



Computational Design I&II

Computational Design I&II

Prof. Dr. Benjamin Dillenburger

[Computational Design I&II](#) Copyright © Prof. Dr. Benjamin Dillenburger. Alle Rechte vorbehalten.

Inhalt

Einleitung	4
I. Computer Aided Design	
1. Computer	8
Digitale Signale und Binärcode	8
Vom Jacquard Webstuhl zur Rechenmaschine	9
Bildtafeln	10
2. Computer Aided Design	14
Technisches Zeichnen	14
Entwicklung von CAD	14
Anatomie einer CAD Software	16
Rhino als Beispiel	17
Grasshopper als Beispiel für Visuelle Programmierung	20
Bildtafeln	20
3. Mensch-Computer-Interaktion	25
Mensch – Maschine Interaktion	25
Mixed Reality	26
Internet of Things	26
Bildtafeln	28
II. Digitale Geometrie	
4. Darstellende Geometrie	31
Entwicklung der Darstellende Geometrie	31
Projektionen	32
Koordinatensysteme	34
Vektoren	36
Transformationen	37
Constructive Solid Geometry	37
Platonische Körper	38
Bildtafeln	39
5. Parametrische Kurven und Flächen	42
NURBS	42
Kurven	42
Flächen	45
Parametrische Flächen	46
NURBS Flächen	47
Bildtafeln	48
6. Geometrische Strukturen	51
Punktwolken	51
Mesh – Diskrete Flächennetze	52
Voxel – Diskrete Volumen	52
Implizite Flächen	53
Bildtafeln	55
7. Topologie, Graphen und Netzwerke	58
Topologie	58
Graphen	59
III. Digitale Ausgabe	
8. Digitale Bilder	63
Bilddaten	63
Bildsynthese – Rendering	65
Bildkomposition	65
Bildtafeln	68
9. Digitale Fabrikation	70
Modelle	70
CNC	71
Rapid Prototyping	71
IV. Digitale Prozesse	

10. Daten und Modelle	73
Technische Entwicklungen	73
Theoretische Grundlagen	75
Digitale Datenstrukturen	77
Digitale Prozessketten	78
11. Code	79
Algorithmus	79
Programmiersprachen	79
Variablen	79
Wiederholung	80
Bedingungen	80
Bildtafeln	81
12. Parametrisches und Procedurales Design	83
Parameter	83
Parametrisches Design	83
Mapping	84
Rekursion	86
Fraktal	86
L-System	87
Shape Grammar	87
Bildtafeln	88
13. Analyse und Simulation	93
Geometrische Analysen	93
Netzwerk Analysen	95
Simulation	95
Physics Engine	95
Partikel	96
Agenten	96
Emergenz	97
14. Suche und Optimierung	98
Entwerfen als Variantenerzeugung und Einschränkung	98
Probleme	98
Heuristik	99
Evolutionäre Algorithmen	99
Fitnesslandschaft	100
Glossary	102

Einleitung

Dieses Buch ist ein Teil der Vorlesungsunterlagen zu dem Kurs Computational Design und deckt den theoretischen Teil ab. Es handelt sich dabei um eine kuratierte und komprimierte Sammlung an Texten aus verschiedenen Quellen, die sich mit den zentralen Aspekten des Themas auseinandersetzen. Diese Sammlung ist mit Illustrationen und Diagrammen zur Verständnishilfe ergänzt.

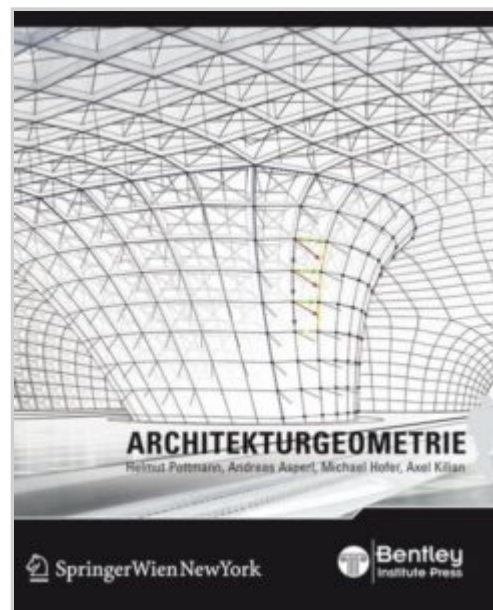
Anmerkung:

Bei dieser Version handelt es sich um die erste Version des Skripts. Kleinere Anpassungen werden wir noch vornehmen. Gleichzeitig sind wir um jedes konstruktives Feedback zum dem Skript dankbar. Bitte nutzt dafür die Moodle Forum Funktion!

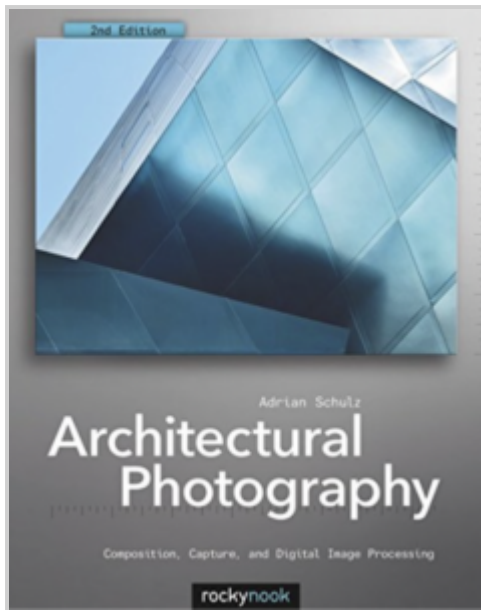
Die relevanten Quellen sind hier aufgelistet:



wikipedia.org

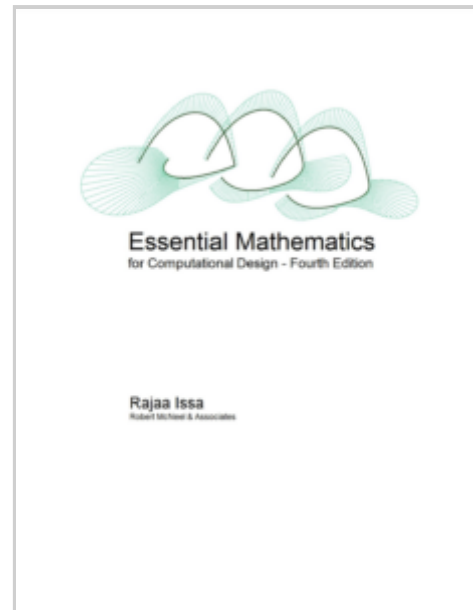


E. Lass, H. Pottmann, A. Asperl, M. Hofer, and A. Kilian, *Architekturgeometrie*, 1st ed. Ambra Verlag, 2009.

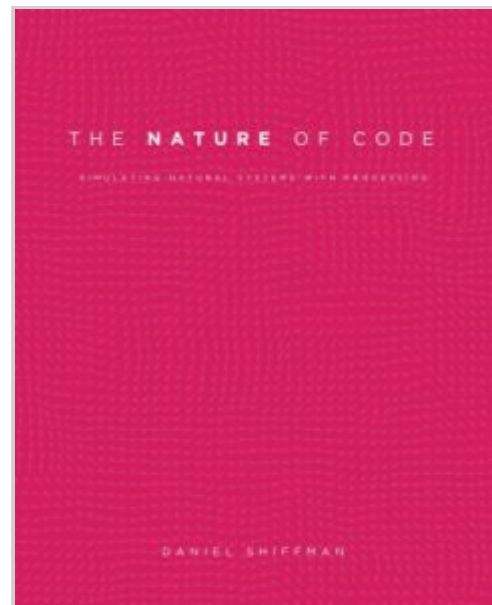


A.

Schulz, *Architectural Photography: Composition, Capture, and Digital Image Processing*, 2nd Edition. Santa Barbara, CA: Sebastopol, CA: Rocky Nook, 2012.

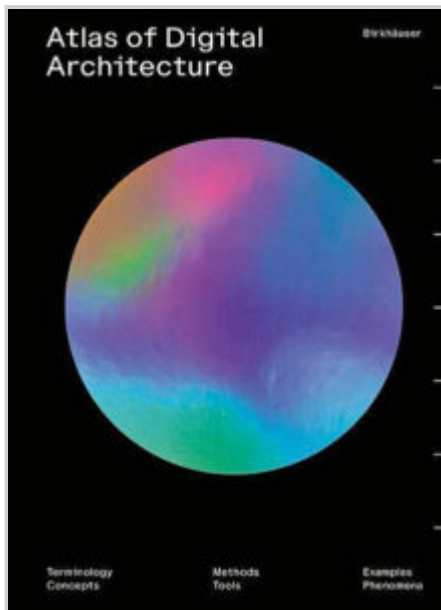


<https://developer.rhino3d.com/guides/general/essential-mathematics/>



The

Nature of Code, Daniel Shiffman



L.

Hovestadt, U. Hirschberg, and O. Fritz, *Atlas of Digital Architecture: Terminology, Concepts, Methods, Tools, Examples, Phenomena*, 1st ed. Boston: Birkhäuser, 2020.

1

Computer Aided Design

Computer

1.1 – Digitale Signale und Binärcode

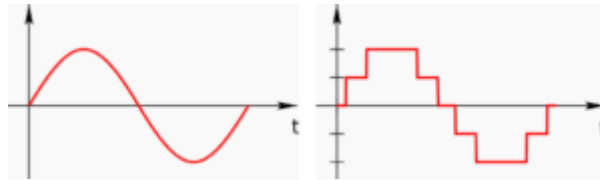


Abbildung 1.1 – Kontinuierliche (analoge) und diskrete (digitale) Signale

Ein **analoges Signal** ist eine physikalische Größe, die im Verlauf der Größe (Amplitude) als auch im zeitlichen Verlauf kontinuierliche Werte annehmen kann.

Ein **digitales Signal** (digitus: Finger, lat.) ist eine physikalische Größe, die nur bestimmte diskrete Werte annehmen kann. Die Werte entsprechen der Anzahl der vordefinierten Zustände. Werden zwei Zustände definiert (0 und 1), dann handelt es sich um binäre (digitale) Signale.

