



CHARITÉ CAMPUS BENJAMIN FRANKLIN

Institut für
**Medizinische Informatik,
Biometrie und Epidemiologie**

Univ.-Prof. Dr. rer. nat. Thomas Tolxdorff

Informationsverarbeitung

Prof. Dr. Thomas Tolxdorff

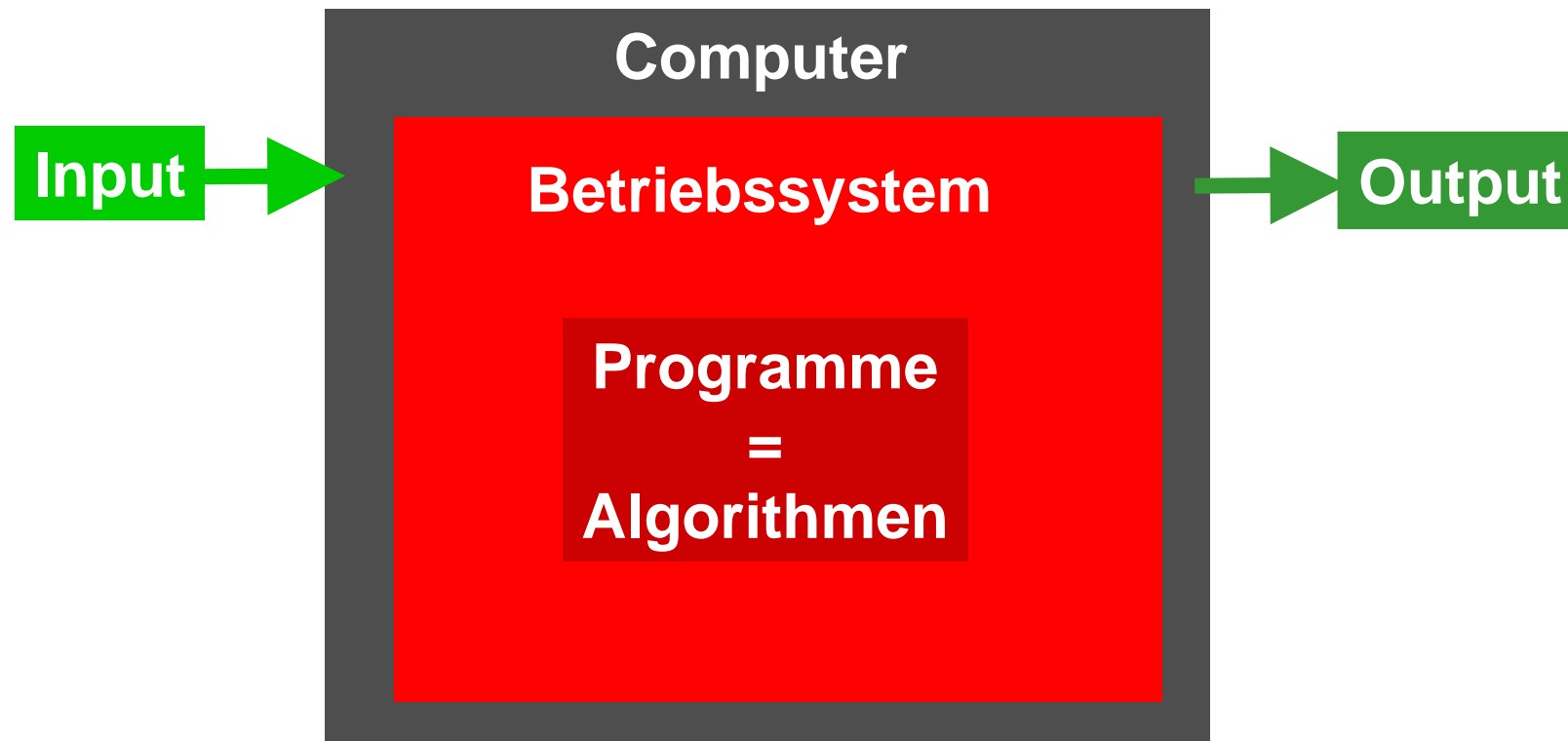
Vorlesung an der Charité - Universitätsmedizin Berlin

- Allgemeine Informatik
 - Computer
 - Programmiersprachen
 - (Neuronale Netzwerke)
- Beispiele aus der Biologie
 - (DNA-Transkription)
 - (Proteinsynthese)
- Anwendungen aus der Medizin
 - AD-Wandler
 - Robotik
 - Radiologie

Wenn Sie diese Vorlesung absolviert haben, dann kennen Sie:

- grundlegende Konzepte der Programmierung wie Input/Output und Schleifenstrukturen,
- (die Grundidee der Programmierung neuronaler Netze),
- Besonderheiten und Vorteile der Verarbeitung digitalisierter Analogdaten,
- die Vielfalt der Einsatzmöglichkeiten der medizinischen Informatik in der Klinik.

