (Raumpunkt, auf den die Augen fokussiert sind), Retinale Position (relative Position zur optischen Achse), Mentales Auge: Fusion kombiniert Information
- Disparität: Unterschied von fusionierten Punkten kann zur Berechnung der relativen Position benutzt werden
- Munsell Farbsystem (Farbton hue, Helligkeit value, Sättigung chroma)
- HLS Farbsystem (Hue, Luminance, Saturation)
- CMY - Farbraum (Cyan, Magenta, Yellow)- $(r, g, b) = (1, 1, 1) - (c, m, y)$
- CMYK - CMY mit Schwarz, damit Schwarz nicht aus Farbe gemischt werden muss (für Drucker toll)
- YUV - Y: Helligkeit (s / w), U, V: Farbbalance
- YC_bC_r - Y: Helligkeit, b: Blau, r: Rot

Digitale Bilder

Digitale Bilder bestehen aus Pixeln ($w * h$), die pro Pixel meist 8 - 32 bit Farbinfos beinhalten (Farbtiefe des Bildes) (ab 24 bit: TrueColor)

- Indizierte Farbformate: Zu Beginn des Bildes Farbtabelle, danach wird jede Farbe mit einem Index auf die Farbtabelle angegeben (sinnvoll bei wenigen verwendeten Farben)
- Chroma Subsampling bei YUV und YC_bC_r (4 : 2 : 2): Halbe Auflösung für g/b (U) und r/gr (V)-Kanäle
- Alphakanal: Transparenzangabe
- Auflösung des Auges: ϕ minimaler Sehwinkel, bei dem 2 Linien im Abstand s erkannt werden, $s = 2a * \tan(\frac{\phi}{2})$
- dpi / ppi: dots per inch / pixels per inch (Pixel pro Zoll, $1in = 2,54cm$)
- Monitor ca. 95 ppi, Laserdrucker ca. 1400 dpi

Digitale Bildformate

- Histogramm: Gibt Verteilung der Graustufen von 0-1 des Schwarzwertes an
- Bildfilter: Beispielsweise Weichzeichner (lineare Mittelwertbildung aus Nachbarn um einen Pixel herum)
- GIF, Graphics Interchange Format: Animierbares farbindiziertes Bildformat mit 256 Farbstufen, kein Alphakanal, verlustbehaftete Kompression
- PNG: Portable Network Graphics: verlustfreie Kompression als GIF, Alphakanäle, freies Format, keine Animation, nicht für Fotos geeignet (verlustfreie Kompression zu schwach), Interlacing (Zunächst werden nur einige Zeilen des Bildes übertragen und dadurch eine Vorschau möglich, bevor das ganze Bild übertragen ist)
- JPEG (Joint Photographic Experts Group): Verlustbehaftete Kompression u. a. mit Huffman-Kodierung, Bildung von 8x8-Blöcken - führt zu Blockbildung, JPEG-2000 als Nachfolger mit geringerer Artefaktbildung - Farbraumrechnung in YUV, Tiefpassfilterung & Unterabtastung der Farbdifferenzsignale U, V (verlustbehaftet 4 : 2 : 2, s.o.), Quantisierung, Umsortierung, Huffman-Kodierung. Gut geeignet für Fotos, nicht gut für alles andere (Diagramme etc)
- JPEG2000 basiert auf XML - ermöglicht Rechte-Management, verschieden starke Komprimierung in verschiedenen Bildteilen, Interlacing-Support
- TIF - Tagged Image Format: mehrere Farbräume wie RGB, LAB, verlustfreie Komprimierung mit LZW, schwache Komprimierung, können viele Browser nicht
- PSD - Adobe Photoshop, Kanäle, Bearbeitungsschritte, Text wird nicht gerastert, kann von vielen Programmen nicht gelesen werden

Vektorgrafik

- Vektorgrafik - Beschreibung des Bildes mittels Formen wie Rechtecken und nicht durch Pixel (Bsp.: SVG)