Ein **Binärcode** ist ein Code, in dem Informationen durch Sequenzen von zwei verschiedenen Symbolen (zum Beispiel 1/0 oder wahr/falsch) dargestellt werden. Die Bezeichnung leitet sich von der lateinischen Vorsilbe bi ab, welche die Bedeutung zwei oder doppelt hat.

```
01010111 01101001 01101011
01101001 01110000 01100101
01100100 01101001 01100001
```

Abbildung 1.2 – Binäre Darstellung des Wortes „Wikipedia“

Im weiteren Sinne war Leibniz wegbereitend für die Rechenmaschine im heutigen Sinne, den Computer. Er entdeckte, dass sich Rechenprozesse viel einfacher mit einer binären Zahlencodierung durchführen lassen, und ferner, dass sich mittels des binären Zahlencodes die Prinzipien der Arithmetik mit den Prinzipien der Logik verknüpfen lassen (siehe De progressionibus dyadicis, 1679; oder Explication de l'Arithmetique Binaire, 1703). Die hier erforschten Prinzipien wurden erst 230 Jahre später in der Konstruktion von Rechenmaschinen eingesetzt (z. B. bei der Zuse Z1). Leibniz hatte beim Bau einer Rechenmaschine, anders als frühere Erfinder, eher philosophische Motive. Mit dem viel bemühten Zitat, es sei „ausgezeichneter Menschen unwürdig, gleich Sklaven Stunden zu verlieren mit Berechnungen“, wird eine Grenze zwischen Mensch und Maschine gezogen. Dem Erfindergeist (Freiheit, Spontaneität und Vernunft) als das spezifisch Menschliche wird das Mechanische der technisch-natürlichen Kausalität gegenübergestellt.

„Es ist unwürdig, die Zeit von hervorragenden Leuten mit knechtischen Rechenarbeiten zu verschwenden, weil bei Einsatz einer Maschine auch der Einfältigste die Ergebnisse sicher hinschreiben kann.“

Gottfried Wilhelm Leibniz

Binärcodes bilden auf Grund ihrer Einfachheit in aller Regel die Grundlage für die Verarbeitung digitaler Informationen und werden deshalb häufig im Zusammenhang mit deren Verarbeitung genannt; „Computer funktionieren mit diesem Code“.

Viele der Binärcode-Arten sind im Gebiet der Informationstechnik entstanden und werden dort verwendet; der Ausdruck „Binärcode“ wird im Computer-Sprachgebrauch auch als Synonym für Maschinencode, Maschinenprogramm oder Maschinensprache verwendet.

1.2 – Vom Jacquard Webstuhl zur Rechenmaschine

Als **Jacquardwebstuhl** bezeichnet man den von [Joseph-Marie Jacquard](#) weiterentwickelten [Webstuhl](#). Beim ursprünglichen Jacquardwebstuhl wurden die Platinenbewegungen durch [Lochkarten](#) gesteuert. Dies war eine der ersten bekannten Anwendungen der Lochkartentechnik überhaupt. Zur **Hardware**, dem Webstuhl, haben wir hier also erstmals eine **Software**, die Lochkarte, die die Hardware steuern kann. Die meisten neueren Einrichtungen arbeiten mit elektromagnetischer Übertragung der vom Computer gegebenen Steuerungsimpulse. Diese Verwendung von austauschbaren Lochkarten zur Steuerung einer Abfolge von Vorgängen gilt als wichtiger Schritt in der Geschichte der Computerhardware und hat die Analytical Engine von Charles Babbage inspiriert.

Die **Analytical Engine** ([englisch](#) für *Analytische Maschine*^[1]) ist der Entwurf einer mechanischen [Rechenmaschine](#) für allgemeine Anwendungen. Sie stammt von dem britischen Mathematikprofessor [Charles Babbage](#) (1791–1871) und stellt einen wichtigen Schritt in der Geschichte des [Computers](#) dar.

Als [Programme](#) gibt es tabellarische Darstellungen; ein Satz von 1837/38 ist erhalten und enthält bereits das Beispiel für die Auflösung eines linearen Gleichungssystems, wie es von [Luigi Federico Menabrea](#) 1842 und [Ada Lovelace](#) 1843 veröffentlicht wurde; von letzterer ist wohl das Beispiel der Berechnung der [Bernoulli-Zahlen](#).

Eine **Turingmaschine** ist ein mathematisches [Modell](#) der [theoretischen Informatik](#), das eine [abstrakte Maschine](#) definiert. Bei diesem Rechnermodell werden nach festgelegten Regeln Manipulationen von [Zeichen](#) vorgenommen. Die Turingmaschine ist benannt nach dem britischen Mathematiker [Alan Turing](#), der sie 1936/37 einführte. Turingmaschinen machen die Begriffe des [Algorithmus](#) und der [Berechenbarkeit mathematisch](#) fassbar, das heißt, sie formalisieren diese Begriffe und bilden eines der Fundamente der [Theoretischen Informatik](#).

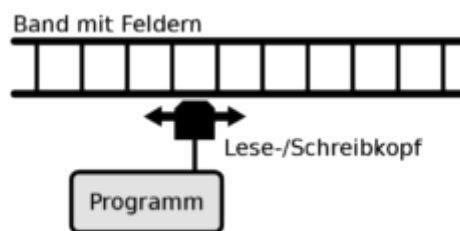


Abbildung 1.3 – 1-Band-Turingmaschine: abstraktes Modell eines Rechners, der mit nur drei Operationen (Lesen, Schreiben und Kopf Bewegen) sämtliche berechenbare Probleme lösen kann

1.3 – Bildtafeln



Abbildung 1.4 – Gottfried Wilhelm Leibniz. Portrait by the French engraver Pierre Savart (1737-1780) from Leibniz (Gottfried Wilhelm), *Opera omnia, nunc primum collecta, in Classes distributa...* (1768).



Abbildung 1.5 – Abgebildete Person: Ada Lovelace Ada Augusta Byron, einzige Tochter von Lord Byron. Heiratet William King im Jahre 1835. Sie wurden Graf und Gräfin von Lovelace im Jahre 1838. Ada Lovelace war eine begeisterte Mathematikerin und wird oft als erste Computer-Programmiererin angesehen, nachdem sie einen Algorithmus für die Analytical Engine von Charles Babbage schrieb.



Abbildung 1.6 - Passport photo of Alan Turing at age 16

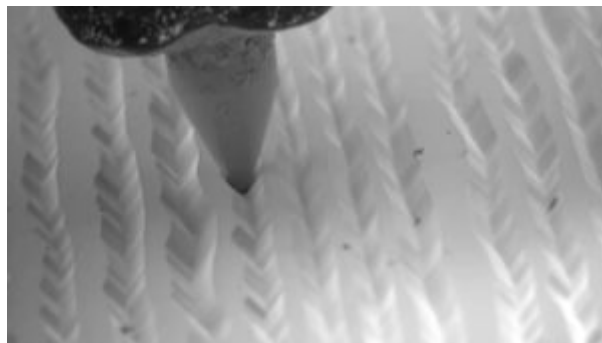


Abbildung 1.7 - Elektronenmikroskopische Aufnahme einer Plattennadel



Abbildung 1.8 - CD Innenrand bei ca. 1000 fach, Auflicht.

TABLE 86 MEMOIRES DE L'ACADEMIE ROYALE
DES
NOMBRES.

bres entiers au-dessus du double du plus haut degré. Car icy, c'est comme si on disoit, par exemple, que 111 ou 7 est la somme de quatre, de deux et que 1101 ou 13 est la somme de huit, quatre & un. Cette propriété sert aux Essayeurs pour peier toutes sortes de monnoies avec peu de poids, & pourroit servir dans les monnoies pour donner plusieurs valeurs avec peu de pieces.

Cette expression des Nombres étant établie, sert à faire tres-facilement toutes sortes d'operations.

Pour l'Addition
par exemple.

Pour la Soustraction.

Pour la Multiplication.

Pour la Division.

Et toutes ces operations sont si aisées, qu'on n'a jamais besoin de rien essayer ni deviner, comme il faut faire dans la division ordinaire. On n'a point besoin non-plus de rien apprendre par cœur icy, comme il faut faire dans le calcul ordinaire, où il faut savoir, par exemple, que 6 & 7 pris ensemble font 13, & que 5 multiplié par 3 donne 15, suivant la Table d'une fois au 100, qu'on appelle Pythagorique. Mais icy tout cela se trouve & se prouve de source, comme l'on voit dans les exemples précédens sous les signes & &.

Abbildung 1.9 – Seite aus „Explication de l'Arithmétique Binaire“, 1703 Gottfried Wilhelm Leibniz – Académie des sciences de Paris, 1703, sur Gallica. Gerhardt's Leibnizens mathematische Schriften, Band 7, p. 224, sur Gallica and on WikiCommons. On Wikisource : Explication de l'arithmétique binaire. Table of the numbers 0 to 32 in binary figures from the Leibniz's paper Explication de l'arithmétique binaire, qui se sert des seuls caractères 0 et 1, avec des remarques sur son utilité, et sur ce qu'elle donne le sens des anciennes figures chinoises de Fohy.

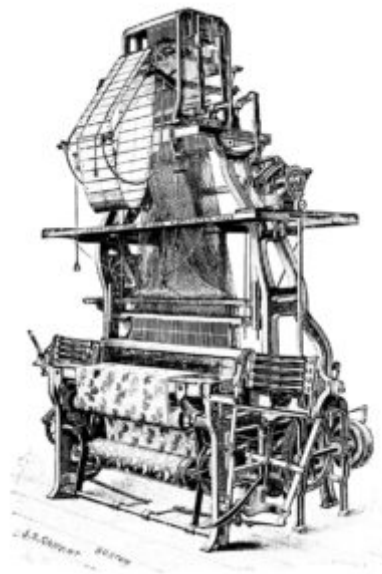


Abbildung 1.10 – The Jacquard Loom. Rights: This image is in the public domain. From: The Popular Science Monthly (New York: Popular Science Pub. Co., etc., 1891), <http://archive.org/details/popularsciencemo39newy>.

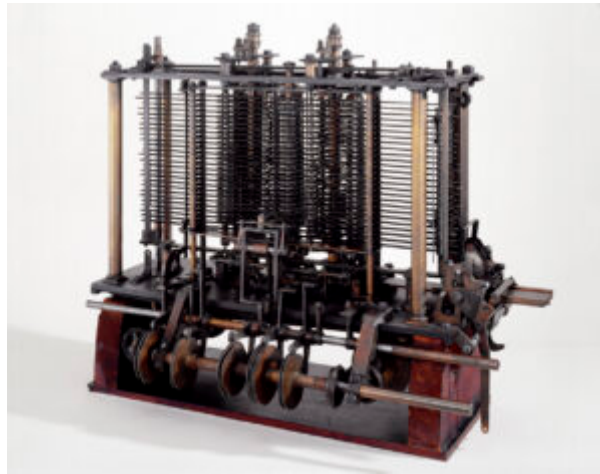


Abbildung 1.11 - This was the first fully-automatic calculating machine. British computing pioneer Charles Babbage (1791-1871) first conceived the idea of an advanced calculating machine to calculate and print mathematical tables in 1812. This machine, conceived by Babbage in 1834, was designed to evaluate any mathematical formula and to have even higher powers of analysis than his original Difference engine of the 1820s. Only part of the machine was completed before his death in 1871. This is a portion of the mill with a printing mechanism. Babbage was also a reformer, mathematician, philosopher, inventor and political economist.