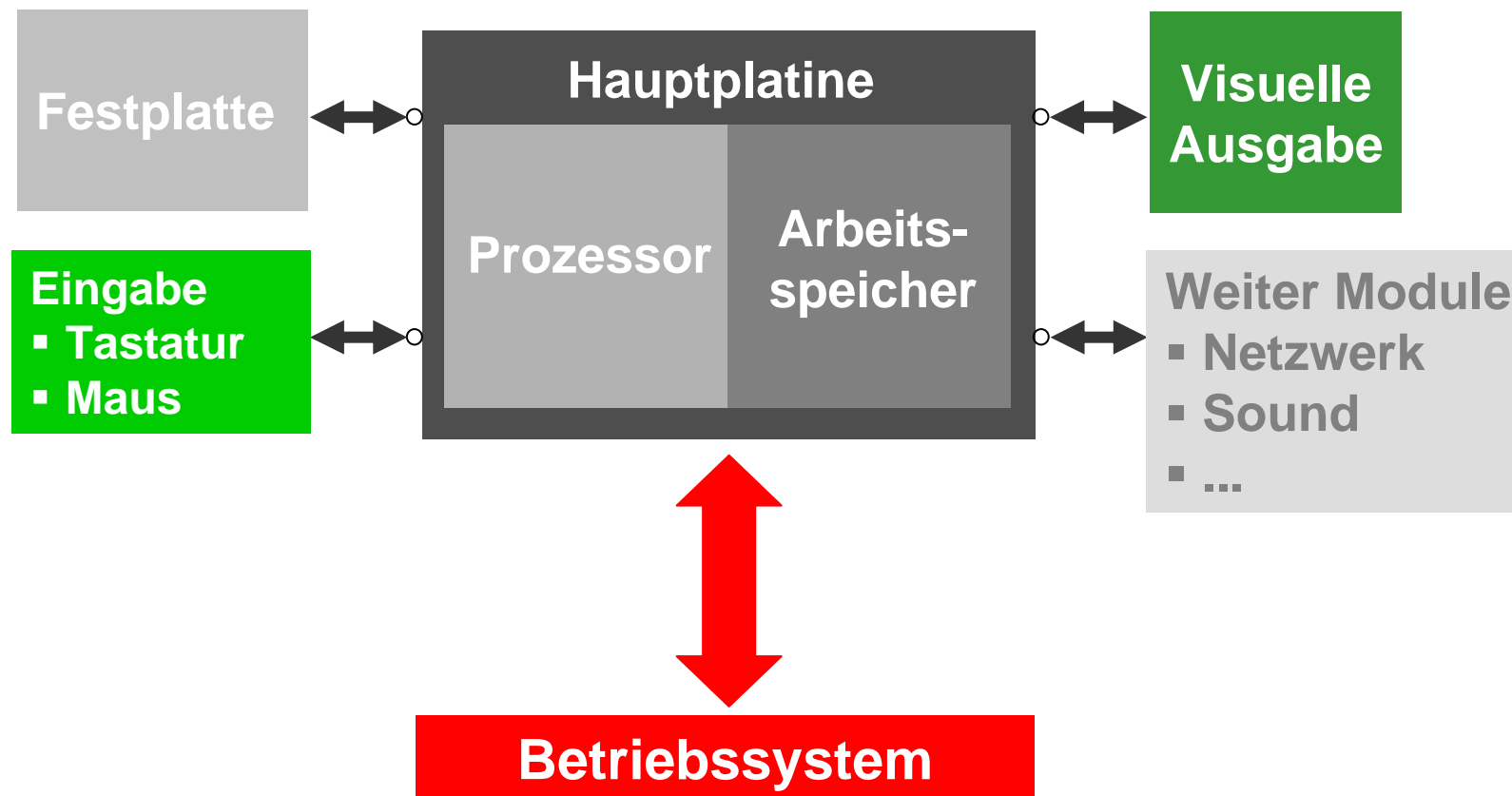
Computer = „Input – Output – Maschine“



Komponenten eines Computer

5

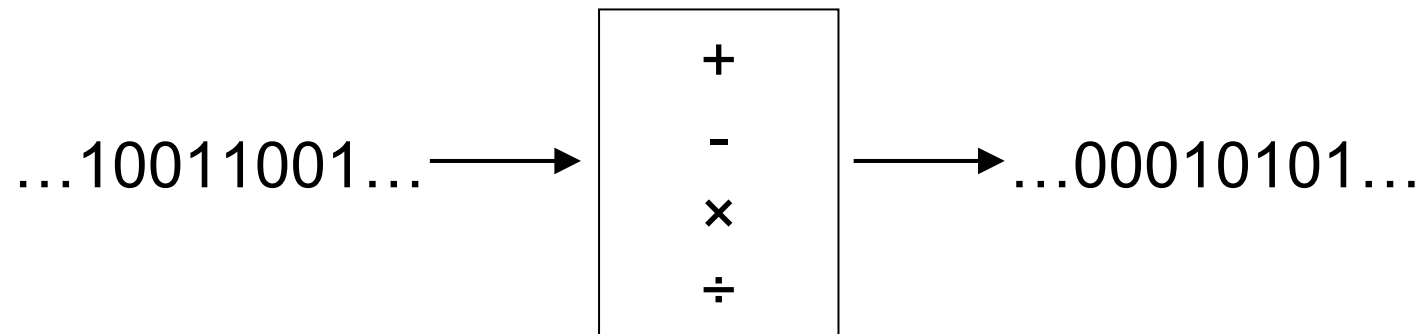
Allgemeine Informatik



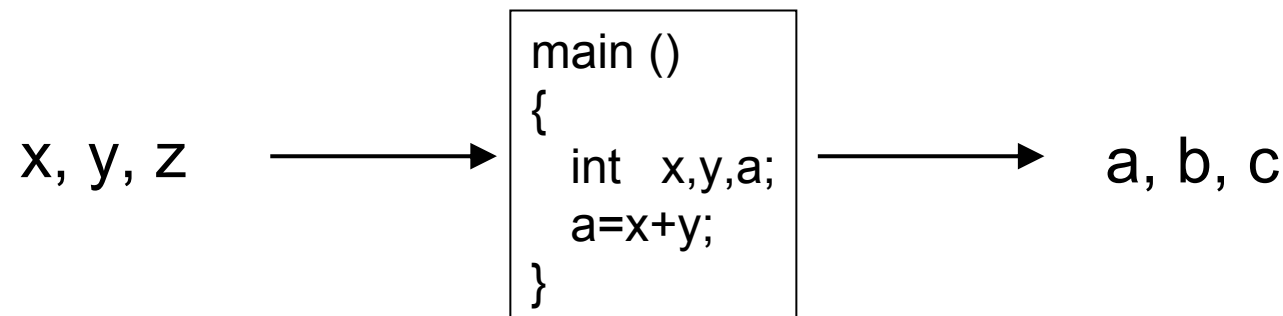
Intel / IBM	80x86	Windows / Linux
Mac	68000	MacOS
Sun	Sparc	UNIX
DEC	Alpha	VMS
SGI	MIPS	IRIX

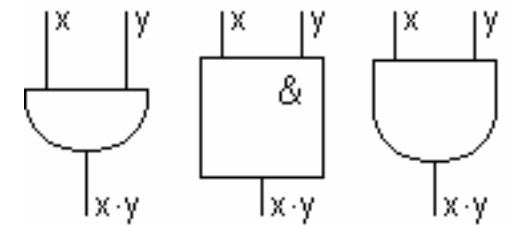
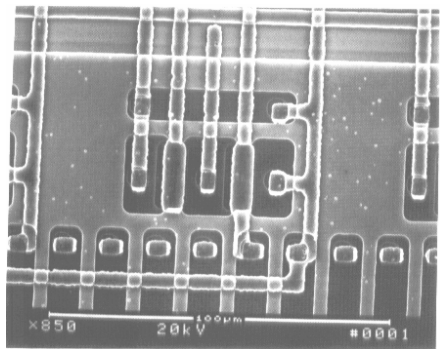
Einige medizintechnische Geräte wie EKG oder Laboranalysegeräte haben oft eigene speziell programmierte kleinere Prozessoren mit spezifischem Bedienmenü.

Binäre Input/Output-Daten und Operationen auf ihnen

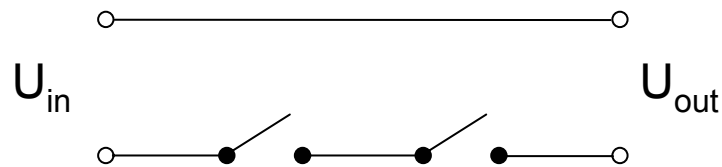


werden in Variablen und Programmen in Textform dargestellt.