- Transformationen (Rotation, Translation, Skalierung, Scherung) können mittels der Multiplikation des alten Vektors mit einer Matrix und der Addition eines neuen Vektors ausgedrückt werden. $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$
- Translation: $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$
- Rotation: $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- Scherung: $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- Skalierung: $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- Bezierkurven dritten Grades / kubische Kurven: Kurven nach dem Format: $\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = (1-t)^3 \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + 3t(1-t)^2 \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + 3t^2(1-t) \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + t^3 \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}$ Dabei sind $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ und $\begin{pmatrix} x_3 \\ y_3 \end{pmatrix}$ die Start- bzw. Endpunkte der Kurve und $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ sowie $\begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$ definieren die Tangenten an diesen Punkten.
- de Casteljau-Schema: $t = \frac{1}{4}$
- TrueType-Fonts: Die Kontur von Zeichen werden durch Bezierkurven beschrieben, wird dann gefüllt
- Meta Font (L^AT_EX): Glyphenkontur wird durch interpolierende Kurve beschrieben
- Gruppierung von einzelnen Formen, damit Transformation auf alle Elemente angewendet werden können
- Tiefe: Definierung einer Koordinate z, damit die Überlagerungsreihenfolge bei überlappenden Formen definiert werden kann
- Rasterisierung: Übersetzung von Vektorgrafik in Pixelgrafik (durch Test jedes Bildpunkts, ob er in

einer Form liegt oder durch zeilenweises Abtasten [Sweep-line], oder durch Abtasten jeder Form und Setzen der ausgefüllten Pixel)

- Dropouts (bei TrueType-Fonts): Nur halb ausgefüllte Pixel werden falsch als nicht gegeben gesetzt, daher setzen TrueType-Abtaster einen Pixel zwischen zwei anderen Pixeln auch dann, wenn zwischen diesen Pixeln zwei Konturlinien der Glyphen liegen (aber nur dann, wenn beide Konturen in beide Richtungen fortgesetzt werden)

Wahrnehmung

- Motivation: Notwendige Daten an menschliche Wahrnehmung anpassen, um Platz zu sparen (z. B. nicht wahrnehmbare Töne bei Audio)
- Ergonomisch physische Dinge gestalten, Gestaltung von Benutzerschnittstellen, Layout etc. auf Menschen optimieren
- $Reiz \rightarrow Sensor(analog) \rightarrow Filter \rightarrow Quantisierung(Digitalisierung) \rightarrow Filter(digital) \rightarrow Kompression \rightarrow Codierung \rightarrow Speichermedium$
- $Speichermedium \rightarrow Dekompression \rightarrow Filter(digital) \rightarrow Rekonstruktion \rightarrow Filter(analog) \rightarrow Display(analog) \rightarrow Reiz(beim Mensch)$
- Wie funktioniert Reiz? Wie werden Reize beim Menschen wahrgenommen (Visuell, Akustisch, Haptisch, Gustatorisch / Geschmack, Olfaktorisch / Geruch), Multimedial: Sprache und Deixis, Multimodal: Gedächtnisleistungen. Wie stark wird welcher Reiz wahrgenommen? Wie gleicht man produzierten Reiz mit echtem Reiz ab?
- "Wahrnehmung ist der Prozess, durch den Lebewesen Kenntnisse über ihre Umgebung und über sich selbst in Beziehung zur Umgebung gewinnen. Sie ist der Anfang allen Wissens."
- Reize / Wahrnehmung / Neuronale Prozesse
- Funktionale Physiologie: Erkenntnisse durch Patienten mit verletzten Gehirnen, elektrische

Störungen von außen, durch teilweise Einschläferung, mit Elektro-Enzephalographie, Magneto-Enzephalographie