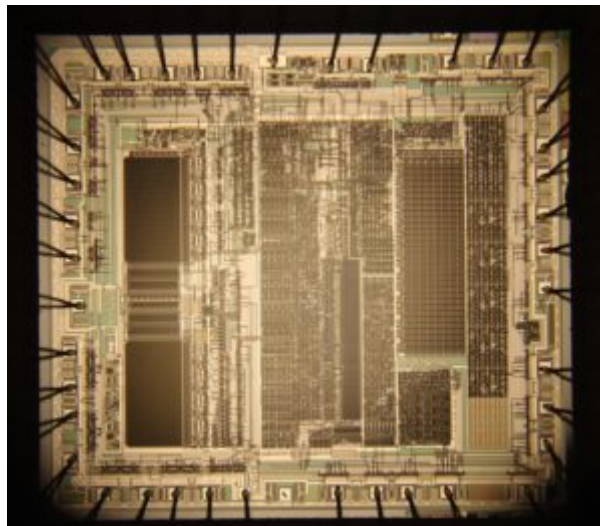


Abbildung 1.12 - 87C51 Mikrocontroller mit EPROM

Computer Aided Design

2.1 – Technisches Zeichnen

Die Anfänge des architektonischen, technischen Zeichnens wurden im bzw. vor dem [Mittelalter](#) gelegt, wie z. B. der [St. Galler Klosterplan](#) oder die Zeichnungen von [Villard de Honne-court](#) beweisen. Weitere Belege für die Entwicklung des technischen Zeichnens finden sich in frühen [Patentschriften](#) aus dem 19. Jahrhundert. Die Disziplin des technischen Zeichnens hat sich also über Jahrhunderte hinweg evolutionär zu einer modernen Technik entwickelt. Beim klassischen technischen Zeichnen am [Zeichenbrett](#) kommen früher wie heute verschiedene [Zeichenwerkzeuge](#), wie [Reißzeug](#), [Zirkelbesteck](#) und [Kurvenlineal](#) zum Einsatz. Bis um 1910 (für manche Anwendungen noch bis etwa 1965) wurde fast ausschließlich mit [Bleistift](#) und [Tusche](#) auf [Transparentpapier](#) gezeichnet. Dieses spannte man auf ein [Reißbrett](#) (Zeichenbrett) oder das Schräge Brett einer [Zeichenmaschine](#).

Blaupause

Die **Blaupause** (engl. **Blueprint**) ist die [Lichtpause](#) von einer durchsichtigen Vorlage, einer grafischen Darstellung, meist einer technischen Zeichnung, die weiße Linien auf einem bläulichen Papier ergibt. Bei dem Verfahren handelt es sich um das [Lichtpausverfahren](#), das auf [John Herschel](#) 1842 zurückgeht. Eine Blaupause ist die Kopie/Vervielfältigung (Lichtpause) einer auf Transparentpapier mit schwarzer Tusche angefertigten technischen Zeichnung. Im übertragendem Sinne steht Blaupause auch für [Schablone](#), [Muster](#), [Entwurf](#) oder [Template](#).^[1]

2.2 – Entwicklung von CAD

Der Begriff „Computer-Aided Design“ entstand Ende der 1950er-Jahre im Zuge der Entwicklung des Programmiersystems [APT](#), welches der rechnerunterstützten Programmierung von NC-Maschinen (NC = Numerical Control bzw. Numerischer Steuerung, siehe [Digitale Ausgabe](#)) diente.^{[10][1]}

CAD (von engl. *computer-aided design*, zu Deutsch **rechnerunterstütztes Konstruieren**^{[1][2]}) bezeichnet die Unterstützung von [konstruktiven Aufgaben](#) mittels [EDV](#) ((Elektronischer Datenverarbeitung) zur Herstellung eines Produkts (Beispielsweise Auto, Flugzeug, Bauwerk, Kleidung). In einem engeren Sinn versteht man unter CAD das rechnerunterstützte Erzeugen und Ändern des geometrischen Modells. In einem weiteren Sinn versteht man darunter sämtliche rechnerunterstützten Tätigkeiten in einem Konstruktionsprozess, einschließlich der geometrischen Modellierung, des Berechnens, des Simulierens und sonstiger Informationsgewinnung und Informationsbereitstellung, von der Konzeptentwicklung bis zur Übergabe an die Herstellung bzw. Fertigung ([Arbeitsvorbereitung](#)).^{[3][4]}

Am MIT zeigte [Ivan Sutherland](#) 1963 mit seiner [Sketchpad](#)-Entwicklung, dass es möglich ist, an einem computergesteuerten Radarschirm interaktiv (Lichtstift, Tastatur) einfache Skizzen (englisch *Sketch*) zu erstellen und zu verändern.

1965 wurden bei Lockheed (Flugzeugbau, USA) die ersten Anläufe für ein kommerzielles CAD-System zur Erstellung technischer Zeichnungen (2D) gestartet. Dieses System, [CADAM](#) (Computer-augmented Design and Manufacturing), basierend auf [IBM-Großrechnern](#) und speziellen Bildschirmen, und mit hohen Kosten verbunden, wurde später von IBM vermarktet und war, zumindest im Flugzeugbau, Marktführer bis in die 1980er-Jahre. Der französische [Flugzeughersteller](#) Avions Marcel Dassault (heute [Dassault Aviation](#)) entwickelte ein Grafikprogramm zur

Erstellung von Zeichnungen, woraus das Programm CATIA entstand. Die [Mirage](#) war das erste [Flugzeug](#), das damit entwickelt wurde. Damals benötigte ein solches Programm noch die Leistung eines Großrechners.

Nachdem in der zweiten Hälfte der 1980er-Jahre die ersten [Personal Computer \(PC\)](#) in den Unternehmen eingeführt wurden, kamen auch CAD-Programme dafür auf den Markt. In dieser Zeit gab es eine Vielzahl von Computerherstellern und Betriebssystemen. [AutoCAD](#) war eines der ersten und erfolgreichsten CAD-Systeme, das auf unterschiedlichen Betriebssystemen arbeitete. Um den Datenaustausch zwischen diesen Systemen zu ermöglichen, definierte Autodesk für sein CAD-System AutoCAD das [DXF](#)-Dateiformat als „neutrale“ Export- und Import-schnittstelle. 1982 erschien AutoCAD für das Betriebssystem [DOS](#). Das Vorgehen bei der Konstruktion blieb jedoch beinahe gleich wie zuvor mit dem Zeichenbrett.

Der Vorteil von 2D-CAD waren sehr saubere Zeichnungen, die einfach wieder geändert werden konnten. Auch war es schneller möglich, verschiedene Versionen eines Bauteils zu zeichnen.

Um 1974 wurden [B-Spline-Kurven und -Flächen](#) für das CAD eingeführt. ^[102]

In den 1980er-Jahren begann wegen der sinkenden Arbeitsplatzkosten und der besser werden-Software ein CAD-Boom. In der Industrie wurde die Hoffnung gehegt, mit einem System alle anstehenden Zeichnungs- und Konstruktionsaufgaben lösen zu können. Dieser Ansatz ist aber gescheitert. Heute wird für jede spezielle Planungsaufgabe ein spezielles System mit sehr leistungsfähigen Spezialfunktionen benutzt. Der Schritt zur dritten Dimension wurde durch die immer höhere Leistungsfähigkeit der [Hardware](#) dann gegen Ende der 1980er-Jahre auch für kleinere Unternehmen erschwinglich. So konnten nun virtuelle Körper von allen Seiten begutachtet werden. Ebenso wurde es möglich, Belastungen zu simulieren und Fertigungsprogramme für computergesteuerte Werkzeugmaschinen ([CNC](#)) abzuleiten.

Seit Anfang der 2000er-Jahre gibt es erste Ansätze, die bis dahin immer noch zwingend notwendige Zeichnung verschwinden zu lassen. In die immer öfter vorhandenen 3D-Modelle werden von der Bemaßung über Farbe und Werkstoff alle notwendigen Angaben für die Fertigung eingebracht. Wird das 3D-Modell um diese zusätzlichen, geometriefremden Eigenschaften erweitert, wird es zum [Produktmodell](#), unterstützt beispielsweise durch das [STEP-Datenformat](#). Die einzelnen einheitlichen Volumenobjekte werden zu Instanzen unterschiedlicher Klassen. Dadurch können Konstruktionsregeln und Verweise zwischen einzelnen Objekten (zum Beispiel: Fenster wird in Wand verankert) realisiert werden.

Meilensteine

1957 PRONTO war das erste kommerzielle, numerisch-gesteuerte Programmiersystem. Es legte den Grundstein für das gesamte CAD.

1959 CAD – der Begriff wurde von Douglas T. Ross 1959 geprägt; als MIT Forscher entwickelte er CAD zur Erstellung von Stromkreis-Diagrammen.

1963 Sketchpad war eines der ersten Designsysteme, das, mit Lichtstift und CRT Bildschirm, eine **grafische Benutzeroberfläche** verwendete.

1967 DIGIGRAPHICS war eines der **ersten kommerziellen CAD-Systeme** mit grafischen Schnittstellen, das zu einem Preis von \$500.000 auf den Markt kam.

1971 ADAM (Automated Drafting and Machining) ein von Patrick Hanratty so entwickeltes **interaktives Grafikdesign-, Entwurfs- und Fertigungssystem**, dass es zur damaligen Zeit auf fast allen Grossrechnern funktionierte.

1975 Nurbs: während seiner Arbeit bei Computervision, führte Dr. Ken Versprille NURBS in CAD ein. NURBS halfen Oberflächen zu definieren und sind auch heute noch weit in der Ingenieurswelt verbreitet.

1977 Glide von Charles Eastman hatte viele der gleichen Funktionen wie das moderne **BIM**

1982 AUTODESK AUTOCAD war die **erste 2D CAD Software für PCs**, um präzise 2D- und 3D-Zeichnungen zu erstellen. Von diesem Zeitpunkt an wurde CAD mehr erschwinglich.

1994 STEP (Standard for the Exchange of Product model data) wurde von der Internationalen Organisation für Standardisierung als das neue **3D-CAD-Dateisystem** entwickelt und ist bis heute das am meisten genutzte Format.

2002 Catia Digital Project: Digital Project, Catia, Gehry Guggenheim Bilbao 1997

1997 Alibre arbeitete eng mit Microsoft zusammen, um 1998 **die erste web-basierte kollaborative 3D-Designsoftware** zu entwickeln.

