Haptische Interfaces

- Motivation
- Physiologie

- Sensoren zur Orts- und Kraftmessung
- Verfahren zur Kraftrückmeldung
- Gerätespektrum

- Datenhandschuh
- Head- / Eye-Tracking
- VR-Anwendungen

- DirectInput
- Codebeispiel

Haptik: Literatur

Thema "haptische / multimodale Interfaces" ist recht neu...

... bisher keine geeigneten Lehrbücher erschienen

Konferenzbände:

Int. Workshop on haptic human-computer interaction, Glasgow, 2000, Springer

Int. Conf. on Cooperative multimodal communication, CMC/95, Eindhoven, Springer

http://haptic.mech.nwu.edu/intro/gallery/

www.immersion.com

www.sensable.com

www.logitech.com / www.microsoft.com/hardware

Bargen, Donelly, Inside DirectX, Microsoft Press, 98

Schäpers, DirectX nicht nur für Spiele, c't 9/99-216, 12/99-238, 15/99-180

Haptik: Motivation

- verbesserte Mensch-Computer Interaktion
- durch Ausnutzen des Tastsinns ("full-duplex")
- Unterstützung von behinderten (blinden) Personen

- direkte Manipulation von Objekten in CAD-Systemen
- auch für unsichtbare / verdeckte Objekte
- Erkennen von Objekten über (simulierte) Oberflächen und Reibung

- verbessertes "Eintauchen" in VR-Umgebungen
- insb. Massenmarkt 3D-Actionspiele

aber:

- Wahrnehmung von Force-Feedback bisher wenig erforscht
- Sicherheitsmaßnahmen notwendig

Haptik: Status

- aktuelles Forschungsthema
- Entwurf und Herstellung von Sensoren und Aktoren
- Systemintegration, Programmierschnittstellen

- breites Gerätespektrum, bisher weitgehend mechanisch:

	Freiheitsgrade / Parameter
Lenkräder	1
Joystick	2 .. 3
Stifte, Roboterarme	5 .. 6
Datenhandschuh	6 .. 20
"smart skin"	1000+

- aktuelle Spiele unterstützen zunehmend FF-Geräte
- billigere Sensoren durch Mikrosystemtechnik

Multimodale Interfaces

"multimodal" := Kombination mehrerer Modalitäten (=Sinne)

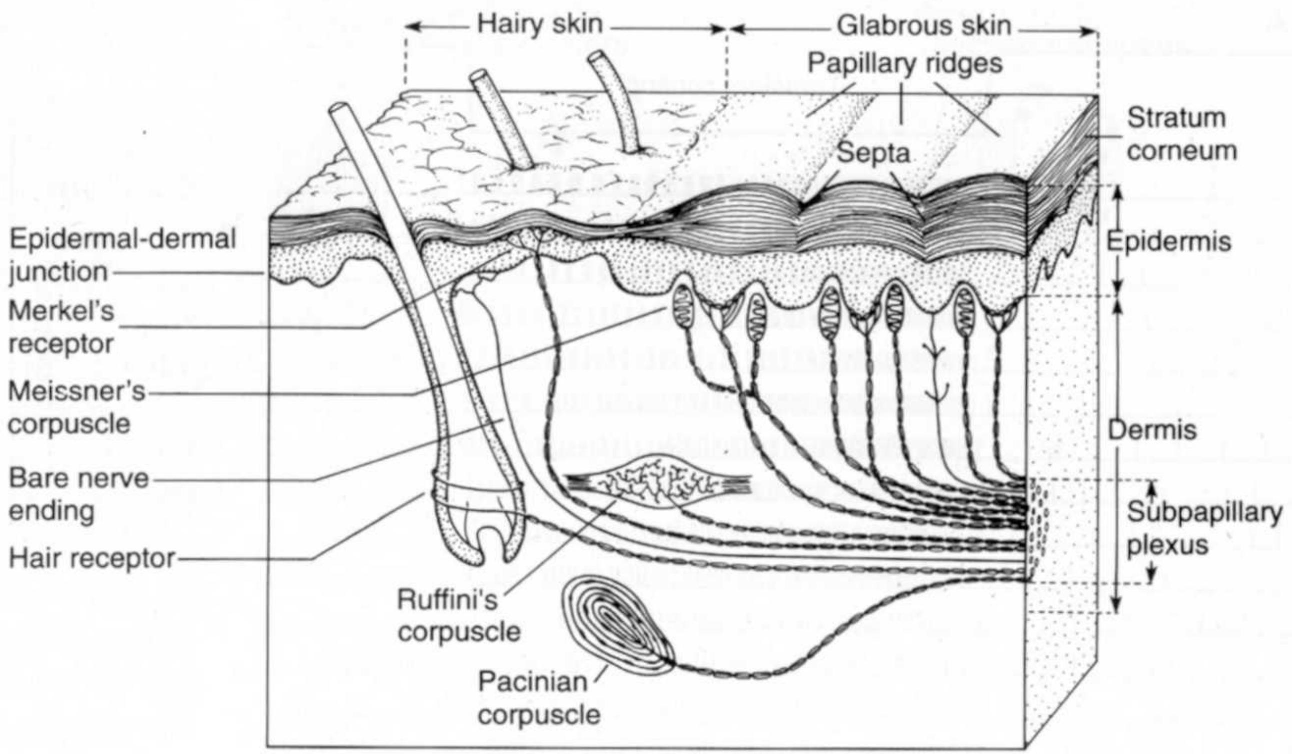
- Maus- / Tastatur- / Joystick- / Datenhandschuh
- Spracheingabe
- Eye-Tracking / Gestenerkennung

- vielfältige neue Möglichkeiten zur Rechnersteuerung
- aktuelles Forschungsthema
- bisher keine einheitliche Theorie

(vgl. Ausschreibung TAMS: "technische Aspekte multimodaler Systeme" am FBI)

Tastkörperchen

FIGURE 24–2
The location of various receptors in hairy and hairless (glabrous) skin of primates. Receptors are located in the superficial skin, at the junction of the dermis and epidermis, and more deeply in the dermis and in subcutaneous tissue. The receptors of the glabrous skin are: Meissner's corpuscles, located in the dermal papillae, Merkel's receptors, also located in the dermal papillae, and bare nerve endings. The receptors of the hairy skin are: hair receptors, Merkel's receptors (having a slightly different organization than their counterparts in the glabrous skin), and bare nerve endings. Subcutaneous receptors, beneath both glabrous and hairy skin, include pacinian and Ruffini's corpuscles. (Adapted from Light and Perl, 1984.)



- mehrere Sorten, in der Haut, Haarwurzeln, ...
- unterschiedliche Empfindlichkeiten
- unterschiedliche Wahrnehmung (lokalisiert, großflächig, ...)

(Kandell, Principles of Neural Science)

Tastsinn: Auflösung

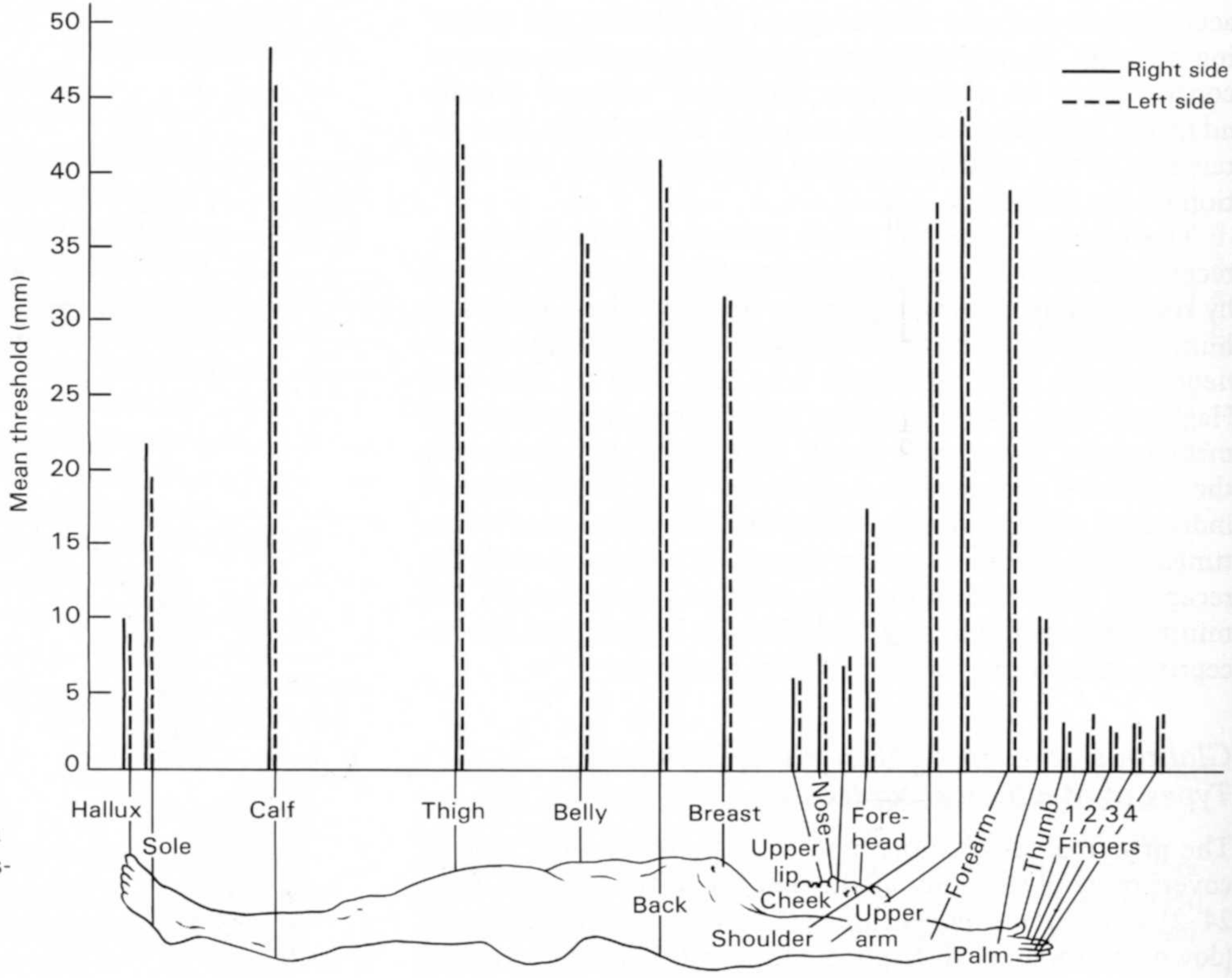
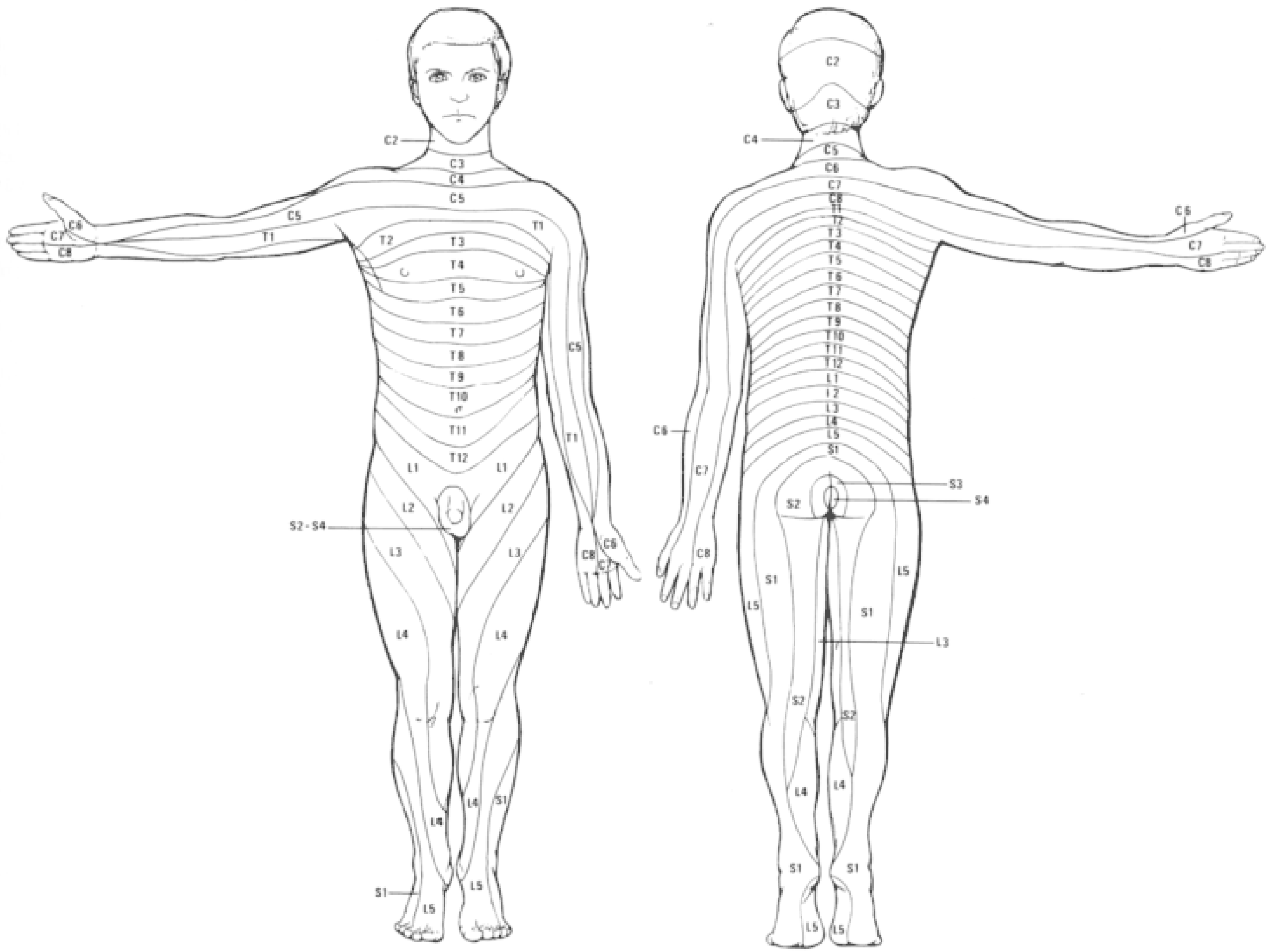


FIGURE 24–7
Two-point discrimination varies with location on body surface. Greatest discriminative capacity is present in the finger tips, lips, and tongue. (Adapted from Weinstein, 1968.)