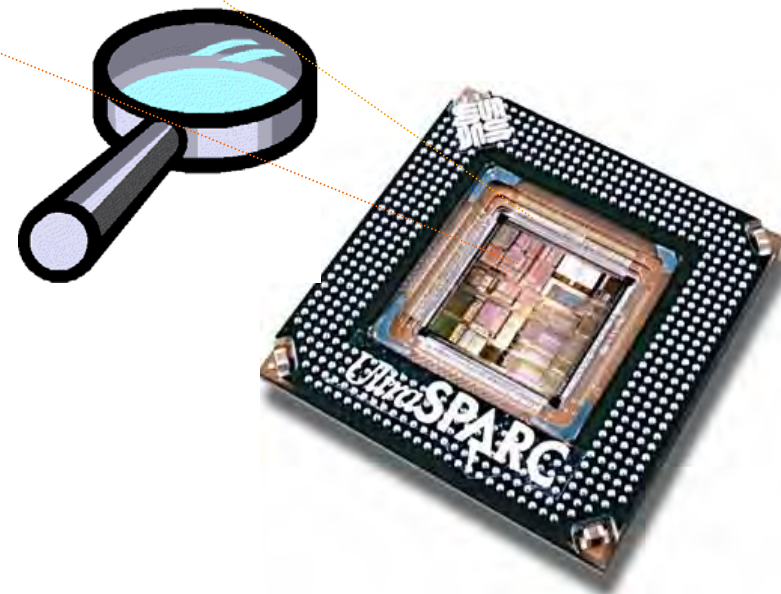




Schallsymbol AND-Verküpfung



Schaltung AND-Verküpfung



Quelle: University of the Negev

Schaltlogik: AND-Verkupfung

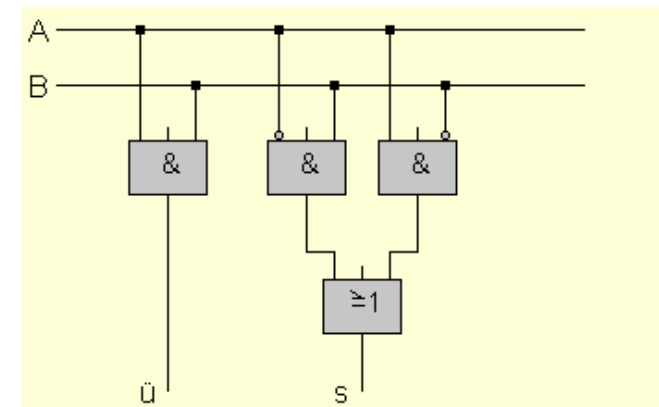
x	y	x AND y
0	0	0
0	1	0
1	0	0
1	1	1

AND
OR
XOR
NOT
NAND
NOR
XNOR

} Elektronisch realisierbare Schaltungen

Beispiel: Addition zweier Dualzahlen (0 oder 1):

A	B	1. Stelle	0. Stelle
1	1	1	0
0	1	0	1
1	0	0	1
0	0	0	0



Höhere Programmiersprache: Quellcode in beliebigem Editor



Compiler: übersetzt Quellcode in Assembler



Assembler: übersetzt Prozessor-Befehlssatz in Maschinencode



Chip-Befehlssatz: Maschinentersprache: 0110 1110 ...

Eine kurze Geschichte:

- **FORTRAN** (*FORmula TRANslation*)
1956, John Backus, IBM Corp.
 - für wissenschaftliche und technische Anwendungen
- **COBOL** (*COmmon Business Oriented Language*),
1959, Grace Hopper et al., U.S. Navy
 - für wirtschaftliche Anwendungen
- **BASIC** (*Beginner's All-Purpose Symbolic Instruction Code*)
1963, Mathematiker J. G. Kemeny und T. Kurtzas,
Dartmouth College
 - wichtigste Sprache der PC-Revolution

Eine kurze Geschichte:

- **Pascal** (für den französischen Mathematiker Blaise Pascal)
1971, Prof. Nicolas Wirth, Schweiz
 - entwickelt als eine besser strukturierte Programmierumgebung
- **C**
1972, Dennis Ritchie, Bell Laboratories
 - entwickelt für UNIX, erweitert auf fast alle anderen Betriebssysteme

1. for

Beispiel: Berechnung der Fakultät: $N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot N$

```
fak = 1;  
for (i=1;i<N;i++) {  
    fak = fak *i;  
}
```

2. while

Beispiel: Berechnung des Sinus mit der Taylor-Reihe

$$\sin(\varphi) = \sum_{n=0}^{\infty} (-1)^n \frac{\varphi^{2n+1}}{(2n+1)!}$$

```
phi = 17;  
genauigkeit = 0.001;  
sinphi = 0;  
while (epsilon > genauigkeit) {  
    tmp = sinphi;  
    sinphi = sinphi + (-1)^n*phi^(2*n+1)/fak(2*n+1);  
    epsilon = sinphi - tmp;  
    ++n  
}
```

1. if

Beispiel: Überprüfung der Eingaben in eine Datenbank

```
if (Patient.Alter>120) {  
    printf("Patientenalter wahrscheinlich zu hoch")  
}
```

2. switch

Beispiel: Auswahl von Menüpunkten

```
scanf("%s", Auswahl);  
switch (Auswahl) {  
    case MRT:  
        strcpy(Path,"C://MRT");  
    case CT:  
        strcpy(Path,"C://CT");  
    case US:  
        strcpy(Path,"C://US");  
}
```

Objektorientierte Sprachen:

- **C++**
ab 1979, Bjarne Stroustrup, Bell Laboratories
 - objektorientierte Erweiterung von C
- **Java**
1991-1992, P. Naughton, M. Sheridan, J. Gosling, Sun Microsystems
 - Netzwerkorientierte, plattformunabhängige Sprache
- **Delphi**
??, Borland
 - objektorientiertes Pascal

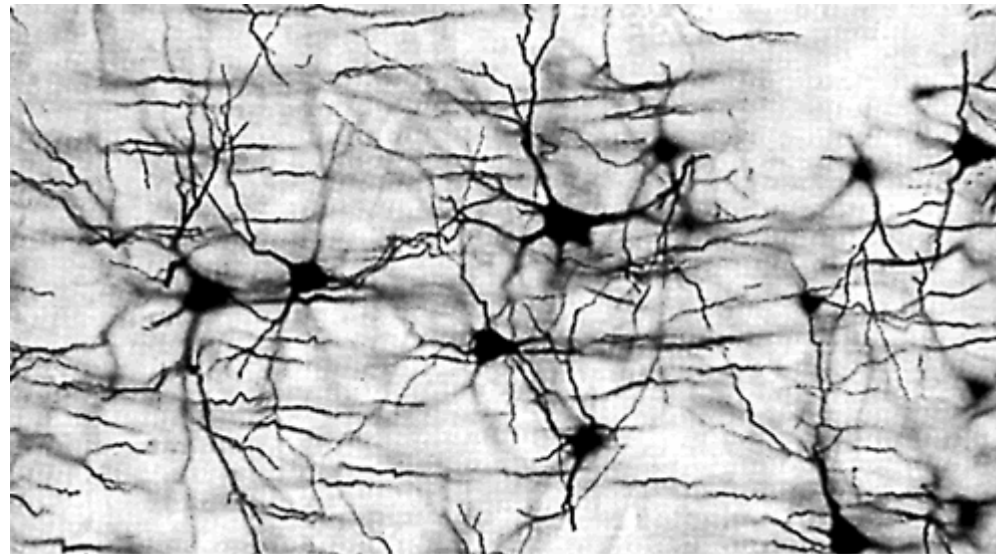
Definition einer Klasse:

```
class point {  
    private:  
        double x;  
        double y;  
    public:  
        void set_x (double);  
        void set_y (double);  
        double get_x ();  
        double get_y ();  
};
```