2.3 – Anatomie einer CAD Software

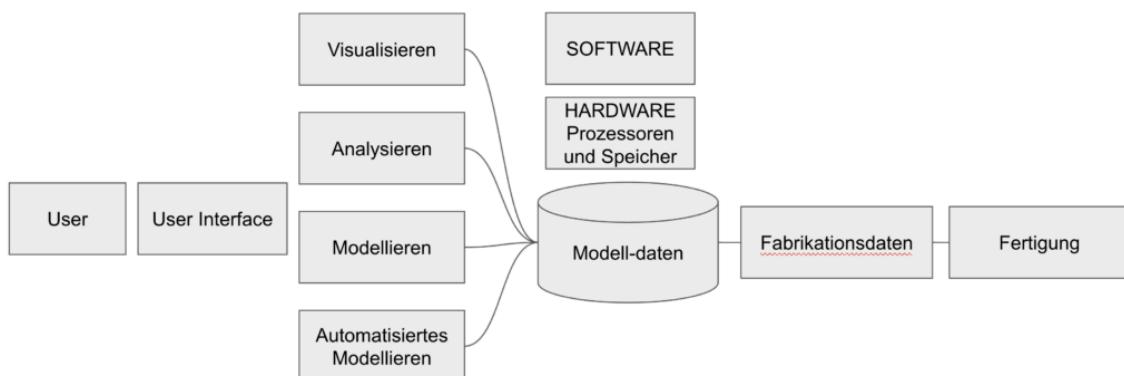


Abbildung 2.1 – Komponenten einer CAD Software

2.3.1 – Direkte Modellierung

Bei der direkten Modellierung (*explicit modeling*) werden die geometrischen Elemente direkt über bestimmte Funktionen (Skalieren, Verschieben, Dehnen etc.) verändert. Die geometrischen Elemente enthalten nur feste Werte (und keine Variablen), welche erst durch das Anwenden von Funktionen geändert werden können. Hierbei wählt man das geometrische Element und die entsprechende Funktion, welche die gewünschte Änderung hervorrufen soll, aus und verändert das Objekt entweder interaktiv mit der Maus oder über Koordinateneingabe mittels Tastatur. Verändert werden dabei nur die gewählten geometrischen Elemente. Es bestehen (im Unterschied zum parametrischen Modellieren) keinerlei dauerhaften Abhängigkeiten zwischen den geometrischen Elementen, wodurch ein sehr intuitives und freies Ändern der Geometrie möglich ist.

2.3.2 – Parametrische Modellierung

Unter parametrischer Modellierung versteht man das Steuern des Modells mittels [Parametern](#). Das heißt, dass – anders als bei der direkten Modellierung – das Modell nicht direkt über seine Geometrie, sondern über seine Parameter angesprochen wird, welche das Modell jederzeit ändern können. Die Arten der Parameter unterscheiden sich je nach Anwendung grob in Geometrieparameter (zum Beispiel geometrische Maße, Positionen), physikalische Parameter (Beispielsweise Werkstoffe, Lasten), Topologieparameter und Prozessparameter.^{[62][63]} Dadurch dass diese Parameter systemintern gespeichert werden, lassen sich Beziehungen und Abhängigkeiten zwischen ihnen herstellen. Dies wird über Restriktion bzw. Zwangsbedingungen (*Constraints*) umgesetzt.

2.3.3 – Chronologie-basierte Modellierung

Hier kommen Datenstrukturen zur Anwendung, die den Erzeugungsverlauf des Modells aufzeichnen. Für den Benutzer wird dies in einem Chronologiebaum (*History tree*) dargestellt, der während der Modellierung laufend aktualisiert wird und in dem die einzelnen Modellerschritte und der Aufbau des Modells eingesehen und bei Bedarf in jeder Phase des Konstruktionsprozesses verändert werden können.^{[72][73]}

2.4 – Rhino als Beispiel

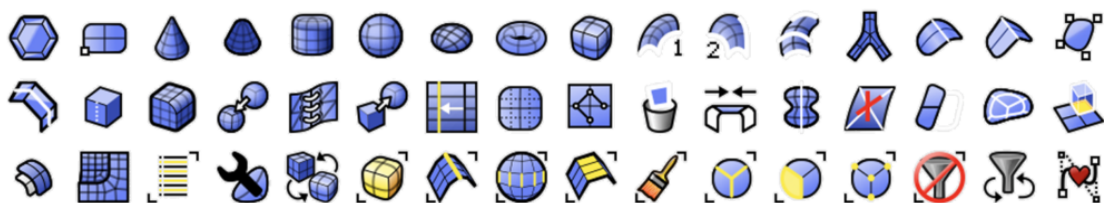
Rhino kann NURBS-Kurven, -Flächen und -Volumenkörper, SubD-Geometrie, Punktwolken und Polygonnetze erzeugen, bearbeiten, analysieren, dokumentieren, rendern, animieren und übersetzen.

CAD Software erlaubt es Modelle zu:

- erstellen
- aufnehmen/scannen
- editieren
- visualisieren
- dokumentieren in Form von Fertigungs-/Herstellungsunterlagen
- analysieren
- Fabrikationsdaten erstellen

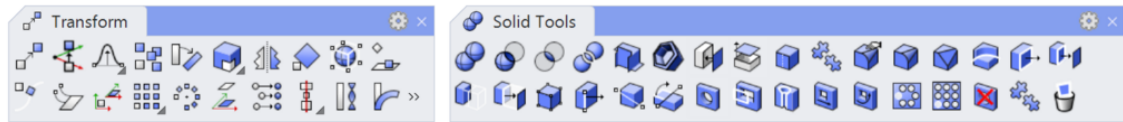
Rhino erlaubt grundsätzlich direkte Modellierung. Durch Grasshopper lassen sich aber Strategien des parametrischen und chronologie-basierten Modellierens verwenden.

2.4.1 – Modellierungs-Werkzeuge



Rhino verfügt über Modellierungswerkzeuge, die das Erstellen von Punkten, Kurven, Flächen, Volumenkörpern und Polygonnetzen erlauben.

2.4.2 – Editieren



Diese Geometrien lassen sich mit unter anderem mit allgemeinen Werkzeugen, Transformierung und Kontrollpunkten bearbeiten.

2.4.3 – Interface



Das Interface bietet eine differenzierte Benutzeroberfläche, mit Konstruktionshilfen. Die Modelle können auf unterschiedlichster Art und in verschiedenen Perspektiven in Echtzeit angezeigt werden, bzw. realistisch gerendert werden.

2.4.4 – Entwurf / Pläne



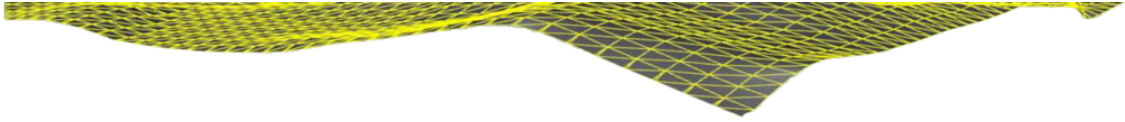
Für das Erstellen von Plänen gibt es verschiedene technische Illustrations und Bemessungswerkzeuge.

2.4.5 – Digitale Herstellung



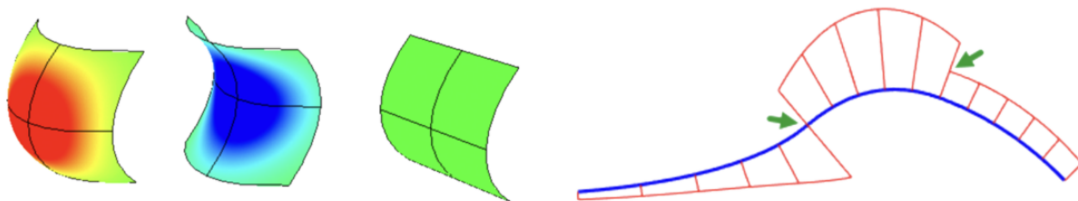
Das Rhino-Entwicklungsprojekt begann vor ungefähr 20 Jahren, um Schiffdesigner mit Werkzeugen für die Erzeugung von Computermodellen auszustatten, die dazu verwendet werden konnten, die digital gesteuerten Fertigungsgeräte in den Schiffswerften zu steuern. Bei rascher Kostensenkung der digitalen Fertigung und 3D-Drucktechnologie haben immer mehr Designer direkten Zugriff auf digitale 3D-Fertigungsgeräte, für die Rhino die Fertigungsdaten exportieren kann.

2.4.6 – Erfassung



Die Erfassung von 3D-Daten ist in einem Designprojekt oft einer der ersten Schritte. Rhino unterstützt 3D-Digitalisierungs-Hardware wie auch gescannte 3D-Punktwolkendaten.

2.4.7 – Untersuchung und Analyse



Die Verwirklichung eines Projekts erfordert hochwertige 3D-Modelle in jeder Phase des Designs, der Präsentation, der Analyse und der Fertigung.

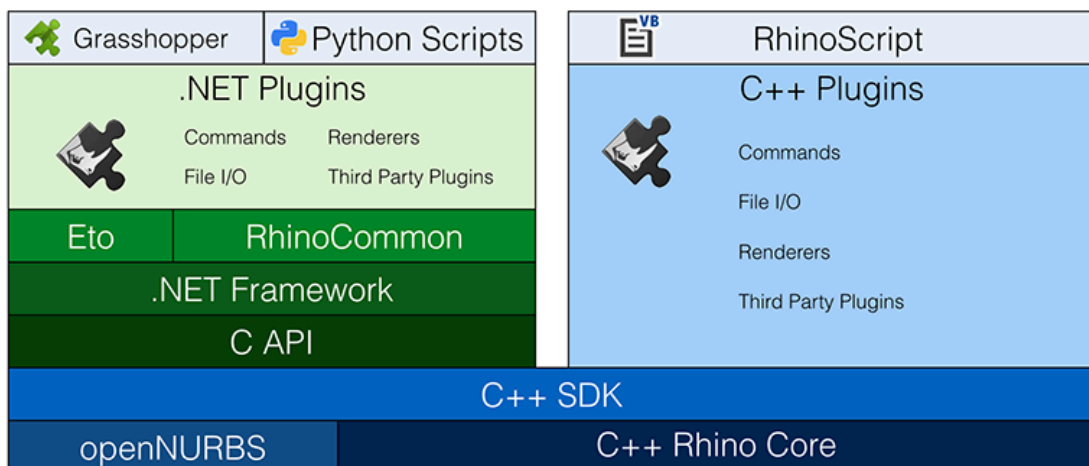
2.4.8 – Export

.3dm .3ds .sat .ai .amf .dwg .dxf .cd .iges
.sat .dgn .m .pdf .ply .sldprt .x .pts .svg .dae
.skp .stl .step .stp .obj .rib .xgl .x_t .zpr .fbx

Um die 3D-Modelle oder die 2D-Zeichnungen in anderen Programmen zu öffnen oder weiterzubearbeiten, erlaubt es Rhino, die Projekte in einer Vielzahl an Datenformaten exportieren zu können.

2.4.9 – API

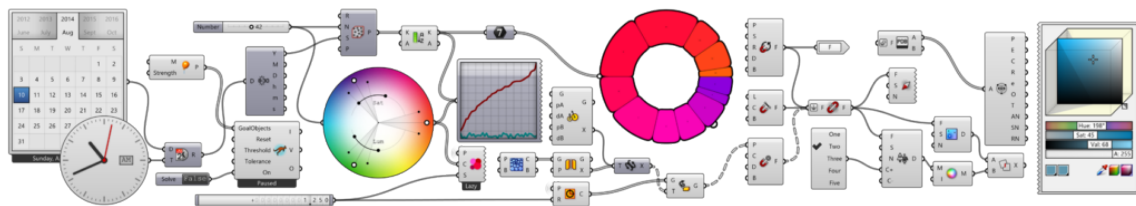
Eine Programmierschnittstelle, häufig nur kurz API genannt, ist ein Programmteil, der von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung gestellt wird.