(Kandell)

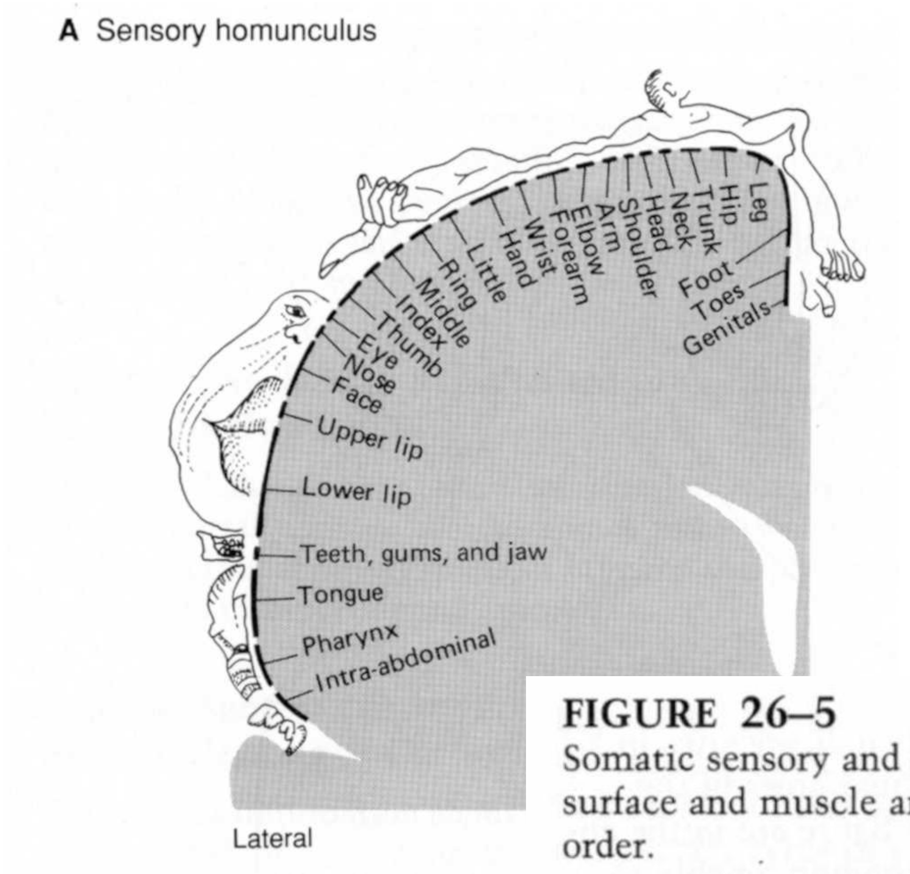
Tastsinn: Nervenbahnen

FIGURE 25–2
The dermatomes follow a highly regular pattern on the body (**S**, sacral; **L**, lumbar; **T**, thoracic, **C**, cervical). In actuality, the boundaries of the dermatomes are less distinct than shown here because of overlapping innervation.



(Kendall)

Tastsinn: Wahrnehmung



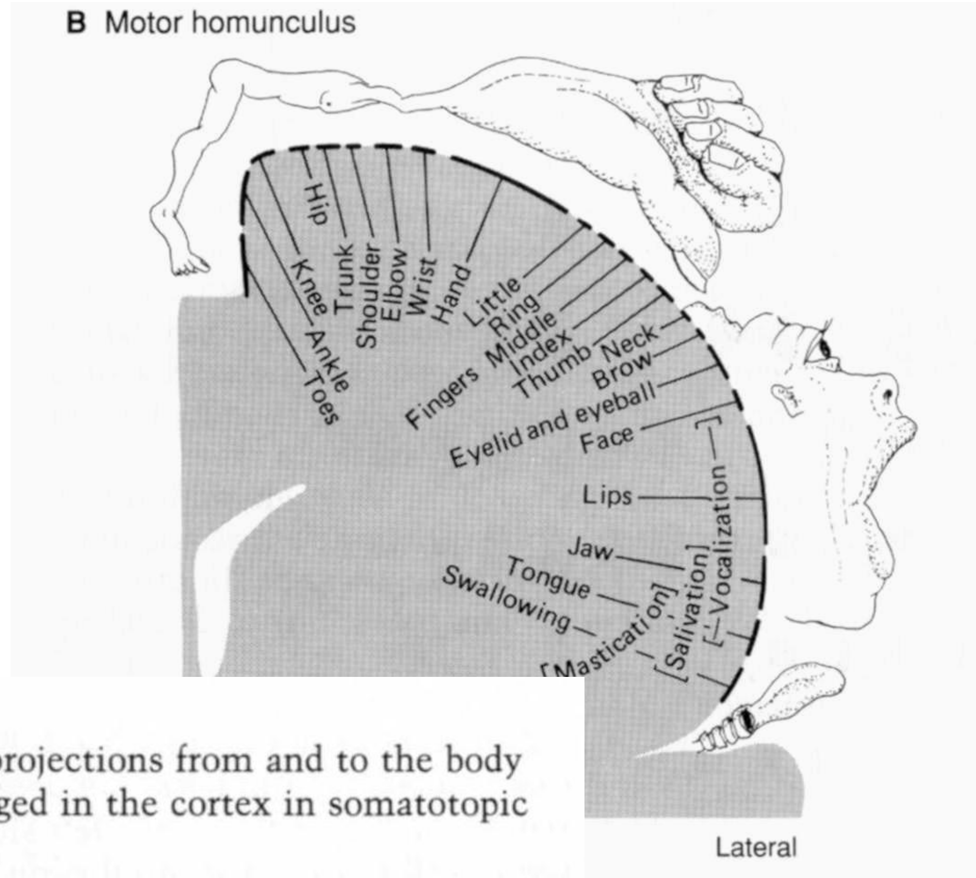


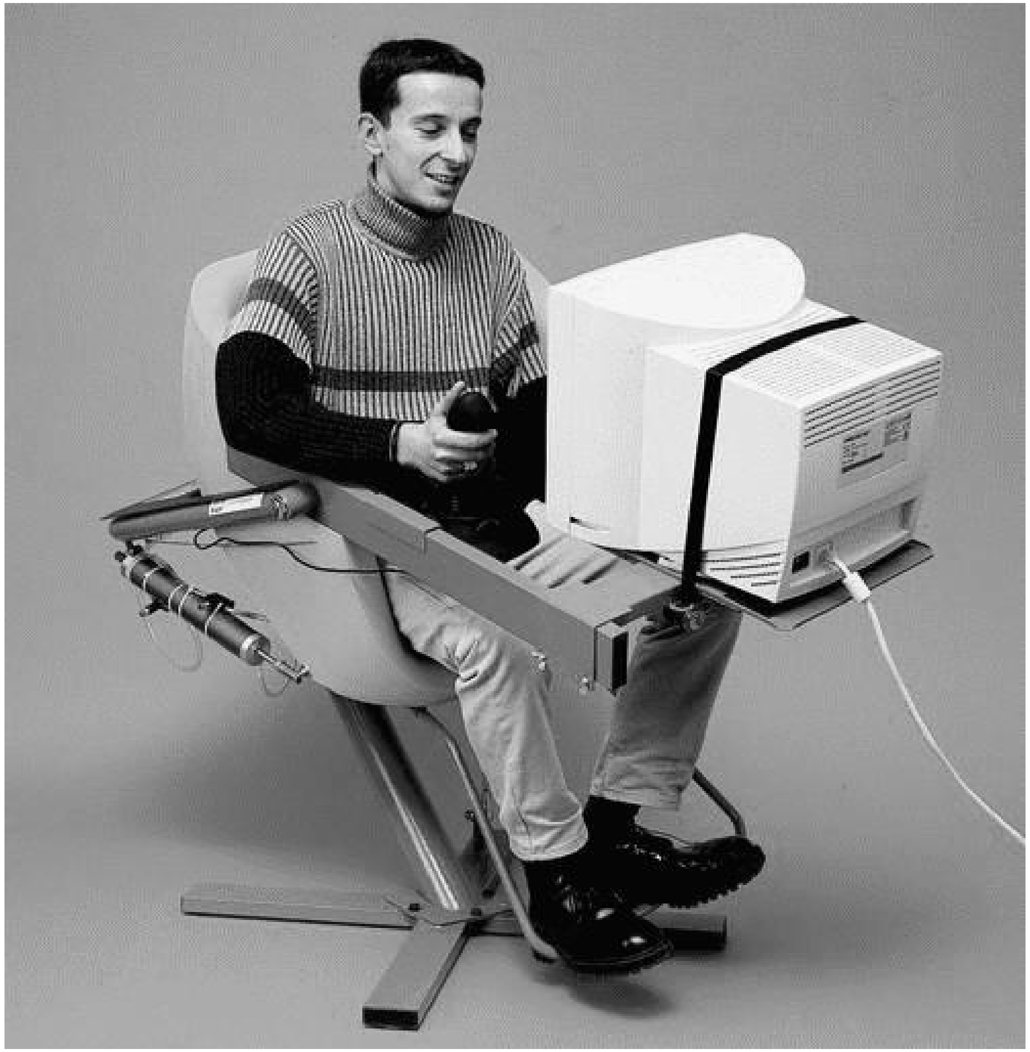
FIGURE 26–5
Somatic sensory and motor projections from and to the body surface and muscle are arranged in the cortex in somatotopic order.

A. Sensory information from the body surface is received by the postcentral gyrus of the parietal cortex (areas 3a and 3b, and 1 and 2). Here the map for area 1 is illustrated. Areas of the body that are important for tactile discrimination, such as the tip of the tongue, the fingers, and the hand, have a disproportionately larger representation, reflecting their more extensive innervation. (Adapted from Penfield and Rasmussen, 1950.)

B. The analogous motor map exists for the motor cortex.

(Kandell)

Lage- / Bewegungswahrnehmung:



Lagewahrnehmung:

- Beschleunigungsmessung
- "Bogengänge" im Innenohr
- nicht von außen zugänglich

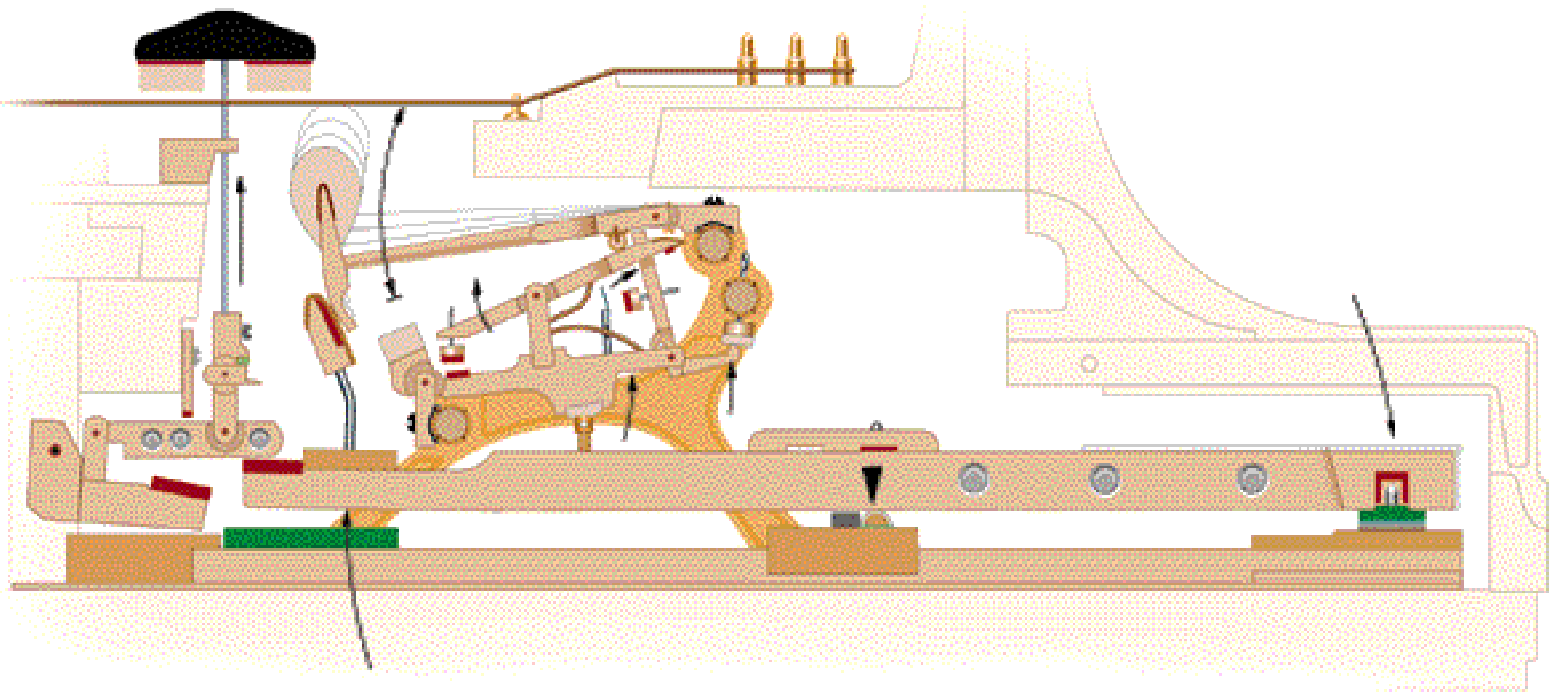
Simulation / VR:

- Bewegung des Körpers nötig
- aufwendige Mechanik, z.B.
- 6-Achsen Hydraulik für prof. Fahr-/Flugsimulatoren
- Rock'n'Ride: derzeit einzige low-cost Alternative :-)

(c't 26/99 116t)

Haptic Interface: Beispiel

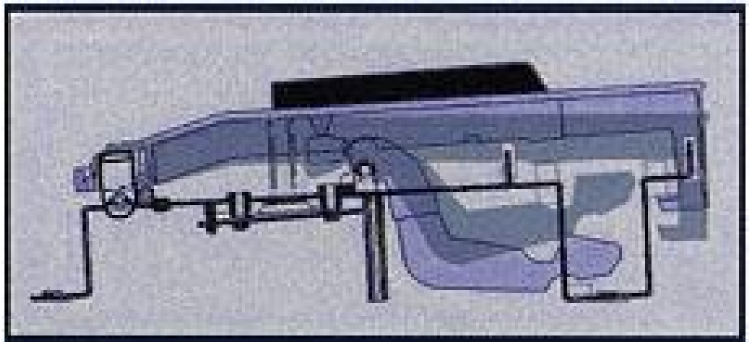
(Scientific American 01/99, Steinway&Sons)



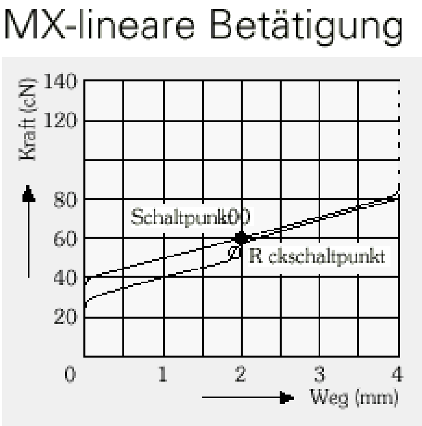
Kunden bezahlen durchaus für gutes "Spielgefühl" :-)

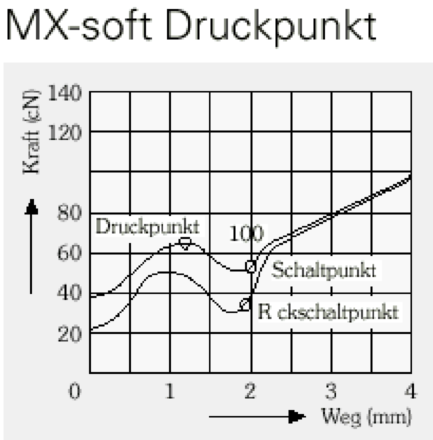
zum Vergleich:

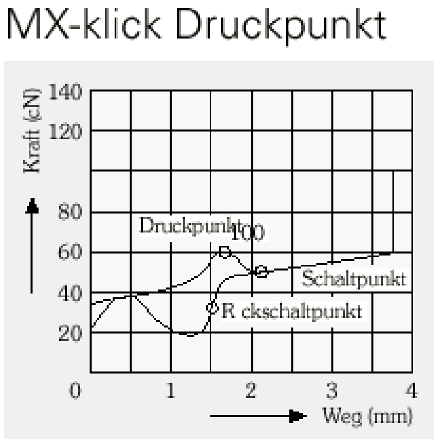
- Mechanik des Alesis QS8 Synthesizers
- diverse Kennlinien umschaltbar
- einstellbare Empfindlichkeit



Tastatur



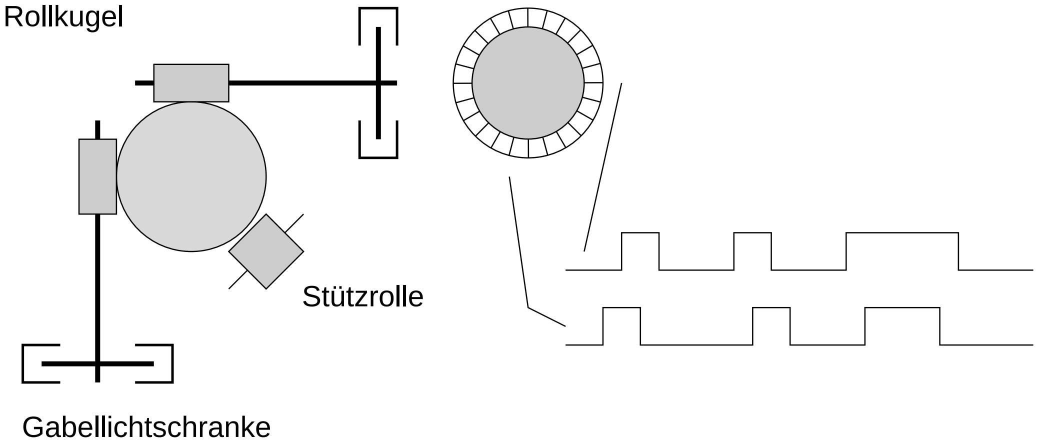




(Kraftkennlinien für Cherry G80-3000, www.cherry.de)

- Tasten (Mikroschalter/Gummimatte) in QWERTZ-Anordnung
- Organisation als Tastenmatrix
- Ansteuerung und Auslesen im Multiplex-Verfahren
- geringe Datenraten (~ 10 Byte/s), serielles Protokoll
- immer noch das Standard-Eingabegerät

Maus



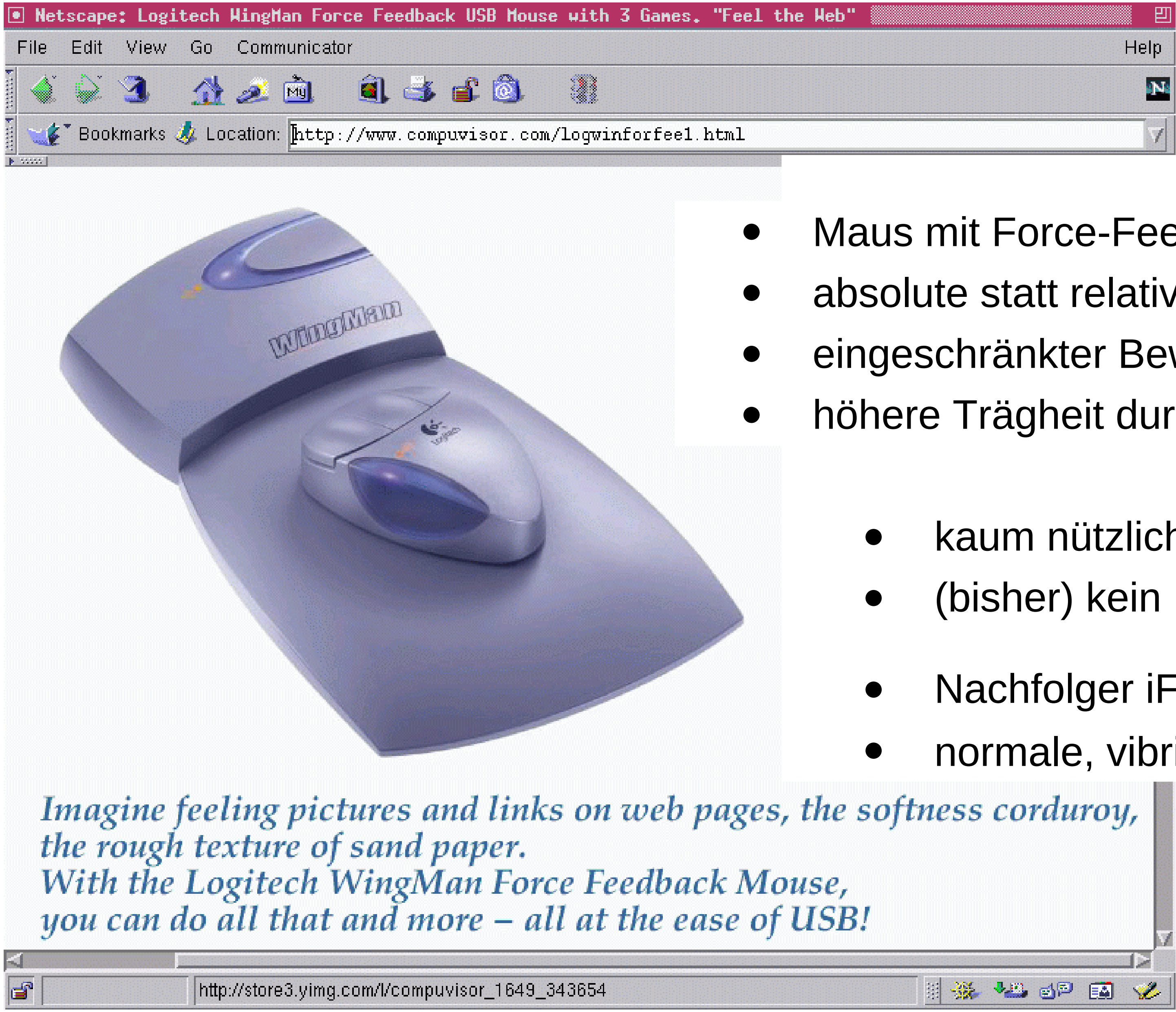
- Messung von relativen x/y-Bewegungen
- über klassische "Kugelmechanik"
- oder mit optischem Sensor auf gemustertem Untergrund

andere Zeigegeräte:

- Lichtgriffel, druckempfindliche Zeichenstifte
- "Spacemouse" mit x/y/z-Freiheitsgraden

(Zitat)

Force Feedback Mouse . . .



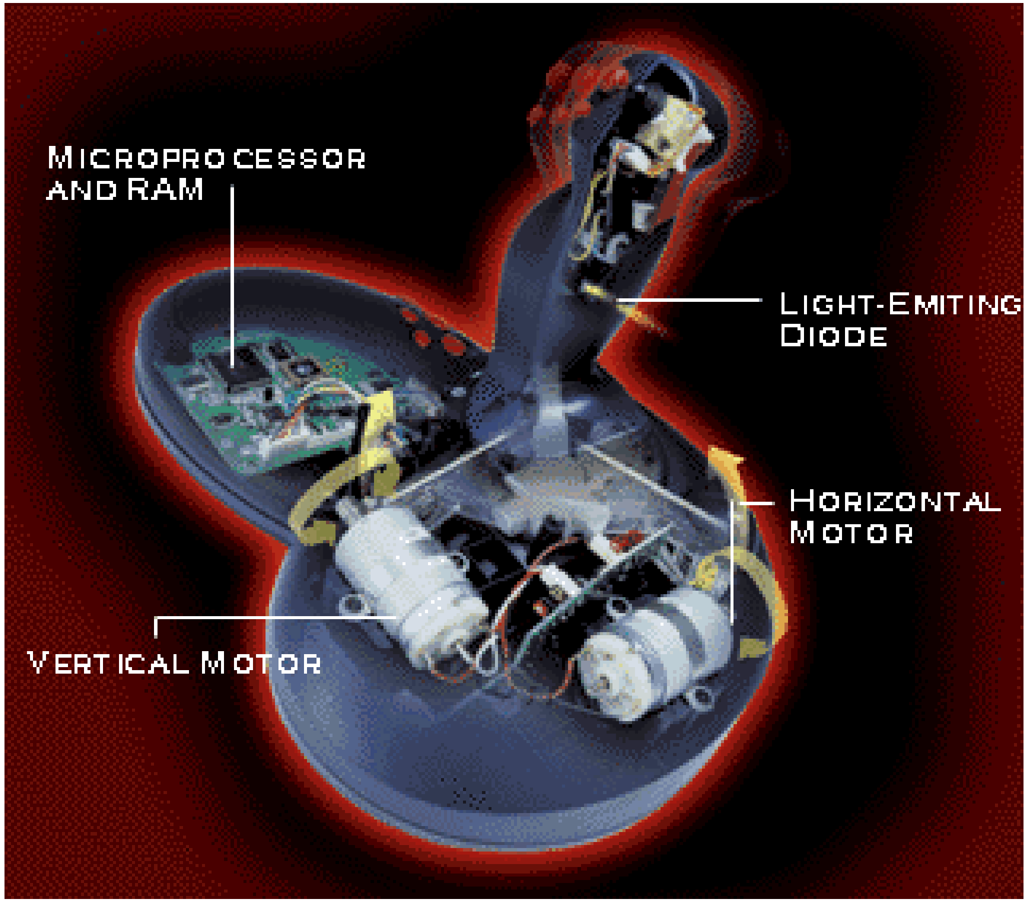
(!)