- Wahrnehmung ist relativ: Größe / Helligkeit
- Wahrnehmung der Länge erfolgt logarithmisch (bei längeren Strukturen müssen Unterschiede größer sein)
- Wahrgenommene Skalen für Größe von Kreisen ist abhängig nach vom Radius
- Spitze Winkel werden unter- und stumpfe Winkel überspitzt
- Reizschwelle R_0 : kleinster wahrnehmbarer Reiz (z. B. Helligkeit des dunkelsten, noch erkennbaren Sterns am Nachthimmel), quantitativs Maß (z. B. Pa beim Hören)
- Unterscheidungsschwelle ΔR : kleinster Unterschied von einem Grundreiz (z. B. Helligkeitsdifferenz für einen Punkt, den man gerade noch auf einem grauen Hintergrund wahrnehmen kann), qualitatives Maß
- Messverfahren erfordern Wahrnehmung von Reizen: Methode konstanter Reize (Zufällige Wiedergabe von verschiedenen starken Reizen, der zu 50% erkannte Reiz ist die Reizschwelle - sehr genau, langsam), Herstellungsmethode (VP stellt den Reiz selbst ein - ungenau, schnell), Grenzmethode (Reiz erhöhen bis ihn VP erkennt - genau, langsam)
- Empfindungsstärke Ψ , Intensität des Grundreizes R , Unterscheidungsschwelle ΔR , k von Sinnesmodalität abhängige Konstante
- Webersches Gesetz: $\frac{\Delta R}{R} = k$
- Weber-Fechnersches Gesetz: $\Psi(R) = c * \ln \frac{R}{R_0}$
- Quadriert man die Stärke des Grundreizes, dann verdoppelt sich die Empfindungsstärke
- Stevensches Potenzgesetz: $\Psi(R) = c * (R - R_0)^n$. Entscheidend ist der Parameter n . Für Helligkeit gilt $n = 0.3$, für ein Gewicht $n = 1.5$, für die Länge einer Linie $n = 1.0$

- Empfindungsstärken wie Lautstärke: Verwendung einer logarithmischen Einheit wie Dezibel (Weglassen von k)
- Bel: Logarithmus des Quotienten zweier Leistungen $Pegel = \lg \frac{R}{R_0}$, Dezibel: Zehntel eines Bels $Pegel = 10 * \lg \frac{R}{R_0}$
- Wahrnehmung: Extraktion elementarer Merkmale (Kanten, Farbe), Präattentive Verarbeitung (Erkennen elementarer Bestandteile), Attentive Verarbeitung (Konstruktion des Gesamten)
- Wörter eines Textes werden nicht Buchstabe für Buchstabe wahrgenommen (Auge liest 6-8 Buchstaben und springt dann weiter), Hören funktioniert ähnlich
- Gesetz der guten Gestalt / Prägnanz: Übergeordnetes Gesetz zur Wahrnehmung von Gegenständen, Zusammenführung von prägnanten Einzeleindrücken zu Gestalt
- Gesetz von Figur und Grund: Figur wird als vor dem Hintergrund stehend wahrgenommen, Figur und Grund kann nicht gleichzeitig wahrgenommen werden
- Gesetz der Gleichheit: Ähnliche Dinge erscheinen zu zusammengehörigen Gruppen geordnet
- Gesetz der Nähe: Dinge, die sich nebeneinander befinden, erscheinen als zusammengehörig
- Gesetz der Geschlossenheit: Von Linien umschlossene Gegenstände werden als zusammengehörig empfunden
- Gesetz des weiterführenden oder gleichen Verlaufs: Bruchstücke reichen zum Bilden von Gestalten aus
- Gesetz der Symmetrie: Symmetrische Gebilde werden als Figur wahrgenommen
- Gesetz des gleichen Schicksals: Objekte, die sich mit gleicher Geschwindigkeit bewegen, werden als zusammengehörig empfunden
- Gesetz der Erfahrung: Wiedererkennen von bestimmten Formen

- Gestaltgesetze betreffen die präattentive Wahrnehmungsphase und finden vor allem z. B. bei der Interfacegestaltung Anwendung (Gestaltung von Icons, Bedienoberflächen)
- Bei rivalisierenden präattentiven Prozessen kommt es zu Wahrnehmungstäuschungen, Linien werden gebogen
- Geone: Dreidimensionale Grundformen werden präattentiv anhand der Kantenanordnung erkannt und dann attentiv zu einem Gegenstand wie einem Tisch zusammeninterpretiert

Internetz

- **HTTP**: Protokoll zwischen Web-Server und Web-Client (Browser)
- **URI**: Bezeichnung und Adressierung beliebiger Daten im Netz (z. B. <http://www.wh2.tu-dresden.de>)
- **Hypertext**: "Nicht-lineares" Textmedium, verbindet Inhalte durch Hyperlinks (URI, entfernte URIs wie "<http://google.de>", "<ftp://...>" oder lokale URIs wie "<mailto:>", die an andere Programme auf dem Rechner gehen)
- **Mark-Up-Sprachen**: Beschreibung von Text durch Text, bspw. HTML mit öffnenden Tags und schließenden Tags und Inhalt dazwischen

HTML

```
<html><head><title>Hallo Welt</title>
</head><body><h1>Grosse Ueberschrift
</h1><a href="uri">Link</a></body></html>
```