2.5 – Grasshopper als Beispiel für Visuelle Programmierung

Der sogenannte Grasshopper ist ein Plug-in, das in Rhino enthalten ist. Dabei handelt es sich um ein Werkzeug für die algorithmische Modellierung, das speziell für die Gestaltung und Bearbeitung komplexer Formen durch bestimmte Parameter verwendet wird. Da sich Formen aus Daten erstellen lassen, ist einer der Hauptvorteile von Grasshopper die Möglichkeit, die Geometrie fast unendlich zu verändern, indem man ganz einfach die Parameterwerte ändert. Außerdem können Änderungen vorgenommen werden, ohne den Entwurf abbrechen oder neu starten zu müssen.

Als **Visuelle Programmiersprache** (englisch visual programming language, **VPL**, auch *grafische Programmiersprache*) bezeichnet man eine **Programmiersprache**, in der ein **Programm**, **Algorithmus** oder **Systemverhalten** durch grafische Elemente und deren Anordnung definiert wird. In der Computertechnik ist eine visuelle Programmiersprache jede Programmiersprache, mit der Benutzer Programme erstellen können, indem sie Programmelemente grafisch manipulieren, anstatt sie textuell anzugeben. Beispielsweise basieren viele VPLs (bekannt als Datenfluss- oder Diagrammprogrammierung) auf der Idee von „Boxen und Pfeilen“, wobei Boxen oder andere Bildschirmobjekte als Einheiten behandelt werden, die durch Pfeile, Linien oder Bögen verbunden sind, welche Relationen darstellen.



2.6 – Bildtafeln



Abbildung 2.13 – Hotel Waldhaus Gasterntal: Blaupause 1902

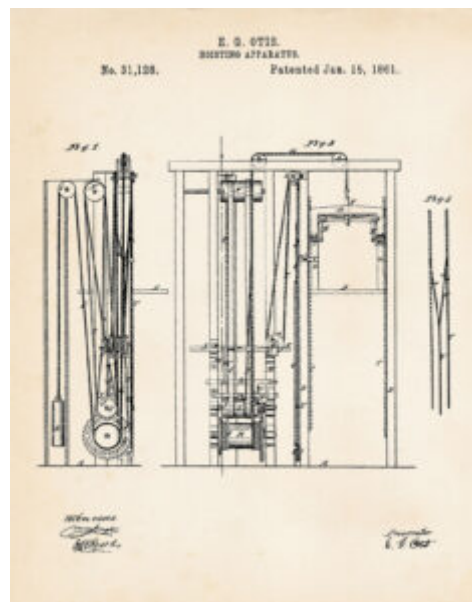


Abbildung 2.14 – Patentzeichnung von Elisha Graves Otis



Abbildung 2.15 – Zeicheninstrumente



Abbildung 2.16 - Architekt beim technischen Zeichnen (Holzschnitt, 1893)



Abbildung 2.17 - Ivan Sutherland demonstrieret Sketchpad.



Abbildung 2.18 – Vera Molnar: Algorithmische Zeichnungen, von 1959 an (u.a. ein Prozent Unordnung)

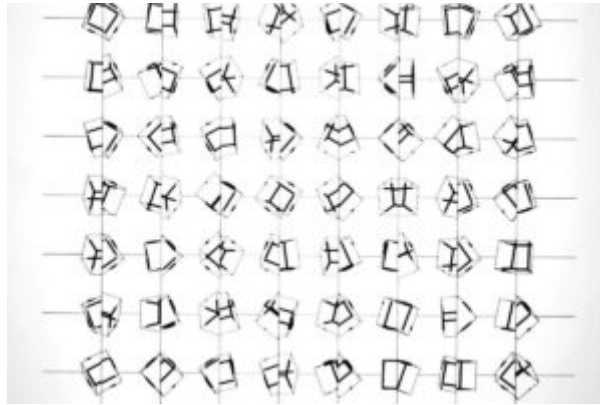


Abbildung 2.19 – Manfred Mohr „Genauigkeit wird zum Gestaltungsmittel.“



Abbildung 2.20 – Digital Project ist eine CAD-Anwendung, die auf CATIA V5 basiert und von Gehry Technologies entwickelt wurde, einem Technologieunternehmen im Besitz des Architekten Frank Gehry. Zu den Änderungen, die Gehry Technologies an CATIA vorgenommen hat, gehört eine neue visuelle Schnittstelle, die für die Arbeit in der Architektur geeignet ist. Digital Project ist weithin als die Software bekannt, mit der das Guggenheim-Museum Bilbao entworfen wurde, das pünktlich und innerhalb des Budgets fertig gestellt wurde, sowie die Louis Vuitton Foundation, deren Kosten das Zehnfache der ursprünglichen Prognosen betrugen und deren Zeitplan überschritten wurde.



Abbildung 2.21 - Foster & Partners Headquarters Riverside Apartments and Studio

Mensch-Computer-Interaktion

3.1 – Mensch – Maschine Interaktion

„A display connected to a digital computer gives us a chance to gain familiarity with concepts not realizable in the physical world. It is a looking glass into a mathematical wonderland.“

Sutherland, Ivan E. (1965). „The Ultimate Display“. Proceedings of IFIP Congress. pp. 506-508.

Mensch-Computer-Interaktion (häufig als **HCI** abgekürzt, [englisch](#) Human-computer interaction) erforscht das Design und die Verwendung von [Computer](#)-Technologie an der Schnittstelle zwischen Menschen (Anwendern) und Computern. Forscher auf dem Gebiet der HCI beschäftigen sich mit der Art und Weise, wie Menschen mit Computern und Design-Technologien interagieren. Dabei werden neben Erkenntnissen der [Informatik](#) auch solche aus der [Psychologie](#) (vor allem der [Medienpsychologie](#)), der [Arbeitswissenschaft](#), der [Kognitionswissenschaft](#), der [Ergonomie](#), der [Soziologie](#) und dem [Design](#) herangezogen.

Der Benutzer interagiert direkt mit Hardware für die menschliche Eingabe und Ausgabe, wie z. B. Anzeigen, z. über eine grafische Benutzeroberfläche. Der Benutzer interagiert mit dem Computer über diese Softwareschnittstelle unter Verwendung der gegebenen Eingabe- und Ausgabe(E/A)-Hardware. Software und Hardware sind so aufeinander abgestimmt, dass die Verarbeitung der Benutzereingaben schnell genug ist und die Latenz (Verzögerung) der Computerausgabe den Arbeitsablauf nicht stört.

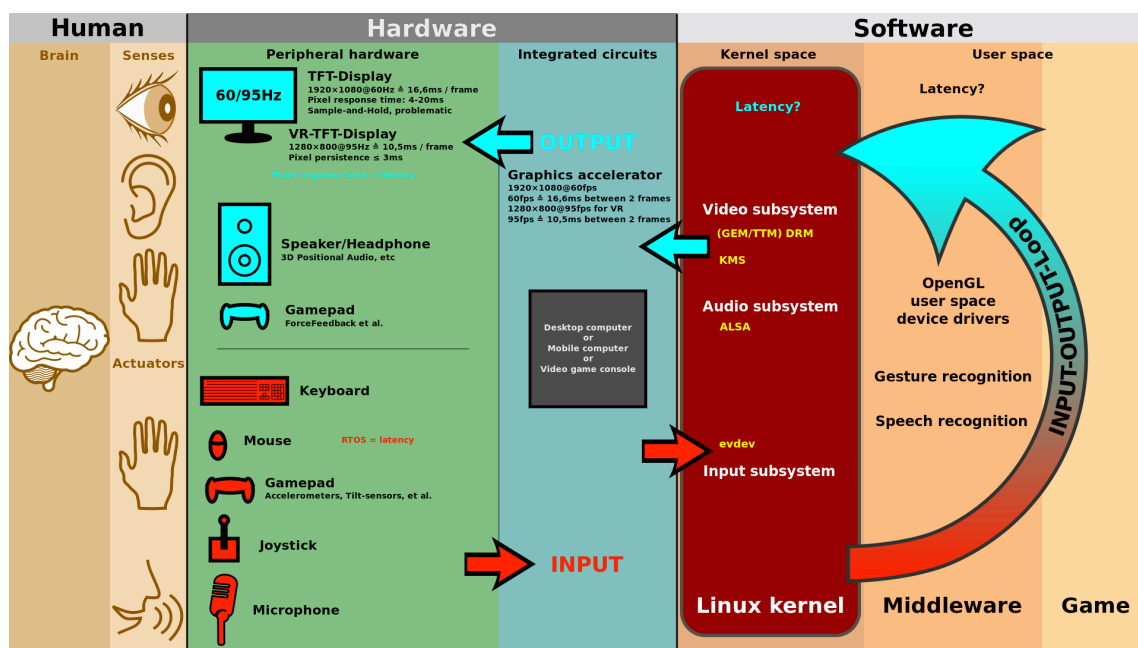


Abbildung 3.1 – Mensch, Hardware und Software Schnittstelle

3.2 – Mixed Reality

Unter **Mixed reality**, **Vermischte Realität** oder **Gemischte Realität** werden Umgebungen oder Systeme zusammengefasst, die die natürliche Wahrnehmung eines Nutzers mit einer künstlichen (computererzeugten) Wahrnehmung vermischen. Neben der hauptsächlich computererzeugten **virtuellen Realität** sind dies insbesondere Systeme der **erweiterten Realität** und der erweiterten Virtualität.

Mixed reality umfasst das gesamte „Realitäts-Virtualitäts-Kontinuum“^[1] mit Ausnahme von *nur Realität* und *nur Virtualität*: Zwischen den beiden Extremen *nur Realität* und *nur Virtualität* gibt es stufenlos Zwischenstadien, die beide vermischen. Insbesondere sind damit *Erweiterte Realität* und *Erweiterte Virtualität* spezielle Ausprägungen des Prinzips *mixed reality*. Am besten veranschaulicht man sich diese abstrakten Begriffe an konkreten Beispielen:

- **Reine Realität** ist ein Mensch, der zum Einkaufen in den Supermarkt geht.
- **Erweiterte Realität** sind z. B. Brillengläser, auf deren Innenseite ein Computer den **Einkaufszettel** des Benutzers projiziert; beispielsweise so, dass beim Benutzer der Eindruck entsteht, der Einkaufszettel sei an die Wand des Supermarktes angeschrieben. Die Wirklichkeit wird hier um virtuelle Informationen angereichert.
- **Erweiterte Virtualität** bezieht sich auf Anwendungen, welche über eine VR-Brille/ ein **HMD** erlebt werden und die z. B. den Ton einer Türsprechanlage auf die Kopfhörer überträgt, wenn es an der Tür klingelt. Die Virtualität wird hier um reale Informationen angereichert.
- **Virtuelle Realität** bezieht sich auf Anwendungen, welche über eine VR-Brille/ ein **HMD** erlebt werden und rein virtuell existieren.