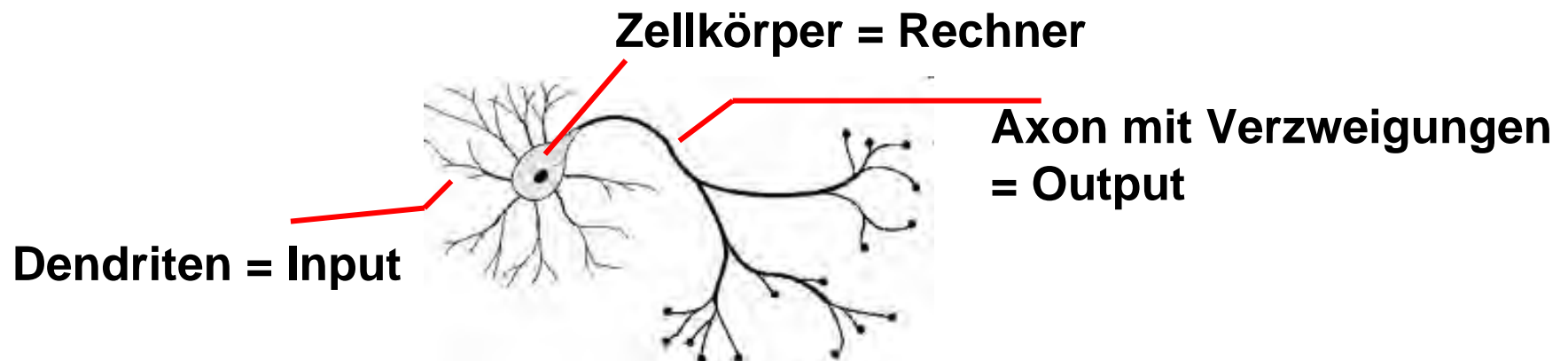
Aufruf einer Klasse:

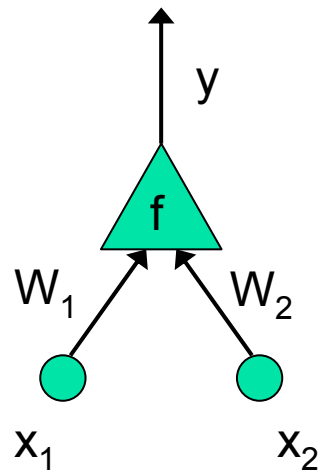
```
main () {  
    point center;  
    center.set_x (4.);  
    center.set_y (7.);  
};
```

- Matlab® Mathematische Berechnungen und Visualisierungen, insbesondere für Matrixrechnung
- Maple® Höhere Algebra, d.h.
 - Gleichungssysteme
 - Integrale
 - Differentialgleichungen...
nicht numerisch, sondern *symbolisch!*
- SPSS® (Statistical Package for the Social Sciences)
 - Statistische Auswertung
 - und Visualisierung→ Übungen zur Biomathematik für Mediziner



Quelle: alanturing.net





$x_1, x_2 =$ inputs

$W_1, W_2 =$ Wichtungen

$f =$ Aktivierungsfunktion

$y =$ output

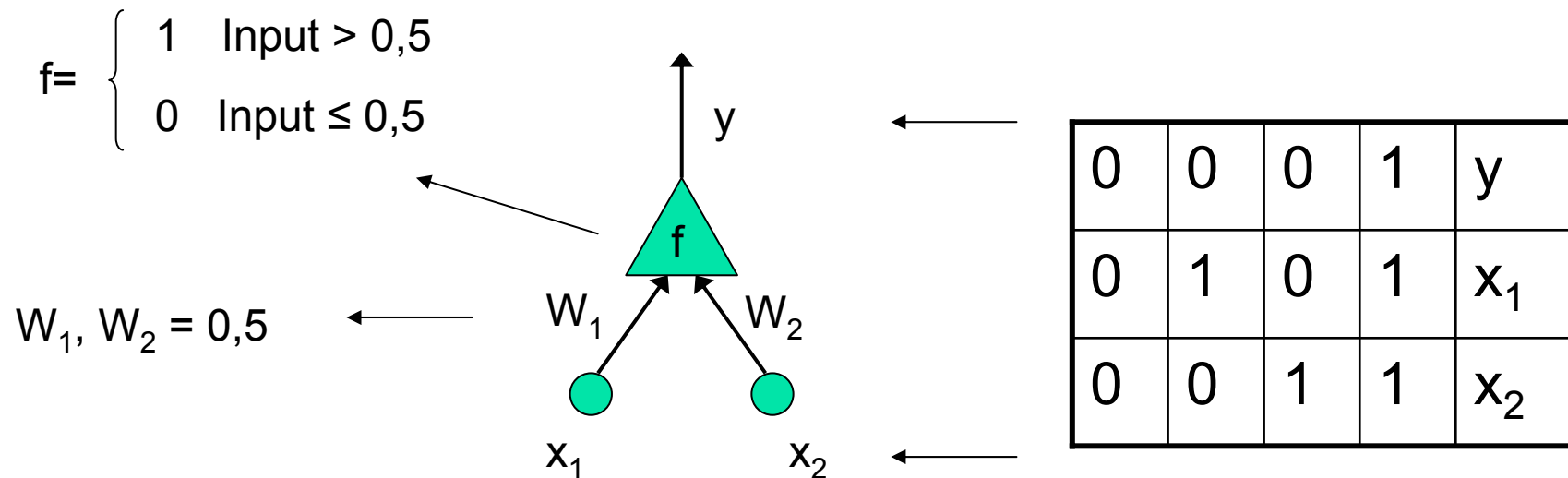
$$y = (W_1 x_1 + W_2 x_2) \cdot f(W_1 x_1 + W_2 x_2)$$

Die Inputs werden gewichtet und erzeugen je nach Zustand f der ‚Zelle‘ einen Output.