- Attribute bspw. href mit Ziel-URI, align mit Ausrichtung o. ä.
- `<a>`-Elemente sind Anker, verlinkt wird per "#ziel" und gesetzt per name-Attribut
- Bilder mit ``

- Listen mit ul (unordered list) und ol (ordered list)-Elementen, Einträge mit li (list item)-Elementen
- Tabellen mit table, tr für Zeilen und td für Spalten.

```
<table><tr><th>1</th><th>2</th></tr>
<td>A</td><td>b</td></tr></table>
```

- div für allgemeine Blöcke

Barrierefreiheit

Barrieren für Behinderte auf Webseiten (Sehbehinderte, Farbenblinde, Gehörbehinderte, Motorisch behinderte, geistig behinderte)

CSS

Strikte Trennung zwischen Präsentation und Design, definierte Formate für XHTML-Tags. Text-Formatierung, Positionierung von Elementen, Rahmen und Leerräume, Hintergrund, Formatierung von Tabellen, CSS einbinden mit `<style type="text/css"></style>` für direktes Reinschreiben von CSS, CSS-Datei im Head des HTML-Dokuments verlinken mittels `<link rel="stylesheet" type="text/css" href="datei.css">`. HTML-Elemente können auch mit style-Attribut und CSS-Angaben gestyled werden.

CSS besteht aus Selektoren für Elemente (mittels ID oder Klasse) und Style-Attributen. Bspw.: `#foo { color: red; }` setzt die Schriftfarbe im Element mit der ID foo auf rot. Mögliche Styleattribute sind z. B. "font-family" (Arial, sans-serif), "font-weight" (bold).

Positionierung

Mit CSS werden Elemente nach dem Boxmodel layoutet, d. h. jedes Element bildet eine Box. CSS-Attribute für jedes Element sind z. B. "margin" (Außenrand um die Box des Elements, z. B. "5px"), "padding" (Innenabstand um den Boxrand), "border-width" (Randbreite), "border-color" (Randfarbe).

Navigationstechniken

Websites sollten sets einwandfrei navigierbar sein (vor / zurück, Nutzer muss sich zurechtfinden können). Bspw. durch Setzen von Navigationsleisten, Suchmaschinen etc.

Karten in HTML

Map-Element definiert Karte. Bsp.:

```
<map name="schland">
<area shape="rect" coords="11,10,59,29"
href="uri" alt="Hier"></map>

```

Beispielsweise rect mit coords="x1,y1,x2,y2" mit 1=Obere Linke Ecke,2=Untere rechte Ecke oder "circle" mit "x,y,r" oder "poly" "x1,y1,x2,y2..."

Dynamisches Web

Dynamik z. B. durch auf dem Web-Server dynamisch generierte Inhalte durch PHP, ASP.net o. ä. Im Browser wird JavaScript benutzt.

JavaScript

1. **Objekt:** Ein JavaScript-Objekt besteht aus Schlüsseln, denen Werte zugeordnet werden (bspw. das document-Objekt hat den Schlüssel bgcolor mit dem Wert white).

2. **Funktionen / Methoden / Aktionen:** Wie Funktionen in C, können auch Variablen zugeordnet werden
3. **Ereignisse:** Das Ereignis "onmouseover" eines HTML-Elements kann durch JavaScript auf eine Funktion gesetzt werden.
4. **Variablen:** Können Zahlen, Strings, Booleans, Objekte, Funktionen sein. Bsp. zur Deklaration: `var month = "Januar";`
5. Ausdrücke ordnen einer Variable einen Wert zu (Beispiel s. o.)
6. Operatoren verändern Variablen (siehe C, bspw. "+", "==")
7. Einbinden einer JavaScript-Datei in HTML:

```
<script src="js.js"
type="text/javascript"></script>
```

Einbetten von JS in HTML:

```
<script type="text/javascript">
```

Zugreifen auf Dokumenteigenschaften in JavaScript über das globale window-Objekt, bspw. "window.location.href = 'http://google.de'". window-Objekt enthält document-Objekt, über das HTML-Elemente mittels "getElementById" abgefragt werden können.

HTML5

HTML5 als zukünftiger Standard führt HTML-Elemente für Strukturierung ein (header, nav) und neue Dinge wie das Video-Tag.

Gebaut mit L^AT_EX