3.3 – Internet of Things

Das **Internet der Dinge** (IdD) (auch: „Allesnetz“; englisch Internet of Things, Kurzform: IoT) ist ein Sammelbegriff für Technologien einer globalen Infrastruktur der Informationsgesellschaften, die es ermöglicht, physische und virtuelle Objekte miteinander zu vernetzen und sie durch Informations- und Kommunikationstechniken zusammenarbeiten zu lassen.

Mit Technologien des „Internets der Dinge“ implementierte Funktionen erlauben die Interaktion zwischen Mensch und hierüber vernetzten beliebigen elektronischen Systemen sowie zwischen den Systemen an sich. Sie können darüber hinaus auch den Menschen bei seinen Tätigkeiten unterstützen. Die immer kleineren eingebetteten Computer sollen Menschen unterstützen, ohne abzulenken oder überhaupt aufzufallen. So werden z. B. miniaturisierte Computer, sogenannte Wearables, mit unterschiedlichen Sensoren direkt in Kleidungsstücke eingearbeitet.

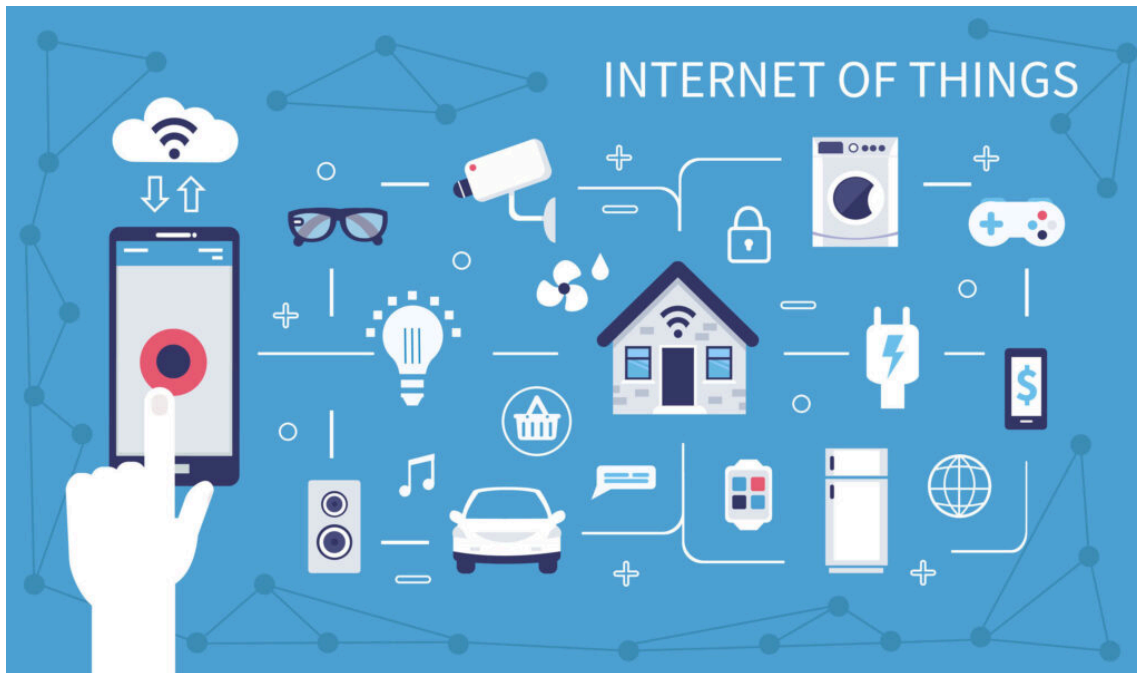


Abbildung 3.2 – Internet of Things Analytics

Technology roadmap: The internet of things

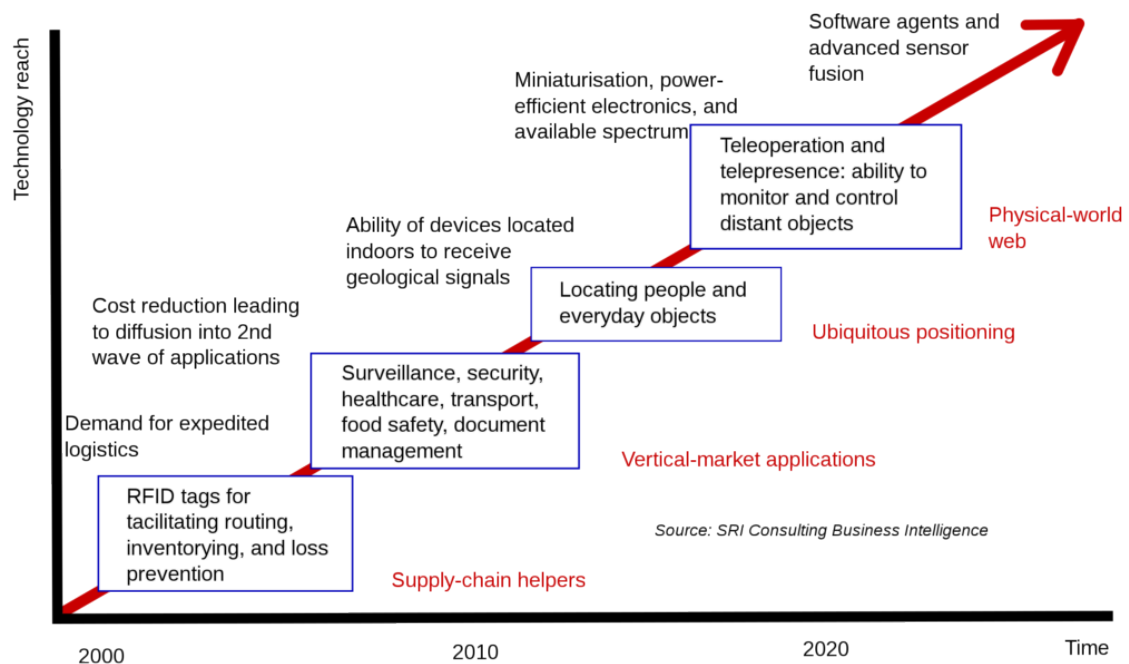


Abbildung 3.3 – A technology roadmap of the Internet of Things. SRI Consulting Business Intelligence/National Intelligence Council – Appendix F of Disruptive Technologies Global Trends 2025 page 1 Figure 15 (Background: The Internet of Things)

3.4 – Bildtafeln



Abbildung 3.4 – William English, Prototyp der ersten Computer Maus, 1964



Abbildung 3.5 – Digitaler Stift, Apple Pencil



Abbildung 3.6 – VR (Virtual Reality) Brille, Oculus quest 2



Abbildung 3.7 – AR (Augmented Reality) – Brille, Hololens 2

Digitale Geometrie

Darstellende Geometrie

Geometrie steckt im Kern des architektonischen Entwurfsprozesses. Sie ist omnipräsent, von den ersten Schritten der Formfindung bis zur eigentlichen Konstruktion. Moderne Konstruktionsgeometrie bietet eine Reihe von Werkzeugen für effizientes Entwerfen, Analyse und die Herstellung komplexer Formen. Das führt zu neuen Herausforderungen, sowohl in der Architektur als auch in der Geometrie. Architekturgeometrie ist daher als integrative Forschungsaufgabe zu sehen, die an der Grenze zwischen angewandter Geometrie und Architektur angesiedelt ist.

*Geometry has an ambiguous reputation, associated as much with idiocy as with cleverness.
At best there is something desperately uncommunicative about it, something more than a little removed from the rest of experience to set against its giant claim of truth. Flaubert, in The Dictionary of Accepted Ideas, defines a geometrician as „travelling on strange seas of thought-alone.“
And when Joseph Conrad wished to characterize the futile effort of concentration made by the earnest but mentally retarded youth Stevie in The Secret Agent, he would describe him as „seated very good and quiet at a deal table, drawing circles, circles, circles; innumerable circles, concentric, eccentric, a coruscating whirl of circles that by their tangled multitude of repeated curves, uniformity of form, and confusion of intersecting lines suggested a rendering of cosmic chaos, the symbolism of a mad art attempting the inconceivable.“ – Robin Evans, The projective cast*

4.1 – Entwicklung der Darstellende Geometrie

Darstellende Geometrie ist der Teilbereich der [Geometrie](#), der sich mit den geometrisch-konstruktiven Verfahren von Projektionen dreidimensionaler Objekte auf eine zweidimensionale Darstellungsebene befasst. Die Anwendungsbereiche ihrer Methoden sind breit gefächert und erstrecken sich neben den heute bekanntesten Anwendungen in der Technik- und Architekturdarstellung auch auf Kunst, Malerei, Kartenwesen und Computergraphik.

Bei der systematischen Errichtung von Bauwerken spielen Pläne mit konkreten Vorgaben eine wichtige Rolle. Schon im Altertum wurden *Grund- und Aufrisse* verwendet. Der älteste schriftliche Beleg dafür ist das Werk [Zehn Bücher über Architektur](#) des römischen Baumeisters [Vitruvius](#). Aber erst [Albrecht Dürer](#) (1471–1528) schrieb in der Frühen Neuzeit das erste wirkliche Lehrbuch der Darstellenden Geometrie: *Underweysung mit dem Zirkel und Richtscheidt*^[2] (Nürnberg 1525). [Gaspard Monge](#) (1746–1818) führte in seinem Buch *Geometrie descriptive*^[3] zum ersten Mal die strenge Zuordnung von Grund- und Aufriss ein, um räumliche Probleme zeichnerisch zu lösen. Die Grundaufgaben der Darstellenden Geometrie sind dort schon in der noch heute gebräuchlichen Fassung zu finden.

4.1.1 – Die Trennung von Planung und Ausführung

Alberti

Auffallend an der Architekturtheorie Albertis ist vor allem ihre erstaunliche Modernität. Es geht um die neue Rolle des Architekten als reiner Planer mit eigenem, nicht mehr handwerklich geprägtem Ausbildungsgang. ¹ Der Architekt, den sich Alberti vorstellte, sollte sich in der Malerei und in der Mathematik auskennen. Dies hatte zwar schon Vitruv gefordert, aber Alberti bezog sich hier auf das Kernstück seiner Abhandlung über die Malerei, nämlich auf die **Zentralperspektive**, die für den Architekten genauso wichtig sei wie für den Maler, da er mit ihrer Hilfe die Wirkung des Gebäudes abschätzen und berechnen könne.