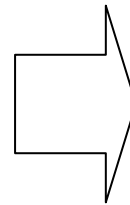
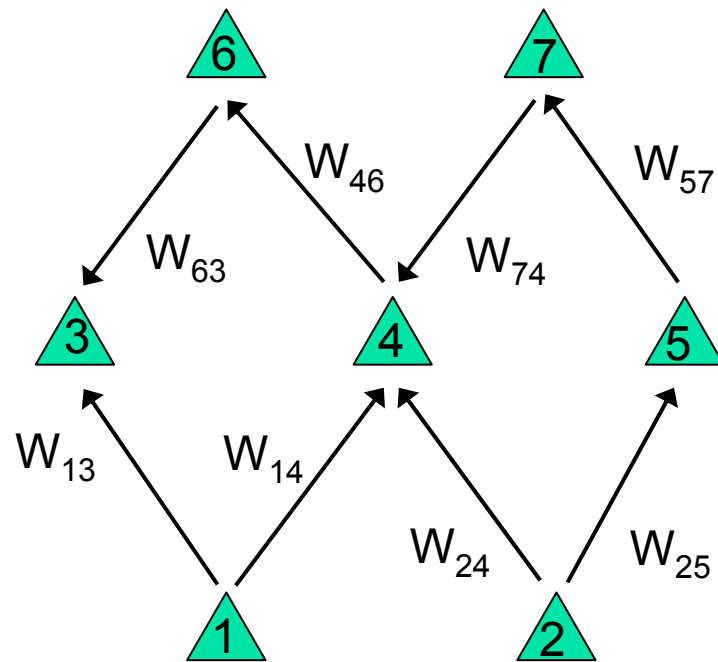
Das künstliche Axon lernt!

20

Allgemeine Informatik



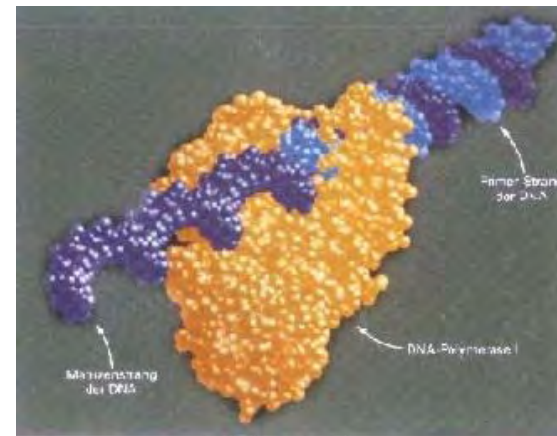
Durch feste Vorgabe von gewünschten Ergebnissen für bestimmte Eingaben kann das System seine Parameter iterativ anpassen, es lernt.



$$\begin{pmatrix} 0 & 0 & W_{13} & W_{14} & 0 & 0 & 0 \\ 0 & 0 & 0 & W_{24} & W_{25} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & W_{46} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & W_{57} \\ 0 & 0 & W_{63} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & W_{74} & 0 & 0 & 0 \end{pmatrix}$$

Netze künstlicher Axone führen zu linearen Gleichungssystemen in Matrixform.

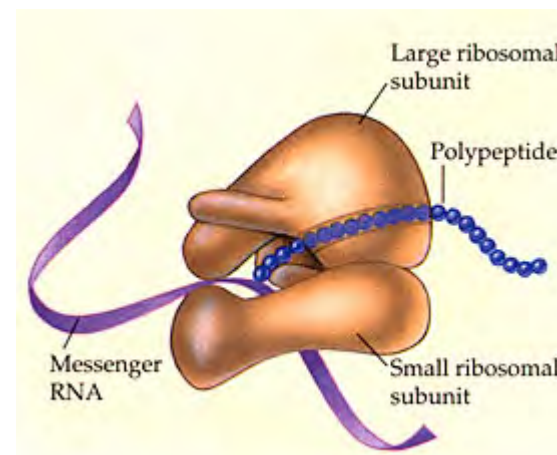
- DNA Replikation



DNA-Replikation

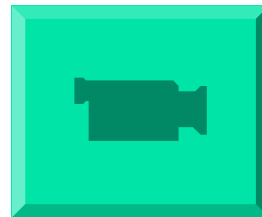
Quelle: Universität Wien

- Proteinsynthese



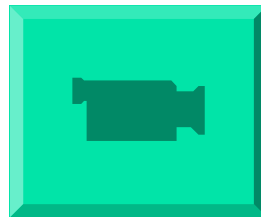
Quelle: Bergen County Technical Schools

- Replikation parallel für beide Stränge



- Replikation parallel für beide Stränge
- Replikation abschnittsweise
 - kein lineares Programm, sondern Schleifenstruktur
 - hochgradig parallele Replikation
 - Anwendung: PCR (Polymerase Chain Reaction)

Beispiele aus der Biologie



Quelle: State University of West Georgia

- Jeweils 3 Basen kodieren eine Aminosäure
 - Für 23 Aminosäuren gibt es $3^4 = 81$ Codes ($2^4 = 16$ zu wenig)
- Sequentielle Verknüpfung des Peptids am Ribosom

!!Schon eine fehlerhafte (mutierte) Base kann für eine falsche Aminosäure kodieren