- Maus mit Force-Feedback (?!)
- absolute statt relativer Position
- eingeschränkter Bewegungsbereich
- höhere Trägheit durch die Mechanik

  - kaum nützliche Anwendungen
  - (bisher) kein Markterfolg

  - Nachfolger iFeel-Mouse:
  - normale, vibrierende Maus

Force-Feedback Joystick



Joystick:

- Arm auf Kugelgelenk
- 2 Freiheitsgrade X, Y
- Rückstellung mit Federn

- diverse Tasten, Coolie-Hat
- evtl. zusätzliche Achsen

mit Force-Feedback:

- X/Y-Achsen mit Motoren statt Federn
- Rückstellkräfte und Vibrationen beliebig einstellbar
- eigener Mikrorechner zur Regelung

Braille-Zeilen

- Darstellung einer Textzeile im Braille-Code
- entsprechende Anzahl einzeln magnetisch betätigter Stifte
- erlaubt Rechnerzugang für Blinde, aber recht teuer

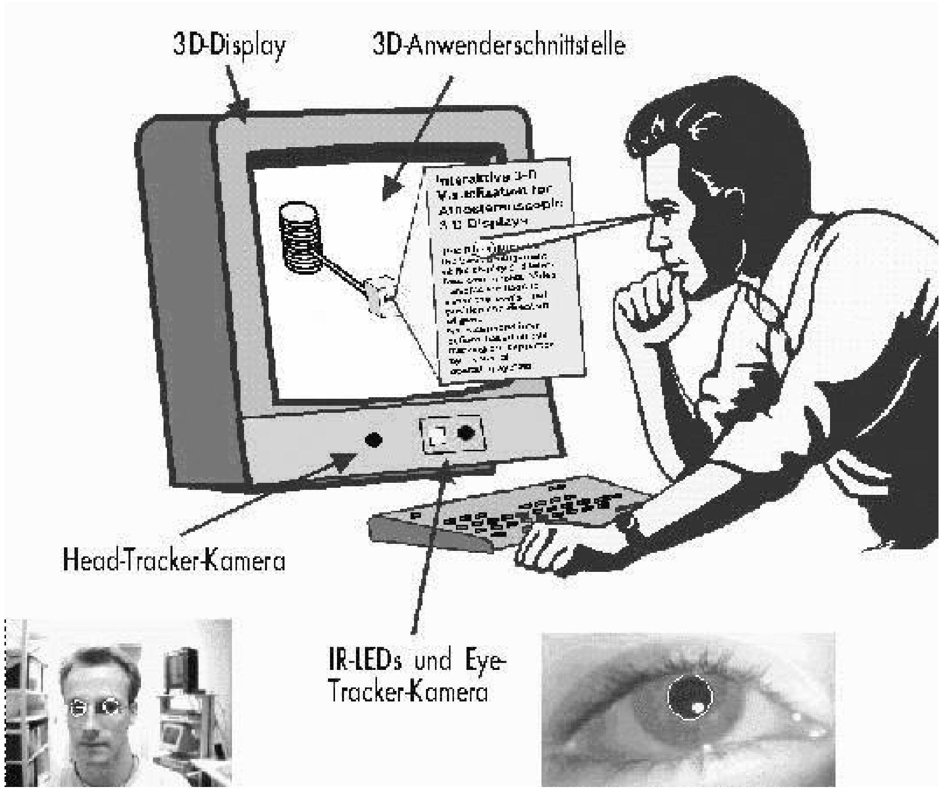
(diverse Hersteller, hier: www.braillenet.jussieu.fr/accessibilite/livreblanc/handvis.html)

Sensable: Phantom

- Positionsmessung des Handgriffs
- 6 Freiheitsgrade: x,y,z-Position, 3 Rotationswinkel
- eingebaute Bremsen für Force-Feedback
- Anwendung z.B. für 3D-Konstruktion

(www.sensable.com)

Eye- und Head-Tracking

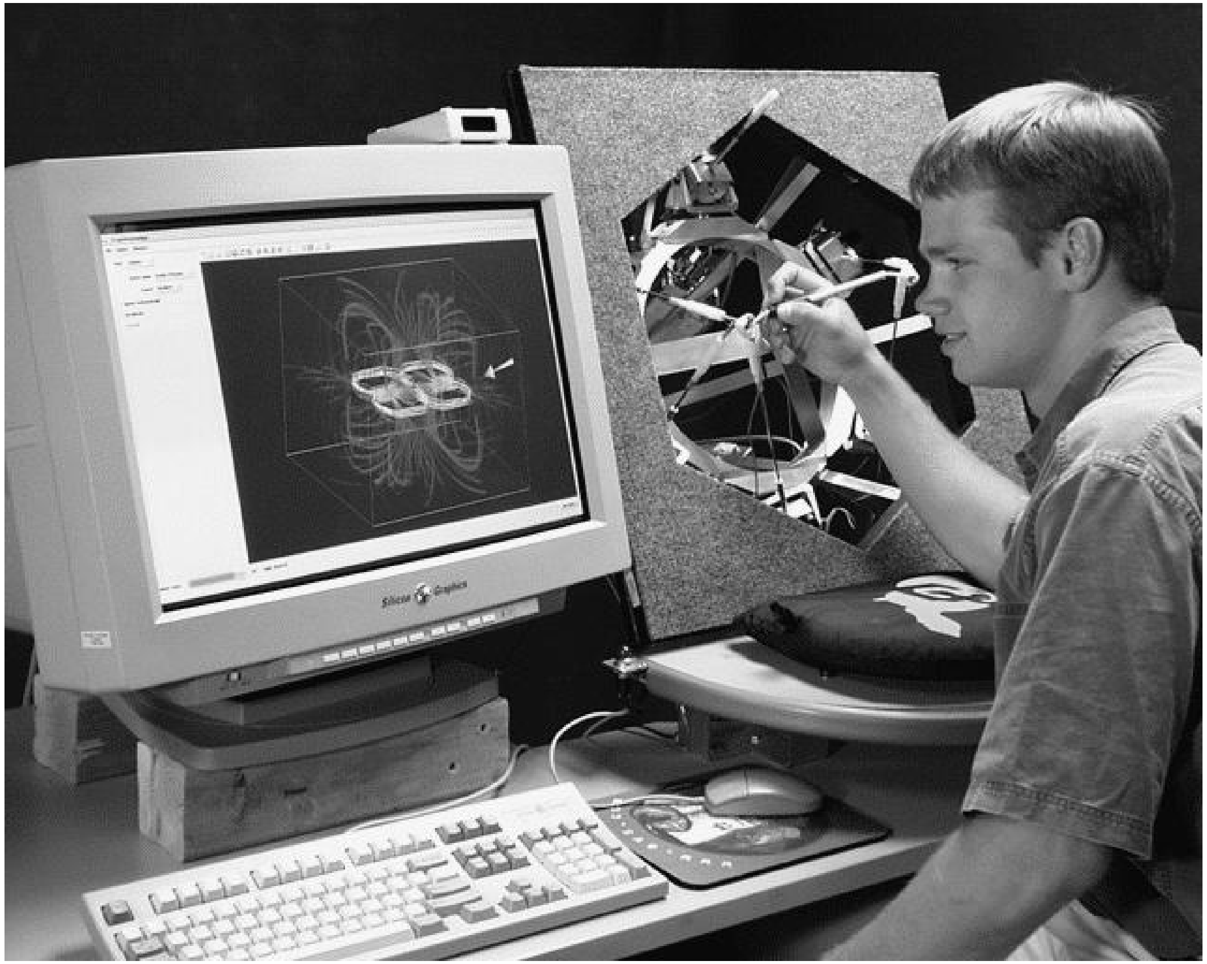


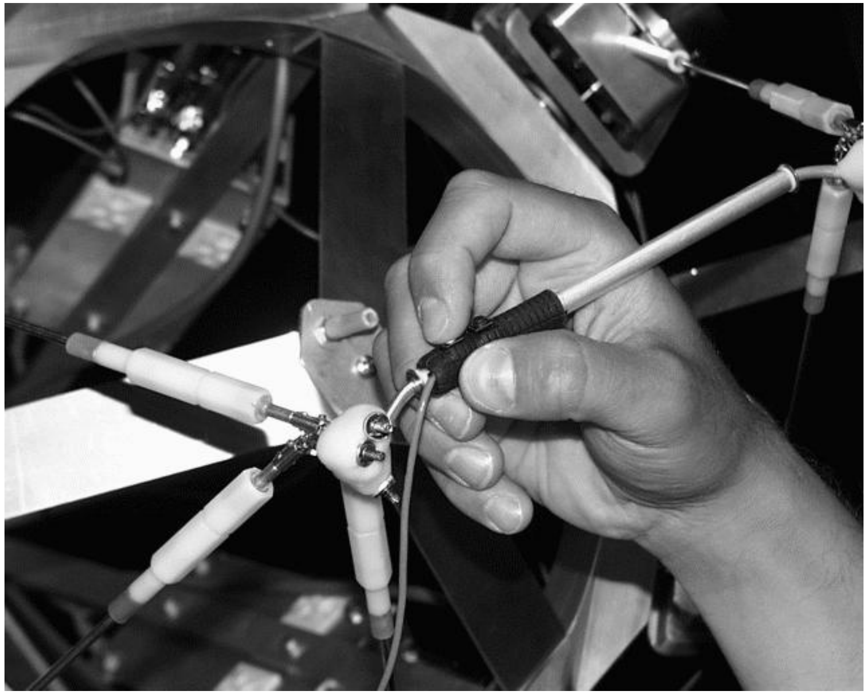
Messung der Augen-/Kopfposition:

- korrekte Benutzerperspektive
- Ansteuerung von 3D-Displays
- LOD: level-of-detail Verfahren
- Darstellung der "Aufmerksamkeitspunkte"

- mit stationären Kameras und (aufwendiger) Bildverarbeitung
- oder Brille mit Sensoren für Pupillenposition
- und Kopfpositionsmessung (z.B. Marker am Helm)

Uni Colorado: Eingabestift





- Prototyp für 6 DOF-Eingabe mit Force-Feedback
- Stift mit 5 Aktuatoren, Sensoren mit 0.08N, Motoren bis 8.0N
- Silicon Graphics Onyx2, Shutterbrille, 5 DSPs zur Motorsteuerung

(http://osl-www.colorado.edu/Research/haptic/hapticInterface.shtml)

Beispiel: Moleküldesign

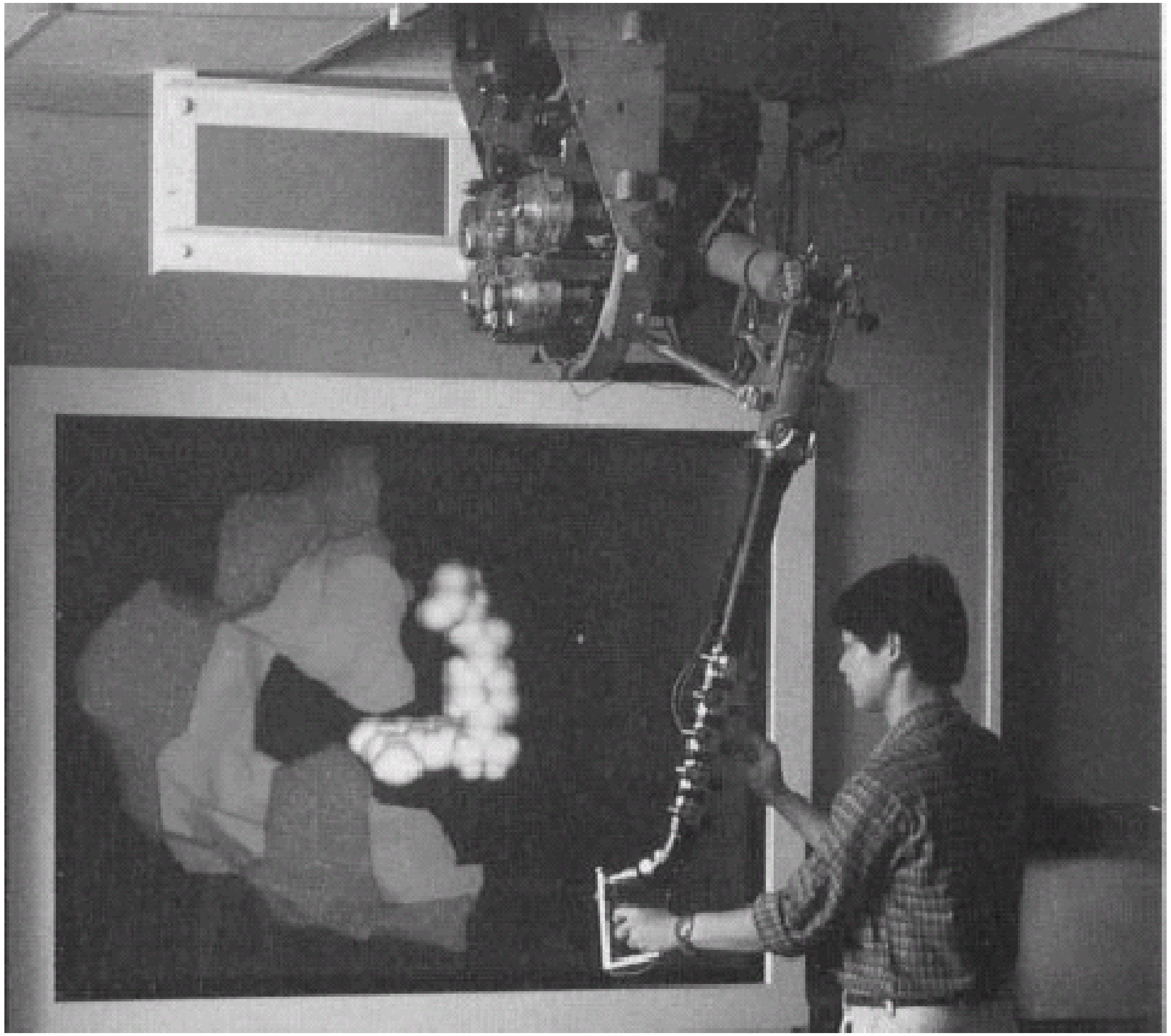
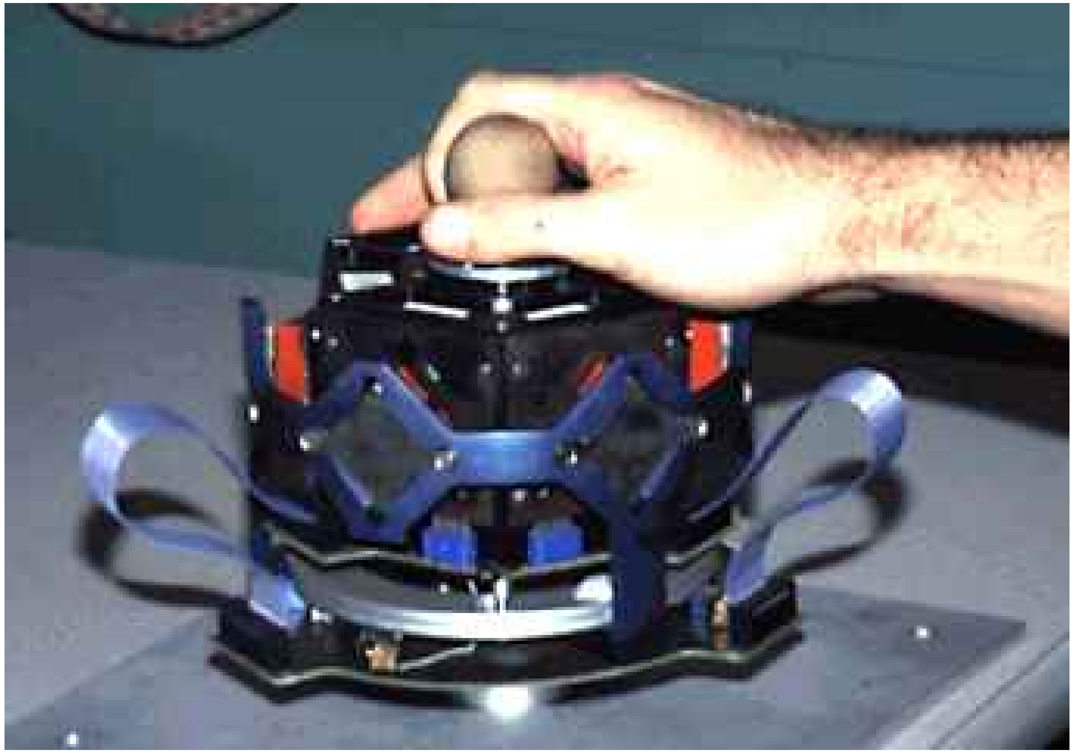