Allerdings handelt es sich bei dem von Alberti anvisierten Architekten um eine andere *species* als sie aus der baumeisterlichen Praxis des Mittelalters bekannt war. Albertis Architekt ist kein Praktiker, sondern ein Entwerfer, der sich als Gelehrter versteht. Seine Bemerkungen über seine eigene Stellung dokumentieren, dass er im *architectus* eine intellektuelle Autorität sah, der für die Ausführung ein Baumeister (*faber*) zur Seite stand, ähnlich wie man dies in Andrea Pisanos Reliefs am Campanile des Florentiner Doms sehen kann, in denen der Architekt in vornehmer Kleidung in seinem Arbeitszimmer (*studiolo*) sitzt, während der Baumeister die Vorgänge auf der Baustelle unter Kontrolle hält. Diesem Muster war das Verhältnis zwischen Architekt und Baumeister verpflichtet, das Albertis eigene Bauvorhaben kennzeichnet. ²

In den frühesten Zeichnungen wurde die anschauliche **perspektivische (dreidimensionale)** Darstellung – anfänglich in primitiver und ab der **Renaissance** in korrekter Weise – benutzt. ^[1] Weil sie relativ hohe Anforderungen an die Begabung der Zeichner stellt, ging man bei Beginn der **Industrialisierung** (18. Jahrhundert) auf zweidimensionale Abbildungen der Objekte in rechtwinkliger Parallelprojektion (**Orthogonalprojektion**) über. Die **Kanten** und **Symmetrie**-Linien der meisten technischen Gegenstände sind untereinander rechtwinklig angeordnet. Man kann ihnen ein rechtwinkliges Koordinatensystem überlagern und von ihnen Ansichten (Vorder-/Rückansicht, Seitenansicht von rechts/links und Drauf-/Untersicht) in je einer Richtung der Koordinatenachsen erzeugen. Die in den drei Koordinatenebenen liegenden Flächen werden unverzerrt abgebildet. Bei der angewendeten Parallelprojektion ist der Abbildungsmaßstab unabhängig davon, welchen Abstand eine Objektebene vom Betrachter hat; alle Längen sind maßstabgleich dargestellt. Nachteilig ist, dass sich der Betrachter die Geometrie des Körpers aus mehreren Ansichten (und gegebenenfalls Schnitten bei Körpern mit Hohlräumen) vorstellen muss (setzt räumliches Vorstellungsvermögen voraus). ³

4.2 – Projektionen

Vor der Einführung von Computergrafik (engl. Computer-graphics) war Wissen um Projektionstechniken nötig. Heute wird das von Computern in Echtzeit durchgeführt. Um die digitalen Techniken möglichst gut zur Visualisierung nutzen zu können, ist dennoch ein grundlegendes Verständnis nötig. (**Helmut Pottmann**, Architectural Geometry). In der Darstellenden Geometrie bedient man sich im Wesentlichen zweier Abbildungsverfahren. Dabei werden Punkte und Kurven eines Objektes mit Hilfe von Strahlen (Geraden) auf eine Bildtafel (Ebene) projiziert:

4.2.1 – Parallelprojektion

Die Abbildungsstrahlen sind parallel, wie z. B. beim Sonnenlicht. Dabei unterscheidet man noch die beiden Fälle:

- Die Strahlen stehen *senkrecht* auf der Bildtafel (*senkrechte*)

1.
quelle
2.
quelle
3.
Quelle

- *Parallelprojektion* oder *Orthogonalprojektion* oder *Normalprojektion*.
 Die Strahlen stehen *nicht senkrecht* zur Bildtafel (*schiefe* oder *schräge Parallelprojektion*).

Parallelprojektionen werden wegen ihrer *Teilverhältnistreue* (Teilverhältnisse auf Geraden bleiben invariant) gerne von Ingenieuren verwendet. Der Spezialfall *Vogelperspektive* ist eine schiefe Parallelprojektion, die insbesondere zur Veranschaulichung von Stadtplänen verwendet wird. Sie lässt sich relativ einfach von Hand herstellen. Parallelprojektionen lassen sich schnell als *axonometrische Bilder* oder bei umfangreicheren Objekten mit Hilfe des *Einschneideverfahrens* herstellen. Für fast alle Konstruktionen in der Darstellenden Geometrie verwendet man Grund- und Aufriss eines Objektes. Das sind senkrechte Parallelprojektionen auf eine horizontale (Grundriss) bzw. senkrechte Ebene (Aufriss) (s. *Zweitafelprojektion*). Durch sie ist (mit den entsprechenden Bezeichnungen) ein Objekt räumlich eindeutig beschrieben.

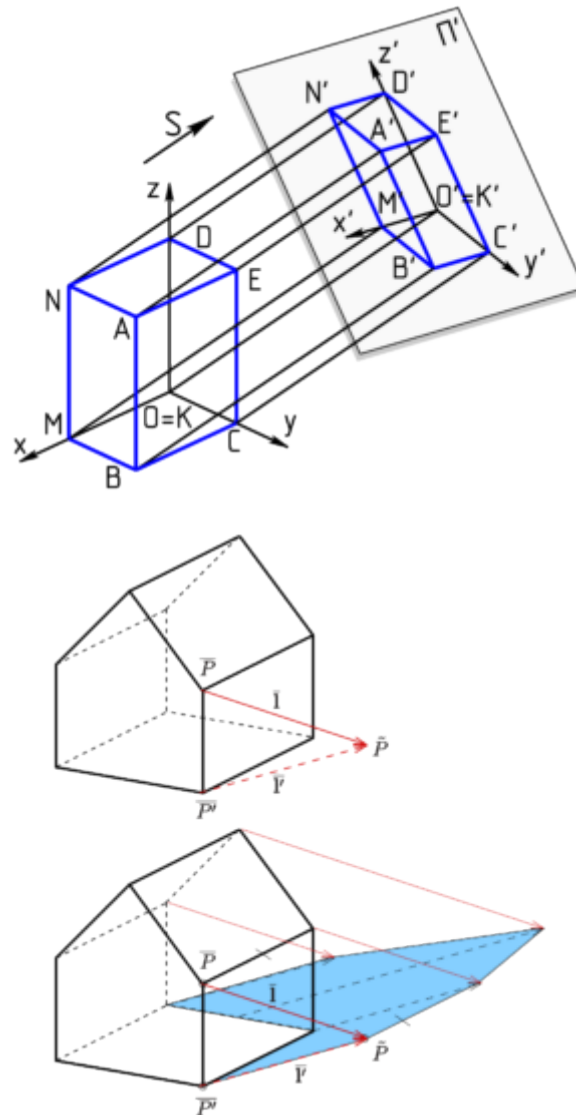


Abbildung 4.2 – Parallelprojektion

4.2.2 – Zentralprojektion

Die Grundlagen der *Zentralprojektion* waren schon den Griechen und Römern bekannt. Aber erst in der *Renaissance* wurde diese Art der Darstellung räumlicher Gegebenheiten durch die Malerei wiederentdeckt und zur Blüte weiterentwickelt. Siehe hierzu *De pictura* von *Leon Battista Alberti* (1404). Die Meister dieser Zeit waren *Albrecht Dürer* (1471–1528), *Leonardo da Vinci* (1452–1519) und *Michelangelo* (1475–1564). Alle Abbildungsstrahlen gehen durch einen Punkt, dem Projektionszentrum oder *Augpunkt*. Bei Parallelprojektion sind die Bilder paralleler Geraden i. A. wieder parallel. Bei *Zentralprojektionen* schneiden sich die Bilder paralleler Geraden i. A. in einem Punkt, dem *Fluchtpunkt* des Parallelbüschels.

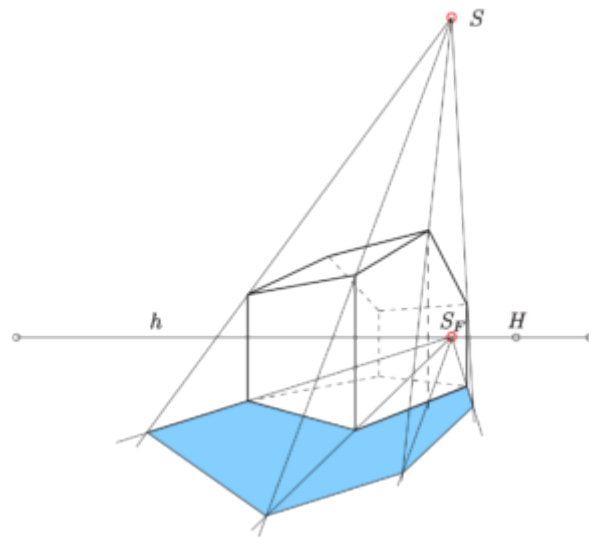


Abbildung 4.3 – Zentralprojektion



Ein YouTube Element wurde aus dieser Textversion entfernt. Sie können es hier online ansehen: <https://wp-prd.let.ethz.ch/computationaldesigniii/?p=21>

4.3 – Koordinatensysteme

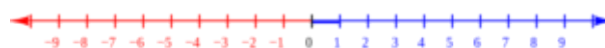


Abbildung 4.4 – 1-Dimensionales Koordinatensystem

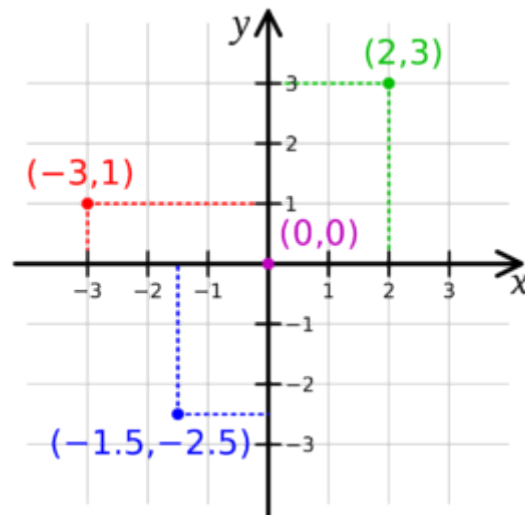


Abbildung 4.5 – 2-Dimensionales Koordinatensystem

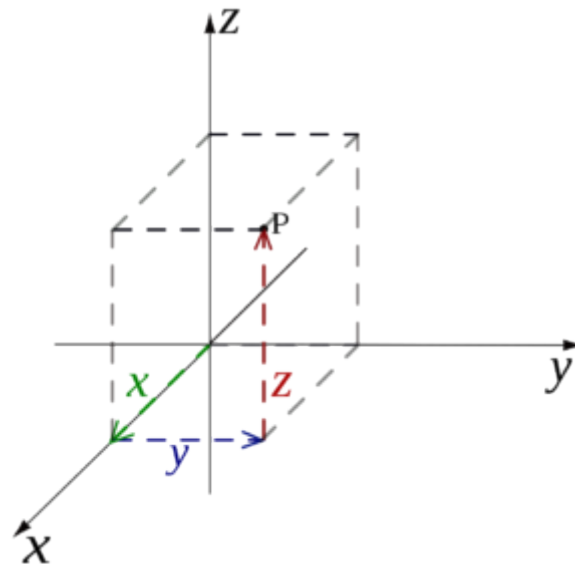


Abbildung 4.6 – 3-Dimensionales kartesisches Koordinatensystem

Eines der bekanntesten Beispiele für ein Koordinatensystem ist das kartesische Koordinatensystem. In der Ebene werden zwei **Senkrechte** gewählt und die Koordinaten eines Punktes als die vorzeichenbehafteten Abstände zu den Geraden aufgefasst. In drei Dimensionen wählt man drei zueinander **orthogonale** Ebenen und die drei Koordinaten eines Punktes sind die vorzeichenbehafteten Abstände zu jeder der Ebenen. Es ist nach dem **latinisierten** Namen *Cartesius* des französischen Mathematikers **René Descartes** benannt, der das Konzept der „kartesischen Koordinaten“ bekannt gemacht hat.^[3] Je nach Anordnung der Koordinatenachsen kann das dreidimensionale Koordinatensystem ein **Rechts-** oder ein Linkssystem sein.