- AD-Wandler

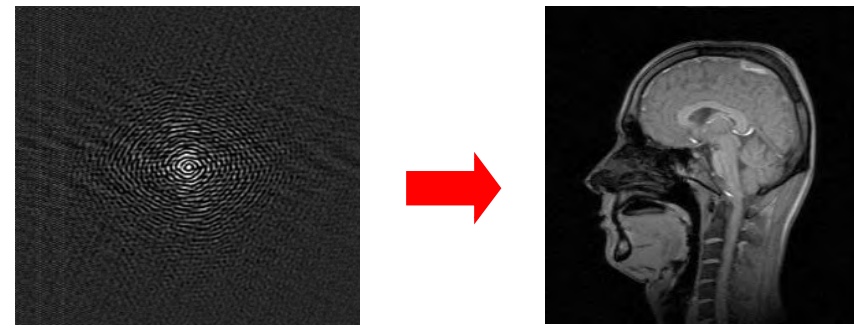


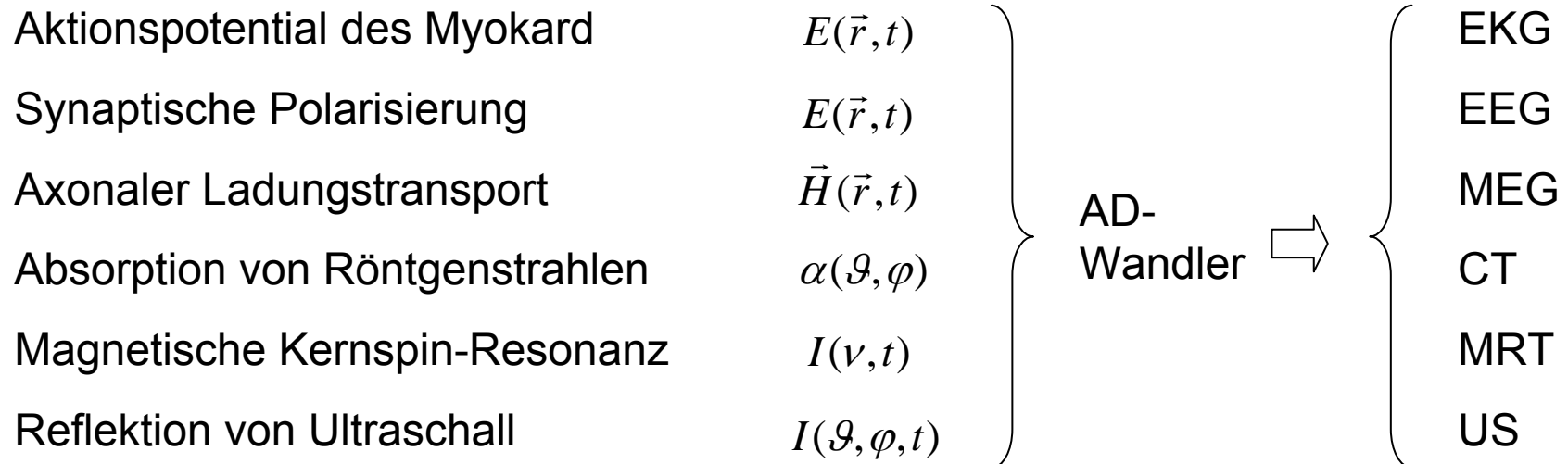
- Robotik: OP-Maschinen



CASPAR

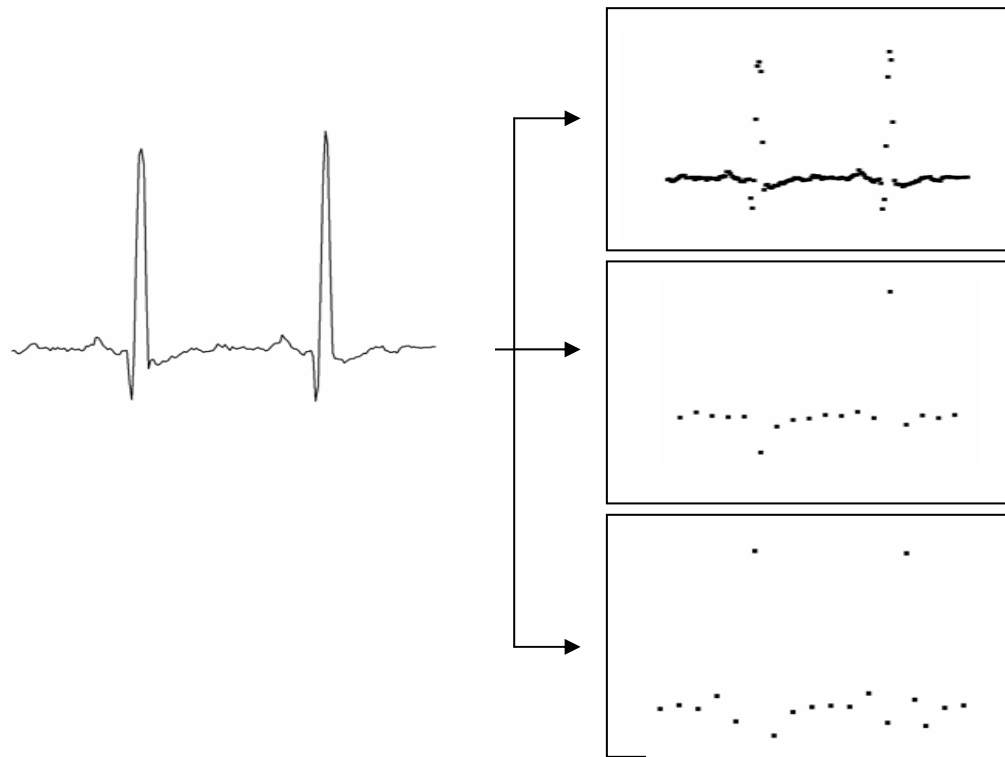
- Radiologie:
Aufnahme digitaler
Bilder





Durch das Digitalisieren der physikalischen Meßdaten werden die Signale für den Computer lesbar und können weiterverarbeitet werden.