Figure 1: The *Docker* application simulates the forces between a drug and its receptor site in a protein as the user guides it to the minimum-energy configuration.

(Taylor, Scientific Applications of Force Feedback, SIGGRAPH'99)

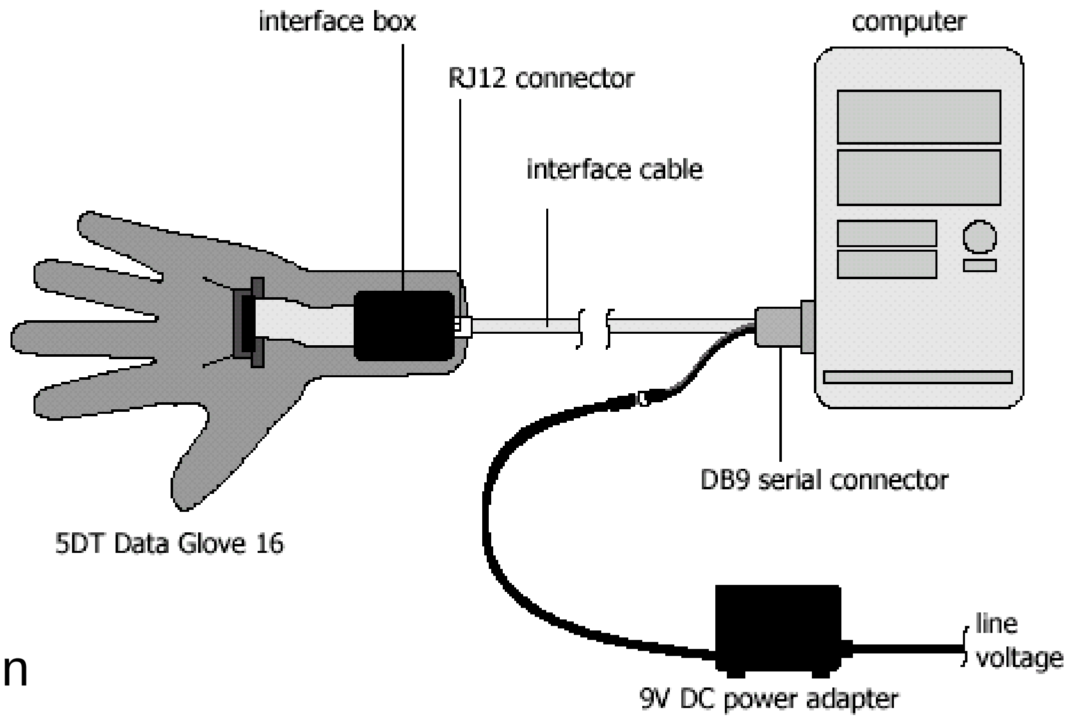
CMU: Magnetic Levitation



- direkte Ansteuerung des "Griffels"
- über rundum angeordnete Elektromagnete
- Aktuator ist vollkommen frei beweglich
- geringe Masse (und daher Trägheit) des Systems
- erlaubt sehr schnelle Bewegungen
- sehr komplexe Ansteuerung

(www-2.cs.cmu.edu/afs/cs/project/msl/www/haptic/haptic_desc.html)

Datenhandschuh



"data glove": Handschuh mit Sensoren

- Messung der relativen Handposition (Neigungswinkel)
- Messung der Fingerposition (Dehnungsmeßstreifen / opt. Sensoren)
- A/D-Wandlung der Werte, Übertragung zum Rechner (Kabel / Funk)

=> Standard-Eingabegerät für VR-Anwendungen

=> direkte Manipulation virtueller Objekte

(Abb.: Five Dimension Technologies)

Datenhandschuh: Freiheitsgrade

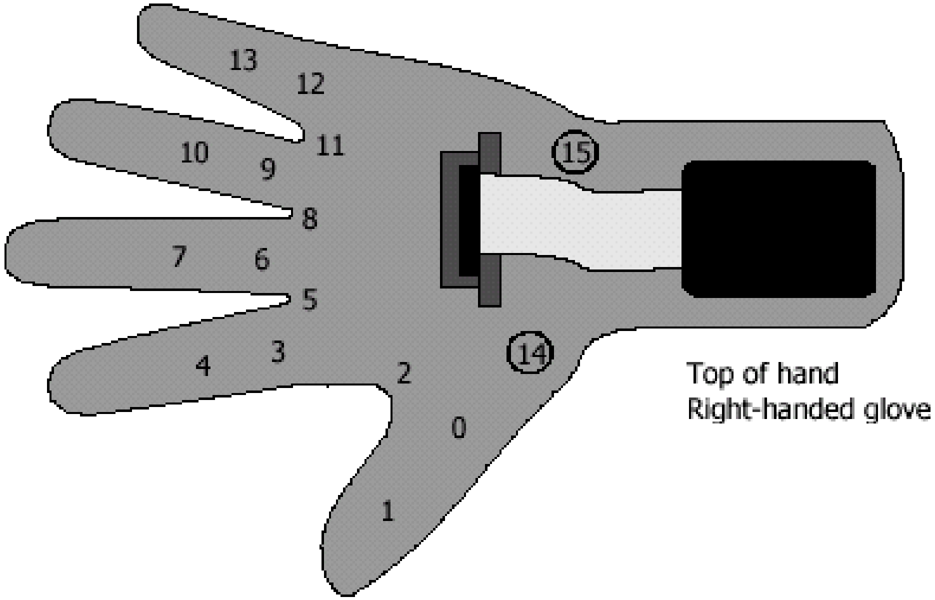


Figure 15 - Sensor mappings for the 5DT Data Glove 16

Sensor	*Driver Sensor Index*	*Description*
0	0	Thumb flexure (lower joint)
1	1	Thumb flexure (second joint)
2	2	Thumb-index finger abduction
3	3	Index finger flexure (at knuckle)
4	4	Index finger flexure (second joint)
5	5	Index-middle finger abduction
6	6	Middle finger flexure (at knuckle)
7	7	Middle finger flexure (second joint)
8	8	Middle-ring finger abduction
9	9	Ring finger flexure (at knuckle)
10	10	Ring finger flexure (second joint)
11	11	Ring-little finger abduction
12	12	Little finger flexure (at knuckle)
13	13	Little finger flexure (second joint)
14	14	Thumb translation [not yet implemented]
15	15	Wrist flexure [not yet implemented]

Beispiel für die Anordnung der Sensoren (5DT Glove16):

- 2 Krümmungssensoren pro Finger
- 4 Sensoren für Fingerspreizung
- Daumen- und Handgelenkneigung geplant
- keine Drucksensoren an den Fingerspitzen (!)

Datenhandschuh: Datenformat

The measured flexures of each of the sensors are the returned in this packet. The packet consists of 36 bytes and has the following structure:

`header` `s1high` `s1low` ... `s16high` `s16low` `checksum` `trailer`

The `header` is always two bytes long:

0x3C (an ASCII "<", decimal value 60)

0x44 (an ASCII "D", decimal value 68)

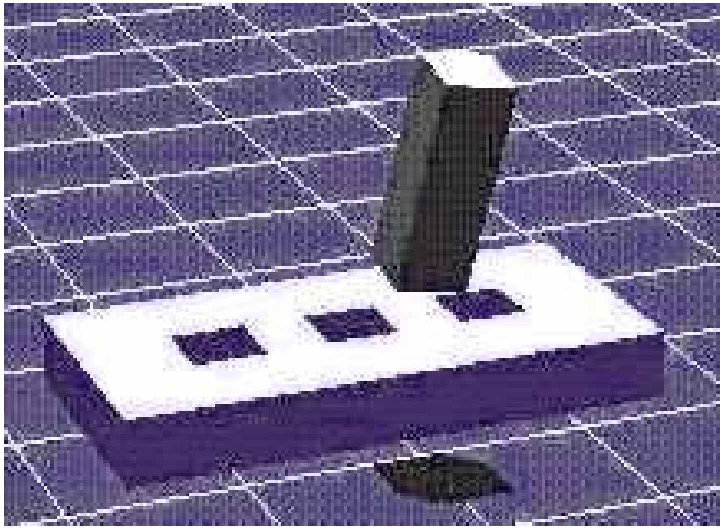
After the header, two bytes are sent for each sensor. The high order byte is sent first, then the low order byte. The value of the sensor is therefore (**s_high**) x 256 + **s_low**.

The `checksum` is the least significant byte of the addition of all the sensor values in the packet.

At the end of the packet, a trailing byte with the value of **0x3E** (an ASCII ">", decimal value 62) is sent.

- einfache RS-232 Schnittstelle, 115 kbps
- 100 Samples aller Sensoren pro Sekunde

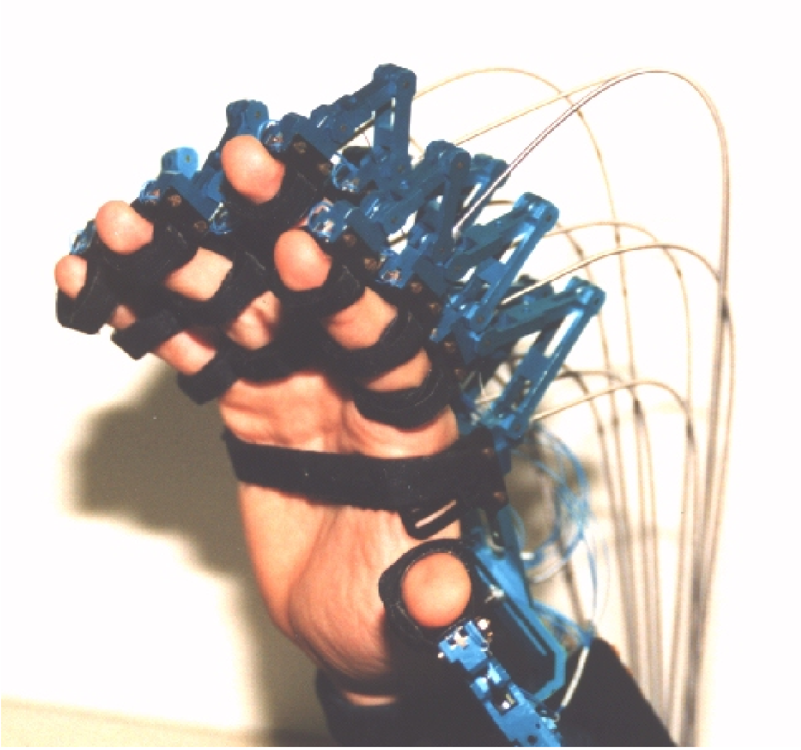
Datenhandschuh: VRML

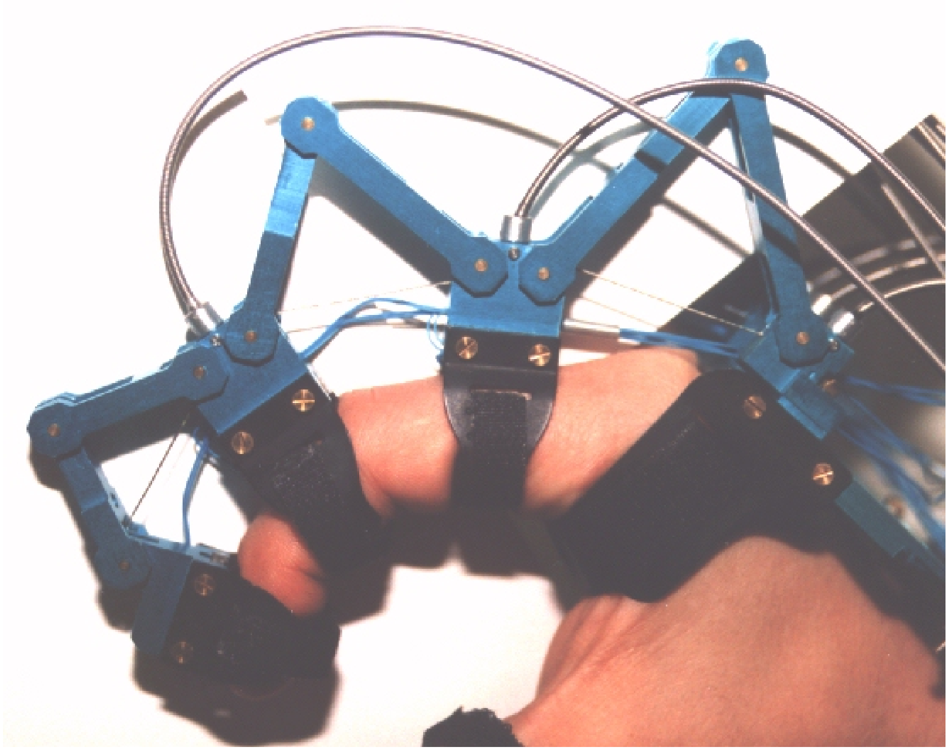


Anwendungsbeispiel:

- direkte Manipulation von Objekten im 3D-Editor
- direkter Zugriff auf verdeckte (unsichtbare) Objekte
- oder die Rückseite von Objekten
- Kombination mit force-feedback wäre ideal ...

Datenhandschuh: Force-Feedback





- erfordert "Außenskelett" mit vielen Motoren
- komplexe Seilzugmechanik zur Gewichtsreduzierung
- für realistische Effekte beträchtliche Kräfte erforderlich
  (Beispiel: antippen an eine harte, virtuelle Wand)

(www.caip.rutgers.edu/~bouzit/lrp/glove.html)

Datenhandschuh: Force-Feedback

- Fernsteuerung des Roboterarms mit Kraftrückmeldung
- ideal für fein"fühliges" Arbeiten

hier die Motoren (!)

Datenhandschuh: CyberForce

- CyberGlove: Datenhandschuh, Positionsmessung der Finger
- CyberGrasp: Kraftrückmeldung für die Finger (Seilzüge, s.o.)
- CyberForce: Positionsmessung und Kraftrückmeldung für die Handposition, Arbeitsbereich ca. 50 cm²

=> volle Messung und Manipulation der Handbewegungen

- Demo (Video)

(www.immersion.com/products/3d/interaction/cyberforce.shtml)

Datenhandschuh: Gestenerkennung

mit Datenhandschuh:

- Tastatureingabe problematisch
- aber neue Anwendungen möglich
- Beispiel: Gestenerkennung

(www.5DT.com)

DirectX: Force-Feedback

DirectX:

- für effizienten Hardwarezugriff unter Windows
- Direct3D, DirectDraw, DirectShow, DirectSound, ...

- Zugriff auf Eingabegeräte via DirectInput
- seit DirectX5 Unterstützung von Force-Feedback Geräten
- also: Joysticks und Lenkräder

- DirectInput definiert Standardfunktionen und Aufrufe
- Applikation listet Geräte und deren Fähigkeiten auf
- Umsetzung der Effekte über gerätespezifische Treiber
- Auswahl und "Download" der Effekte bei Programmbeginn

(Inside DirectX)

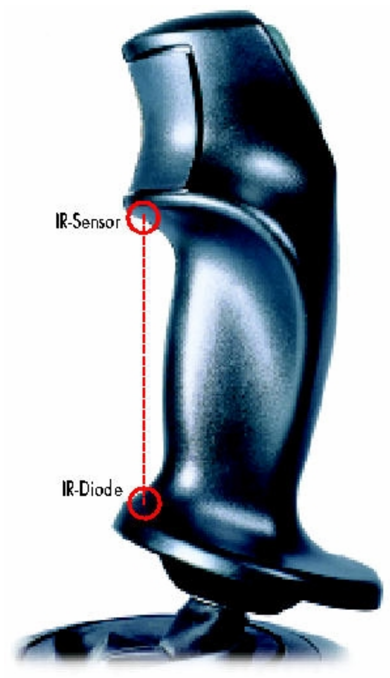
DirectInput: Sicherheit . . .

Force-Feedback-Systeme:

- können den Anwender verletzen
- oder zumindest irritieren

- Gefahr selbst bei "schwachen" Geräten
- Joysticks u.ä. bis ca. 10 N (1Kg)

- Schutzmaßnahmen erforderlich
- Beispiel: Sensor im Microsoft SideWinder Joystick

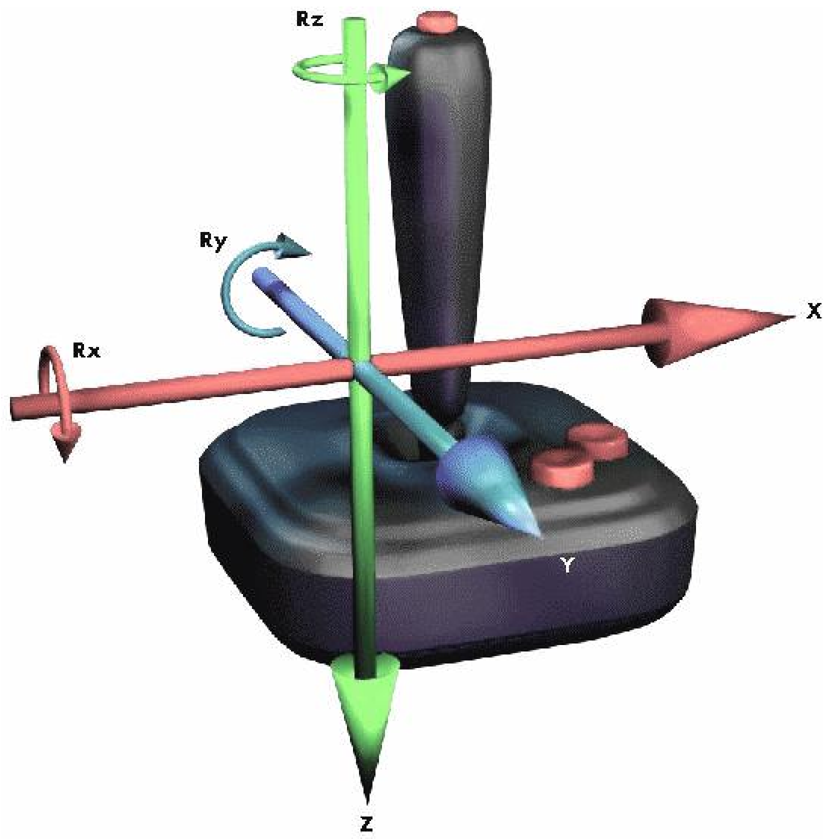


DirectInput: Koordinatensystem

- starrer Körper hat 6 Freiheitsgrade
  Position (x,y,z)
  Lage (Rx,Ry,Rz)

- Zuordnung zu Mausposition
- bzw. Joystick-Achsen

- kartesisches / Polar-Koordinatensystem
- DirectX übernimmt die Umrechnungen
- Kraftwirkung wie "Windrichtung" angeben



- siehe DirectX Dokumentation und DDK-Beispiele

(c't 09/99 216)

DirectX: Enumeration

```
LPDIRECTINPUTDEVICE8  g_lpDIDevice = NULL;
hr = g_lpDI->EnumDevices(DI8DEVTYPE_GAMECTRL,
                         DIEnumDevicesProc,  // Callback function
                         NULL, DIEDFL_FORCEFEEDBACK | DIEDFL_ATTACHEDONLY);
if (FAILED(hr)) ... // No force-feedback joystick available...


BOOL CALLBACK DIEnumDevicesProc( LPCDIDEVICEINSTANCE lpddi, LPVOID pvRef )
{
   HRESULT hr;
   GUID DeviceGuid = lpddi->guidInstance;

   // Create game device, request exclusive access
   hr = g_lpDI->CreateDevice(DeviceGuid, &g_lpDIDevice, NULL);
   hr = g_lpDIDevice->SetCooperativeLevel(g_hwndMain, DISCL_EXCLUSIVE | ... );

   // Set game data format
   hr = g_lpDIDevice->SetDataFormat(&c_dfDIJoystick);
   DIPROPDWORD DIPropAutoCenter;
   DIPropAutoCenter.diph.dwSize       = sizeof(DIPropAutoCenter);
   DIPropAutoCenter.diph.dwHeaderSize = sizeof(DIPROPHEADER);
   DIPropAutoCenter.diph.dwObj        = 0;
   DIPropAutoCenter.diph.dwHow        = DIPH_DEVICE;
   DIPropAutoCenter.dwData            = DIPROPAUTOCENTER_OFF;

   hr = g_lpDIDevice->SetProperty(DIPROP_AUTOCENTER, &DIPropAutoCenter.diph);
   return DIENUM_STOP;   // One is enough.
} // end DIEnumDevicesProc
```

(DirectX8.1 SDK)

DirectX: Enumeration der Effekte

```
LPDIRECTINPUTDEVICE8  g_lpDIDevice = NULL;
g_lpDIDevice = ... // Created by CreateDevice

// Try to find a periodic effect on the joystick device
HRESULT  hr;
GUID     guidEffect;
BOOL     EffectFound = FALSE;  // global flag

hr = g_lpDIDevice->EnumEffects(
         (LPDIENUMEFFECTSCALLBACK) DIEnumEffectsProc,
         &guidEffect, DIEFT_PERIODIC);
..

if (FAILED(hr)) ... // Internal error



// Callback function
BOOL CALLBACK DIEnumEffectsProc(LPCDIEFFECTINFO pei, LPVOID pv)
{
   *((GUID *)pv) = pei->guid;
   EffectFound = TRUE;
   return DIENUM_STOP;  // One effect is enough
}
```

DirectX: Effekt erzeugen

```
DWORD      dwAxes[2] = {DIJOFS_X, DIJOFS_Y};
LONG       lDirection[2] = {0, 0};
DIPERIODIC diPeriodic;      // type-specific parameters
DIENVELOPE diEnvelope;      // envelope
DIEFFECT   diEffect;        // general parameters

diPeriodic.dwMagnitude = DI_FFNOMINALMAX;
  diPeriodic.lOffset = 0;
  diPeriodic.dwPhase = 0;
  diPeriodic.dwPeriod = (DWORD)(0.05 * DI_SECONDS);

diEnvelope.dwSize = sizeof(DIENVELOPE);
  diEnvelope.dwAttackLevel = 0;
  diEnvelope.dwAttackTime = (DWORD)(0.5 * DI_SECONDS);
  diEnvelope.dwFadeLevel = 0;
  diEnvelope.dwFadeTime = (DWORD)(1.0 * DI_SECONDS);

diEffect.dwSize = sizeof(DIEFFECT);
  diEffect.dwFlags = DIEFF_POLAR | DIEFF_OBJECTOFFSETS;
  diEffect.dwDuration = (DWORD)(2 * DI_SECONDS);

  diEffect.dwSamplePeriod = 0;              // = default
  diEffect.dwGain = DI_FFNOMINALMAX;        // no scaling
  diEffect.dwTriggerButton = DIJOFS_BUTTON0;
  diEffect.dwTriggerRepeatInterval = 0;
  ...
LPDIRECTINPUTEFFECT  g_lpdiEffect;  // global effect object
hr = g_lpDIDevice->CreateEffect(
                       guidEffect,      // GUID from enumeration
                       &diEffect,       // where the data is
                       &g_lpdiEffect,   // where to put interface pointer
                       NULL);           // no aggregation
```

DirectX: Effekt abspielen

```
...
diEffect.dwSize = sizeof(DIEFFECT);
  diEffect.dwFlags = DIEFF_POLAR | DIEFF_OBJECTOFFSETS;
  diEffect.dwDuration = (DWORD)(2 * DI_SECONDS);
  ...

LPDIRECTINPUTEFFECT  g_lpdiEffect;  // global effect object

hr = g_lpDIDevice->CreateEffect(
                       guidEffect,      // GUID from enumeration
                       &diEffect,       // where the data is
                       &g_lpdiEffect,   // where to put interface pointer
                       NULL);           // no aggregation
if (FAILED(hr)) ...

diEffect.dwTriggerButton = DIEB_NOTRIGGER;


// To make a chain saw that starts and keeps going, change the
// dwDuration member as follows:

diEffect.dwDuration = INFINITE;
diPeriodic.dwPeriod = (DWORD)(0.08 * DI_SECONDS);
hr = g_lpdiEffect->SetParameters(&diEffect, DIEP_TYPESPECIFICPARAMS);

//  Next, start the effect.
g_lpdiEffect->Start(1, 0);

//  The effect keeps running until you stop it.
g_lpdiEffect->Stop();
```

Immersion: Force-Feedback Motivation

Even under excellent conditions for viewing and hearing, touch substantially improves user performance. Consider this:

How is it that a person can drink a cup of coffee while simultaneously reading a newspaper?