Beispiel: EKG

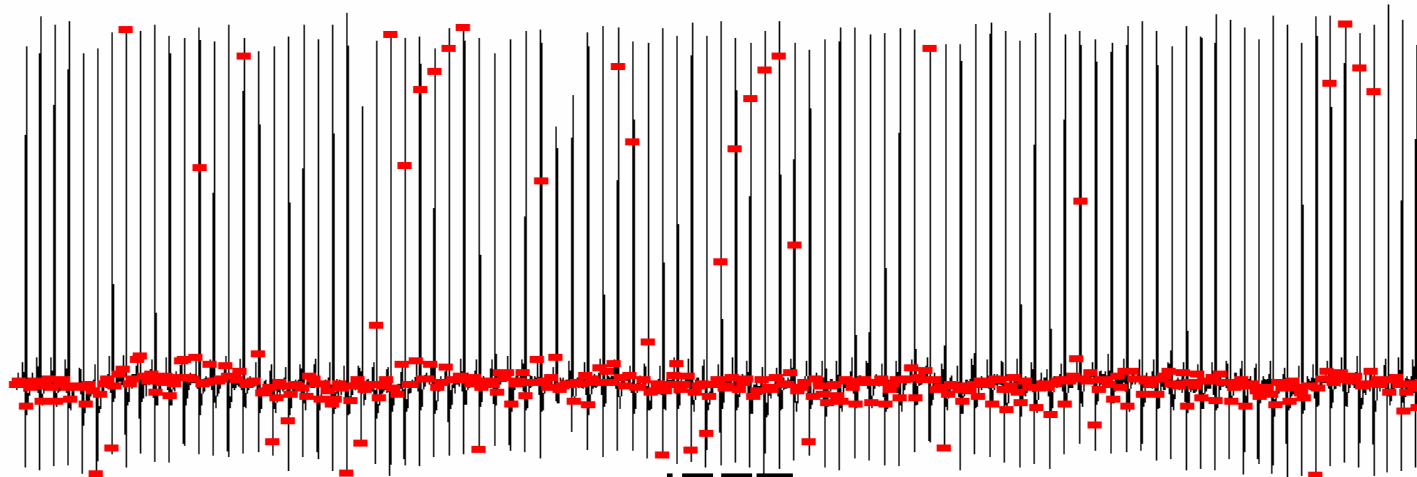


Sampling	
Rate	Zeit
125 Hz	8 ms
12,5 Hz	8 ms
12,5 Hz	80 ms

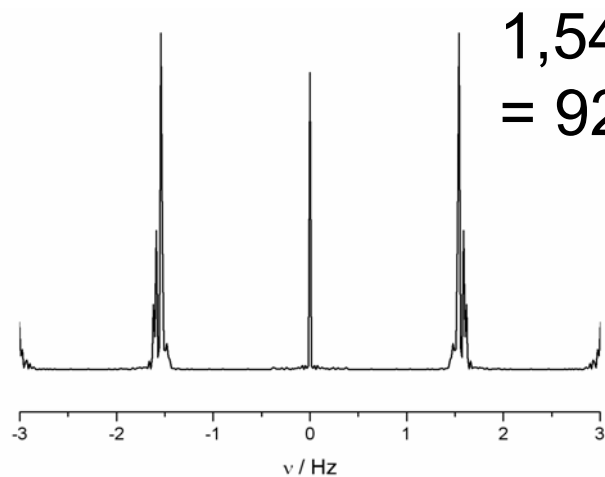
Automatisches Auslesen der Herzfrequenz

30

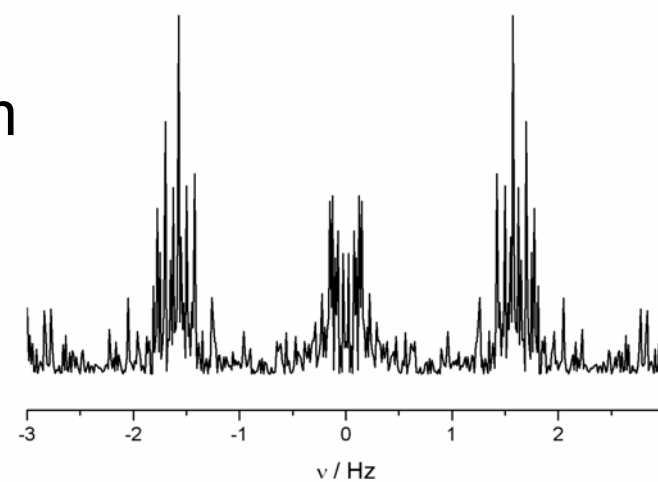
Anwendungen in der Medizin



iFFT



1,54 Hz
= 92 bpm



Durch digitale Patientendaten-Erfassung

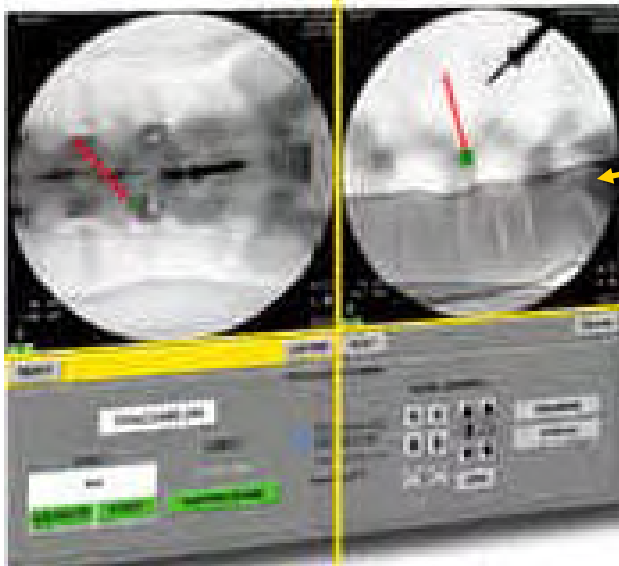
- Automatische Überwachung von
 - Herzfrequenz
 - Atmungsfrequenz
 - Körpertemperatur

Mit Schwellwerten

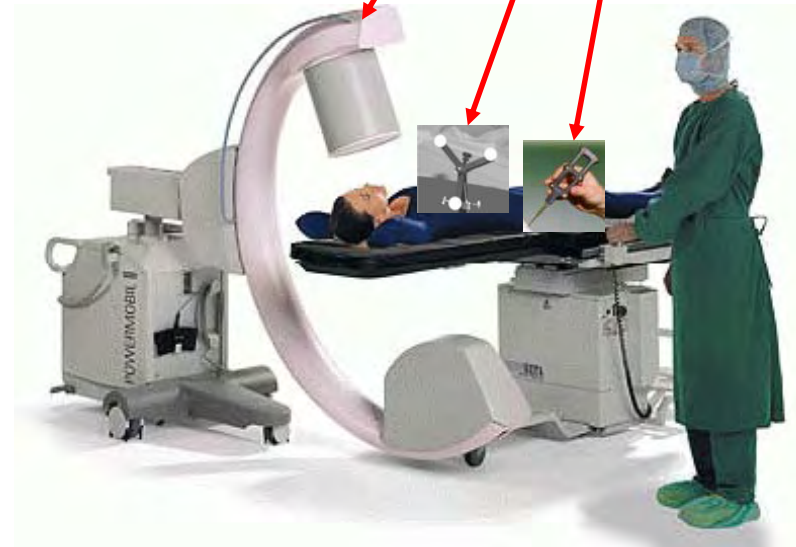
- Automatisierte Warnsignale
- Deutliche Erleichterung in der Intensivpflege

Robotik (1): Tracker-Systeme

Anwendungen in der Medizin



Marker am C-Arm,
Patient und Werkzeug

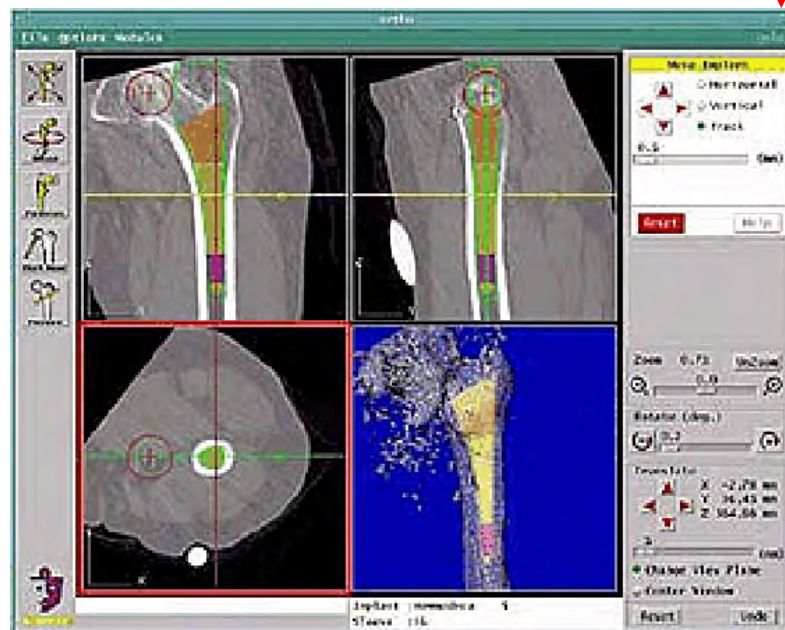


Einmalige fluoroskopische Aufnahme, durch Tracker real-time auf aktuelle Positionen registriert.

Abbildungen: Johns Hopkins University

OP-Planung für Hüftgelenkprothese mit CT Daten.

- Die Prothesenhals wird anatomisch durch Segmentierung optimal in den virtuellen Patienten eingepaßt.



Quelle: umin.ac.jp

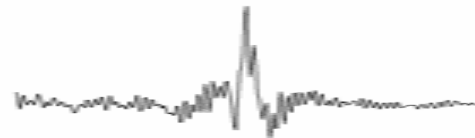


Quelle: Berufsgenossenschaftliche Unfallklinik, Frankfurt / M.