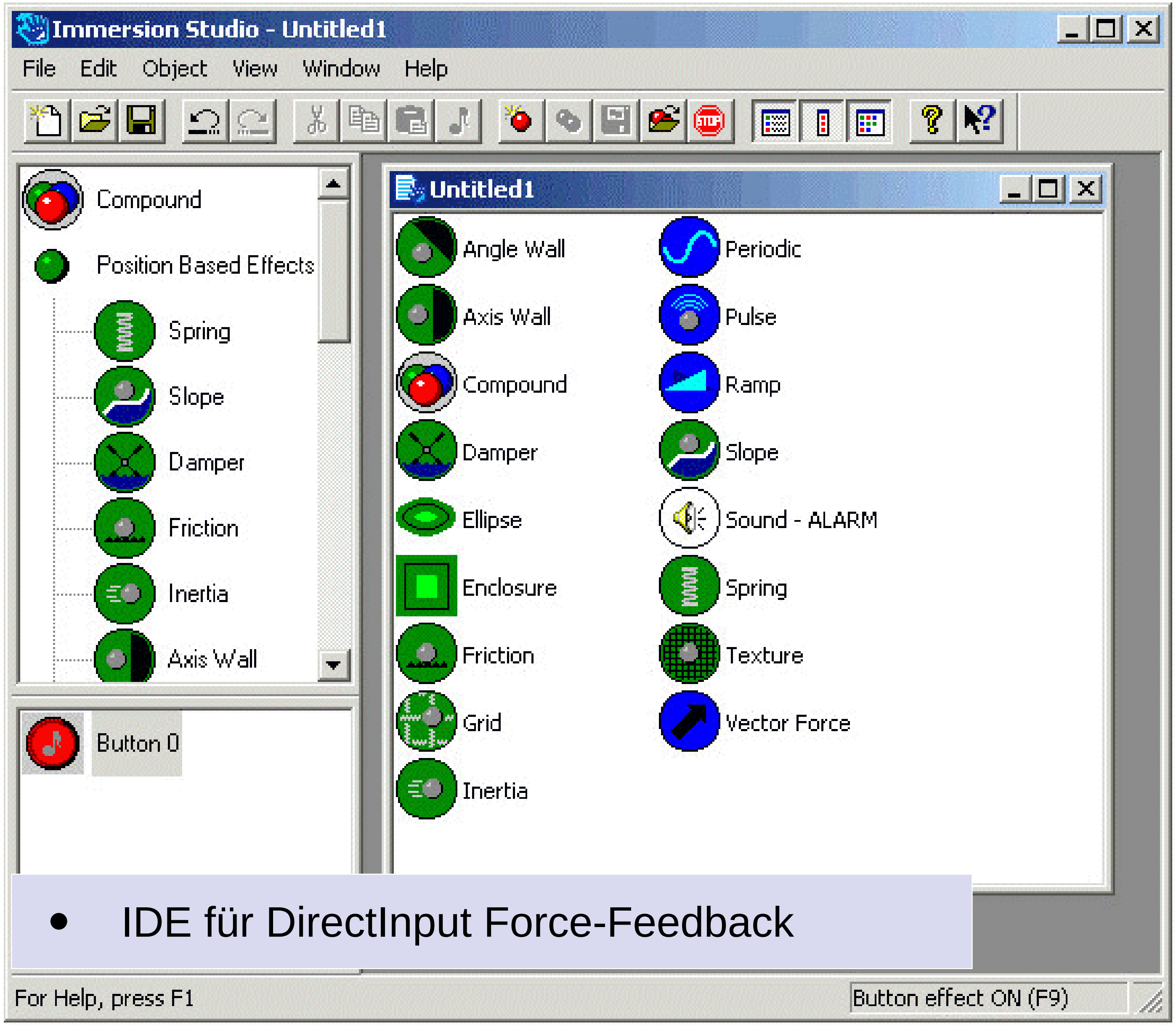
- Release grip on paper, casually adjusting other hand's grip to compensate for paper sag. Perhaps rest bottom of paper in lap (without looking)
- While still reading (perhaps slower), slowly feel around table for coffee mug, adjusting direction based on feel.
- When found, feel for handle and grip.
- Bring mug to lips, keeping mug level, rotating arm and wrist appropriately (by weight and grip pressure).
- Tilt mug back when mug is felt against lips.
- Return mug to table slowly, adjusting course if small collisions detected. Notice that you do not even need to look at the table, you can feel it through the mug!
- And you never lost your position on the page.

Now imagine without force feedback...

First, try to imagine holding the paper or turning the pages without feeling them!

(www.immersion.com)

Immersion: iForce Studio



- IDE für DirectInput Force-Feedback

(www.immersion.com)

Immersion: Geräte vs. Effekte

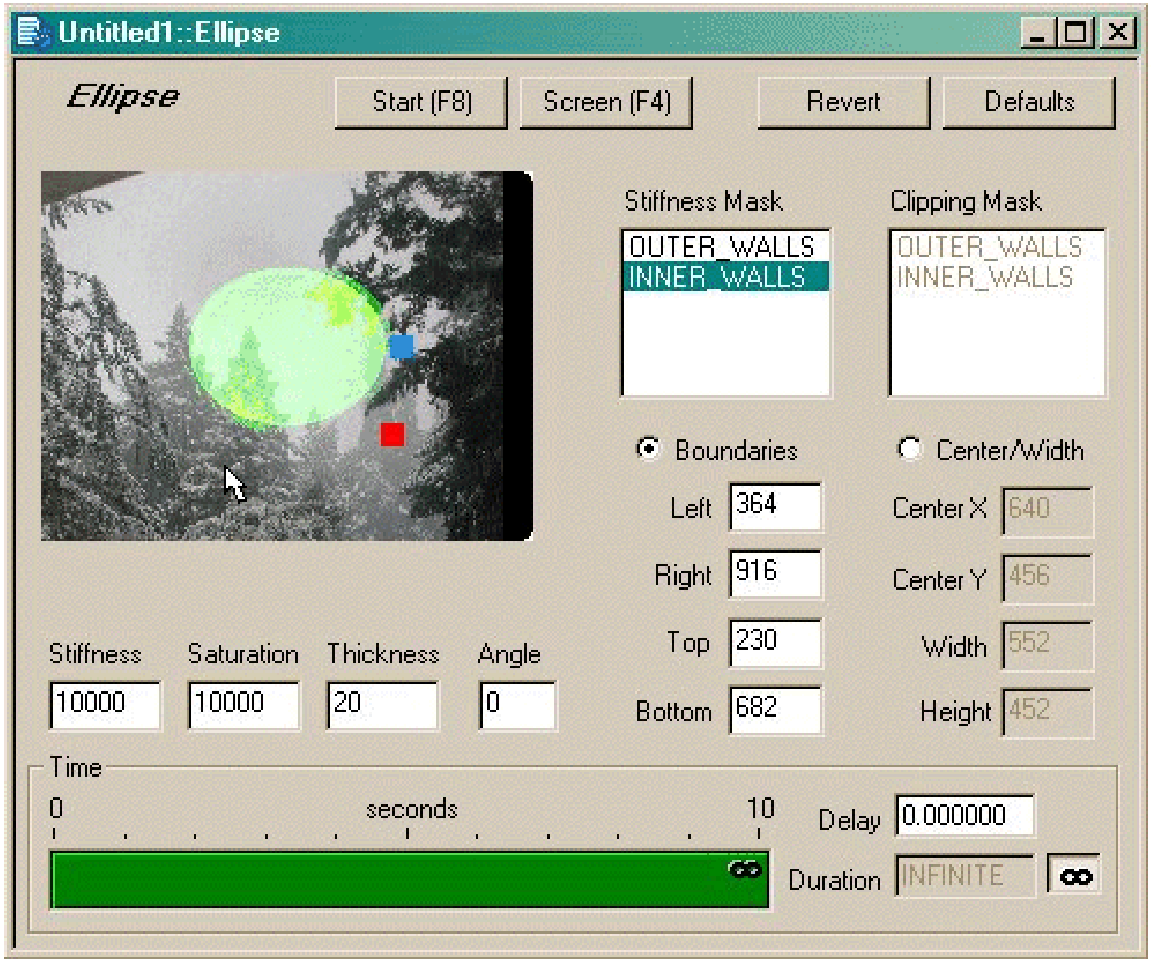
The nature of the device limits the types of touch effects felt. A steering wheel, for instance, turns left and right, so it plays effects on the x-axis only. A joystick, attached to its base at a single point, cannot play effects which require an exploration of 2-dimensional space, like textures and contours. A tactile mouse, free from a base attachment, is unable to exert lateral forces, making it unable play effects like pushes, pulls, gravity, or resistance to movement.

	Tactile Feedback Mouse or Trackball	Joystick	Steering Wheel	Full Force Feedback Gamepad	Full Force Feedback Mouse	Rumble Feedback Gamepad
Periodic	X	X	X	X	X	X
Texture	X				X	
Enclosure	X				X	
Ellipse	X				X	
Spring		X	X	X	X	
Grid					X	
Constant		X	X	X	X	
Ramp		X	X	X	X	
Damper		X	X	X	X	
Friction		X	X	X	X	
Inertia		X	X	X	X	

Immersion: "Ellipse"

Ellipse Effect (Position Based Effect)

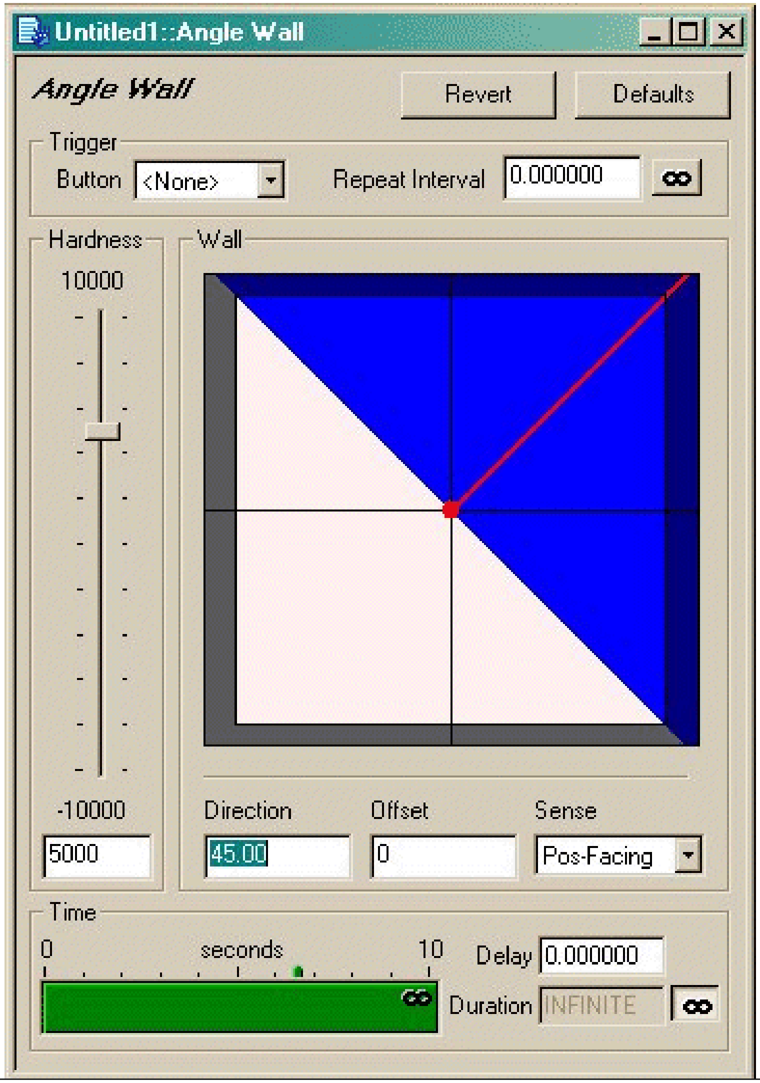
Ellipse can be used to attract the cursor to the inside of the ellipse, keep the cursor outside of the ellipse, or attract the cursor to the border surrounding the ellipse. Common uses of ellipse are for snapping to grid points or nodes, or attracting the cursor to icons. Enclosure effect is only available to TouchSense mouse devices only.



Immersion: "Angle Wall"

Angle Wall Effect (Position Based Effect)

Angle Wall is a position-based effect. It should feel like a wall at an arbitrary angle. Based on Spring, it uses the Immersion TouchSense IMM_CONDITION effect with a Spring GUID and the PositiveCoefficent or NegativeCoefficient set to zero.

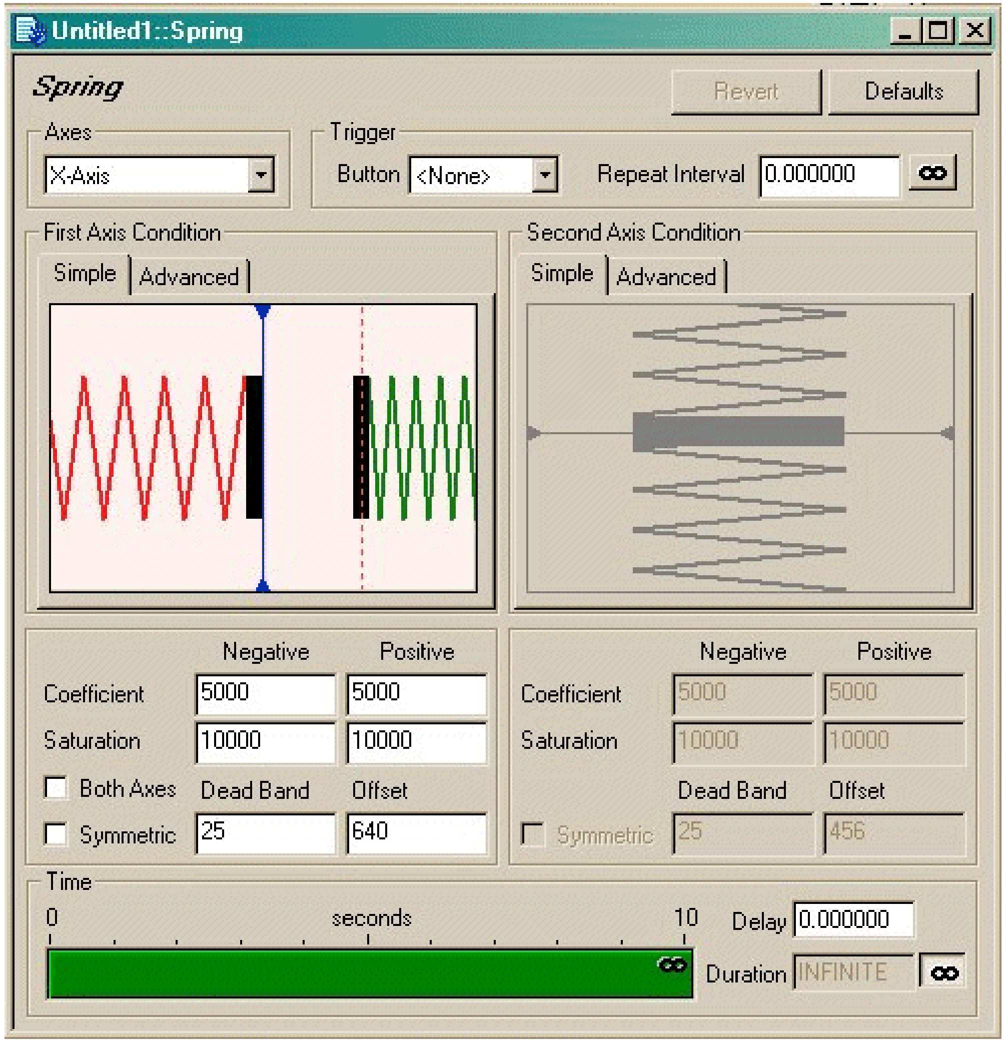


- basiert auf dem "Feder"effekt
- "hardness" = Federkonstante
- keine Wirkung im neg. Bereich
- Richtung frei wählbar

Immersion: "Spring"

Spring Effect (Position Based Effect)

Spring is a position-based effect. It should feel like compressing a spring. It uses the Immersion TouchSense IMM_CONDITION effect with a Spring GUID.