OP: Robodoc fräst selbständig den Kanal für die Prothese in Oberschenkelhals und Hüftknochen.

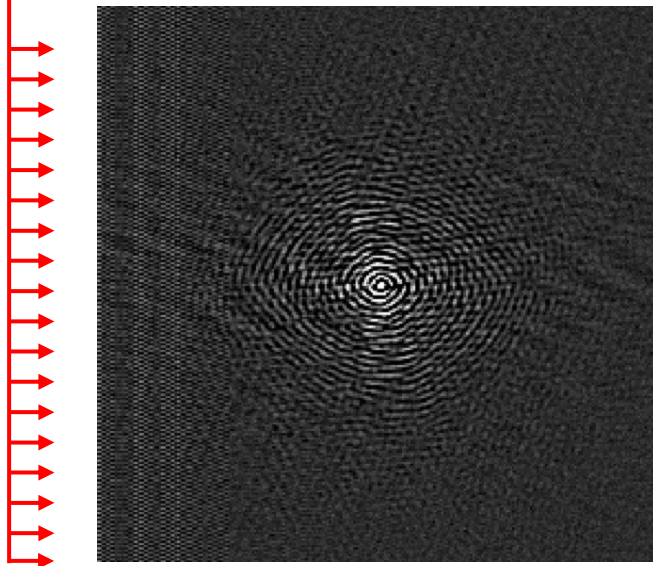
- Auch hier kommt ein Tracking-System zum Einsatz.

1. Schritt:	Signalerzeugung	MR: Kernspinresonanz CT: Röntgenabsorption
2. Schritt:	Datenerfassung	MR: Induktionsspule CT: Photomultiplier
3. Schritt:	AD-Wandler (für beide Methoden)	
4. Schritt:	Bildrekonstruktion	MR: Fourier-Transformation CT: Radon-Transformation



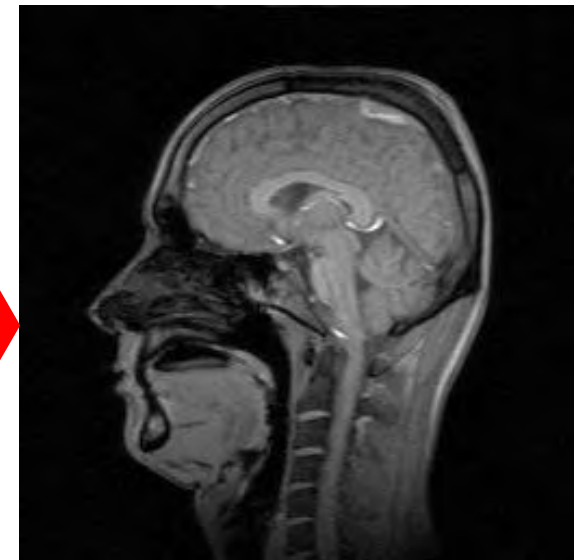
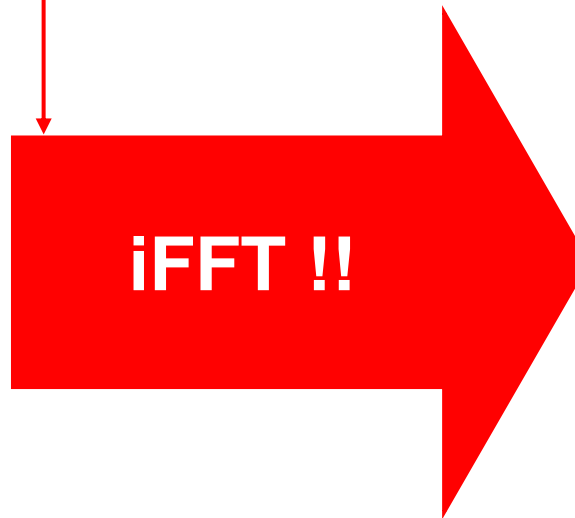
Primärecho

256 Primärechos



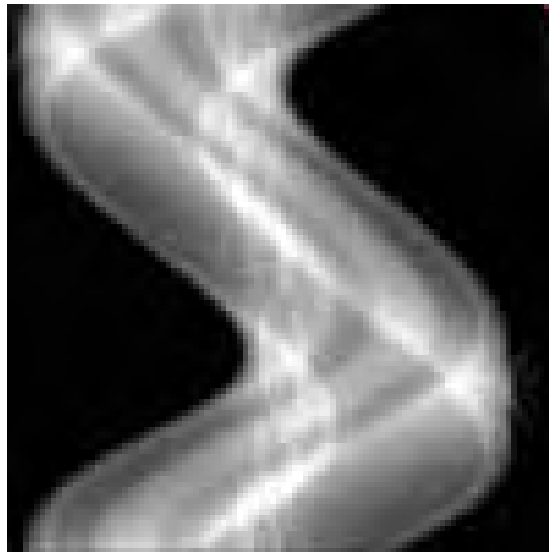
k-Raum

iFFT = inverse Fast Fourier Transform
→ Algorithmisch anspruchsvolle Operation auf 2D-Datensätzen mit 256×256 (=65536) Pixeln.



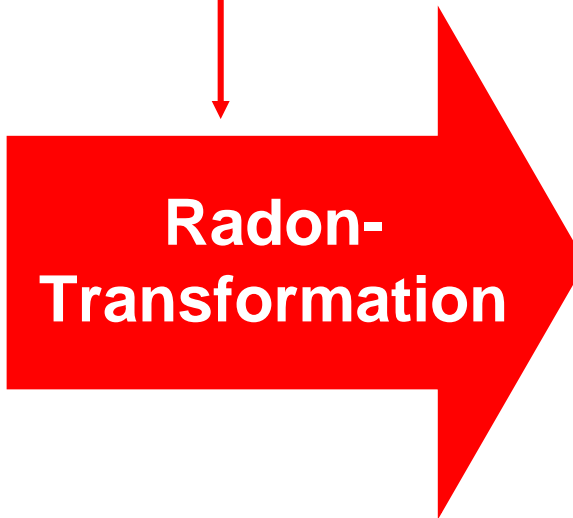
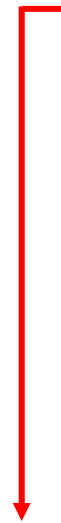
r-Raum

500 – 1000 Projektionen
des Objektes mit je
500 – 1000 Datenpunkten



CT-Rohdaten

Oder: gefilterte Rückprojektions-
transformation.
Wegen der großen Datenmenge
numerisch anspruchsvoll.



- Für die Informationsverarbeitung in der Medizin gelten dieselben Grundlagen wie allgemein in der Informatik
- Biologische Systeme wie die DNA und die neuronale Struktur des Gehirns können Pate stehen für neue Entwicklungen
- Digitale Signalverarbeitung weist einige Besonderheiten auf, die man bei der Interpretation der gewonnenen Daten berücksichtigen muß
- Durch schnelle Rechner und Algorithmen werden bestimmte bildgebende Verfahren erst möglich