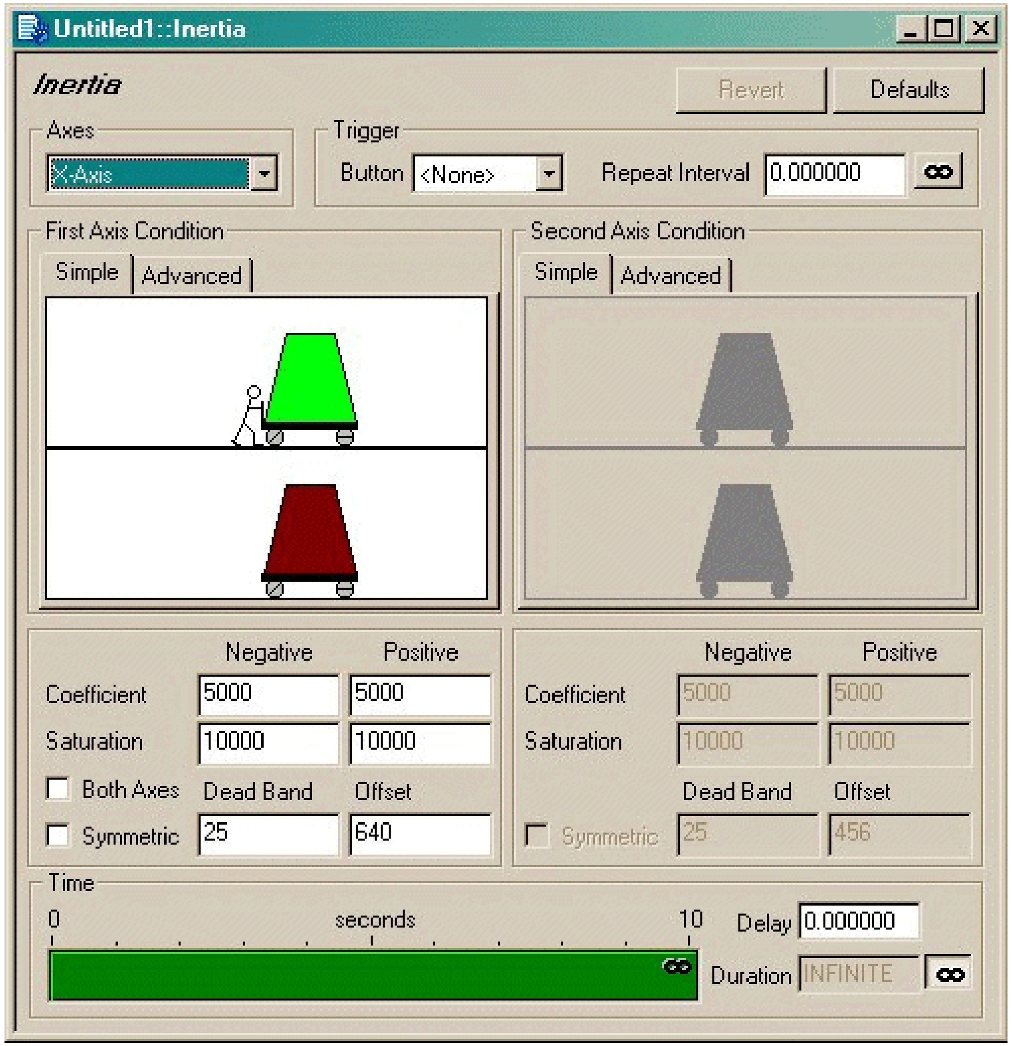


- modelliert eine Feder
- Rückstellkraft proportional zur Auslenkung
- für alle elastischen Medien
- optional mit Sättigungswert
- Richtung frei wählbar

Immersion: "inertia"

Inertia Effect (Position Based Effect)

Inertia is an acceleration-based effect. It should feel like pushing a mass on wheels.
It uses the Immersion TouchSense IMM_CONDITION effect with an Inertia GUID.

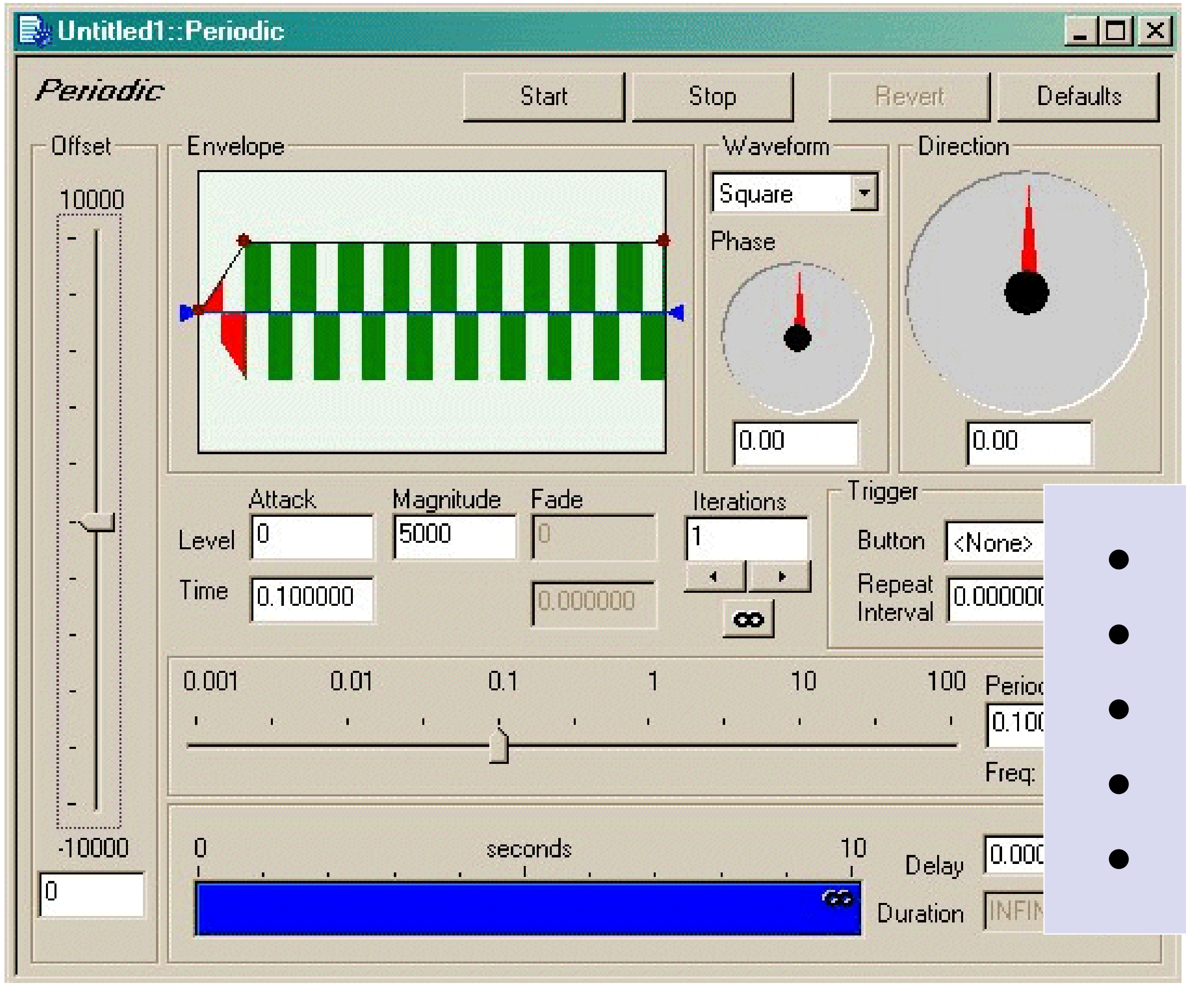


- Trägheitseffekt
- Objekt (Masse) auf Rädern...
- Masse des Objekts
- max. Geschwindigkeit
- Richtung frei wählbar

Immersion: "periodic"

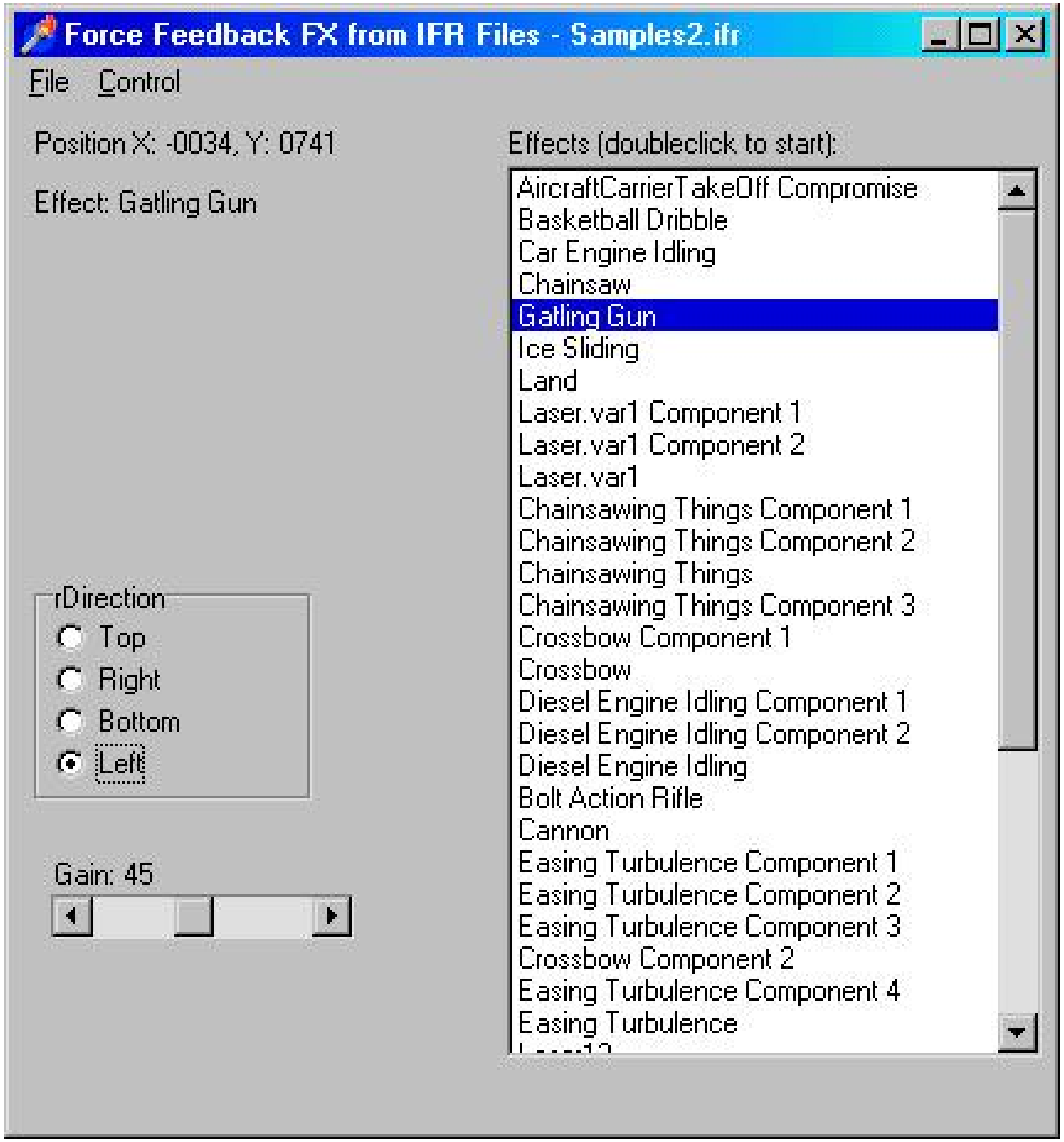
Periodic Effect (Time Based Effect)

The Periodic effect is a force that varies over time, such as a sine wave, square wave, etc. It can feel like a simple back and forth motion or a high frequency vibration. It uses the Immersion TouchSense IMM_PERIODIC effect.



- Vibrationseffekt
- Sinus, Rechteck, usw.
- Frequenz, Amplitude,
- Hüllkurve (Start)
- Richtung frei wählbar

DirectX: Beispiel für Force-Effekte



(c't 15/99 180)