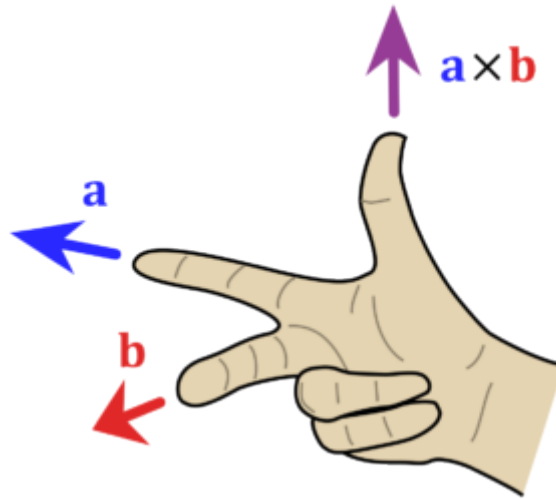


Abbildung 4.7 - In Rhinos rechtshändig ausgelegtem System wird die Richtung von $a \times b$ durch die Dreifingerregel bestimmt (a = Zeigefinger, b = Mittelfinger und $a \times b$ = Daumen).

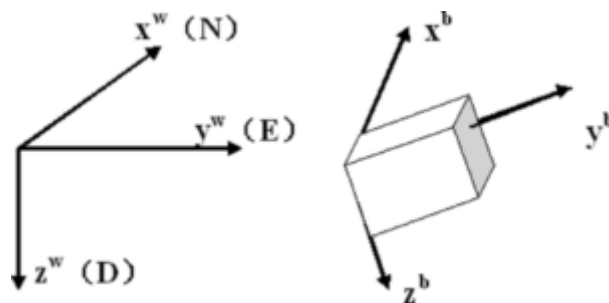


Abbildung 4.8 - Lokales Koordinatensystem

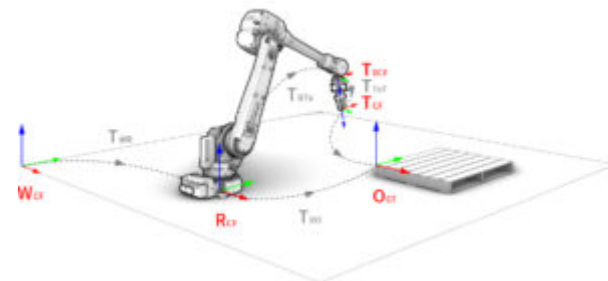


Abbildung 4.9 - Koordinatensysteme eines Roboters

4.4 – Vektoren

Ein Vektor zeigt eine Menge an, etwa Geschwindigkeit oder Kraft, welche eine Richtung und eine Länge aufweist. Vektoren in 3D-Koordinatensystemen werden mit einem geordneten Satz drei reeller Zahlen dargestellt.

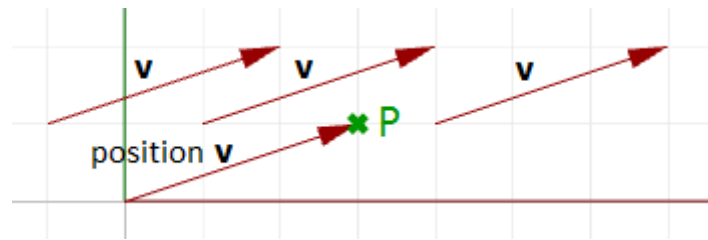
4.4.1 – Vektoren im Vergleich zu Punkten

Vektoren und Punkte dürfen nicht verwechselt werden. Es handelt sich dabei um sehr verschiedene Konzepte. Wie bereits erwähnt, stellen Vektoren eine Menge mit einer Richtung und Länge dar, während Punkte einen Standort anzeigen. Norden als Richtung ist zum Beispiel ein Vektor, während der Nordpol ein Standort (Punkt) ist.

Angenommen, wir haben einen Vektor und einen Punkt mit den gleichen Komponenten, etwa:

$$v = \langle 3, 1, 0 \rangle$$

Dann können wir Vektor und Punkt folgendermaßen zeichnen:



Ein Einheitsvektor ist ein Vektor, dessen Länge einer Einheit entspricht.

Eine Möglichkeit zum Bestimmen einer Ebene besteht, wenn Sie einen Punkt haben und einen Vektor, der sich rechtwinklig zur Ebene befindet. Dieser Vektor wird üblicherweise als der Normalenvektor zu dieser Ebene bezeichnet.

4.5 - Transformationen

Transformationen bezieht sich auf Operationen wie das Verschieben (auch Versetzen genannt), Drehen und Skalieren von Objekten. Befehle zur Transformation eines Objekts bezüglich Größe, Form, Position oder Ausrichtung. Die meisten Transformationen behalten die Parallelbeziehung der Geometrieteile zueinander bei. Kollineare Punkte, welche auf einer Geraden liegen, bleiben zum Beispiel auch nach der Transformation kollinear. Auch Punkte auf einer Ebene bleiben nach der Transformation kollinear. Diese Art von Transformation wird auch affine Transformation genannt.

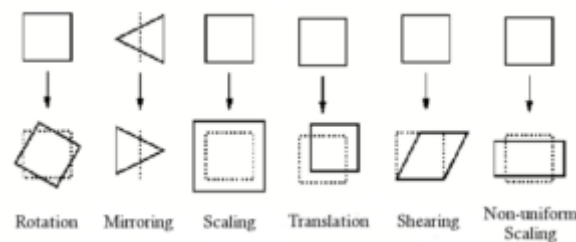


Abbildung 4.11 – verschiedene affine Transformationen

4.6 – Constructive Solid Geometry

Constructive Solid Geometry (CSG) oder **konstruktive Festkörpergeometrie** ist eine Technik zum **Modellieren** von Körpern, die u. a. in der 3D-**Computergrafik** und bei **CAD**-Programmen genutzt wird. Constructive Solid Geometry ermöglicht einem Designer einen komplex geformten Körper zu erzeugen, indem er mittels booleschen Operatoren verschiedene Grundkörper zu einem neuen Körper kombiniert. Aus der CSG hervorgegangene Körper wirken oft sehr komplex, sind aber in Wirklichkeit nichts anderes als geschickt verknüpfte Objekte.

In der **Mathematik** ist eine **boolesche Algebra** eine spezielle **algebraische Struktur**, die die Eigenschaften der **logischen** Operatoren UND, ODER, NICHT sowie die Eigenschaften der **mentheoretischen** Verknüpfungen Durchschnitt, Vereinigung, Komplement verallgemeinert. Die boolesche Algebra ist die Grundlage bei der Entwicklung von digitaler Elektronik und wird dort als **Schaltalgebra**, etwa bei der Erstellung von **Schaltnetzen** durch logische Schaltungen, angewandt. Sie wird in allen modernen Programmiersprachen zur Verfügung gestellt.^[1] Die Operatoren der logischen Verknüpfung werden **Boolesche Operatoren** genannt. Mit Hilfe der logischen Verknüpfungen lassen sich aus einfacheren **Aussagen** kompliziertere Aussagen zusammensetzen. Praktische Anwendung finden die logischen Verknüpfungen unter anderem als Suchoperatoren bei Datenbanken, Anfragen an Suchmaschinen, aber auch in der Constructive Solid Geometry.

Die Basisobjekte, aus denen CSG-Körper hervorgehen, nennt man Primitive (vgl. [Grafisches Primitiv](#)). Typischerweise handelt es sich dabei um Körper, deren Oberfläche mittels einer relativ einfachen mathematischen Formel beschrieben werden kann, wie z. B. [Würfel](#), [Zylinder](#), [Prismen](#), [Pyramiden](#), [Kugeln](#) oder [Ringe](#). Die Menge der möglichen Primitive wird gewöhnlich von der verwendeten Software begrenzt. Einige Software-Pakete erlauben CSG auf gekrümmten

Objekten (prozedurale oder parametrische Oberflächen), während andere nur auf polygonalen Meshes (Dreiecksnetze) arbeiten. Der prozedurale oder parametrische Ansatz erlaubt eine mathematisch exakte Berechnung und Repräsentation der Körper, während Meshes immer nur eine mehr oder weniger ungenaue Annäherung an die Wirklichkeit sind.

Ein komplexer Körper wird von Primitiven erzeugt, die durch Operationen verknüpft sind. Gewöhnlich handelt es sich dabei um **boolesche Operationen** auf Mengen: **Vereinigung** (Union, $\cup$), **Differenz** (Difference, $-$) und **Schnitt** (Intersection, $\cap$). Folgende Abbildung zeigt die Wirkung der Operatoren exemplarisch an der Verknüpfung von Würfel mit Kugel:

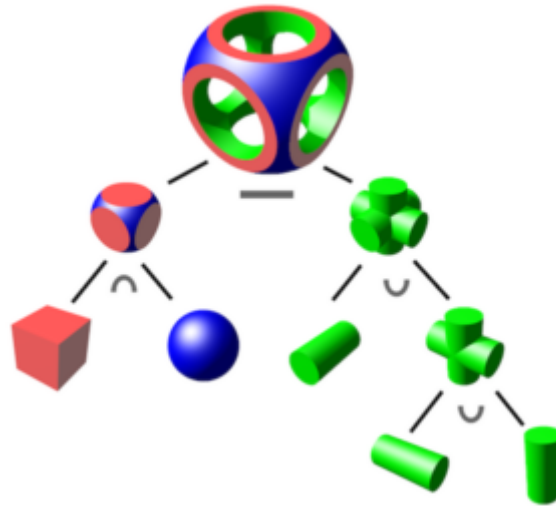
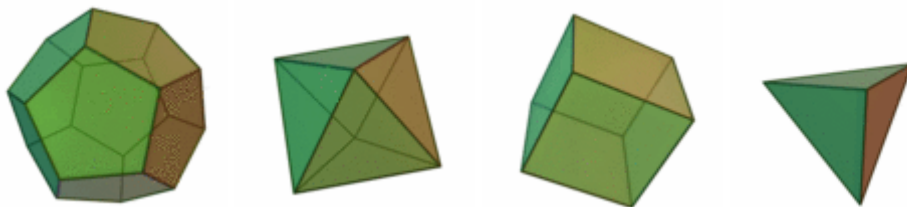


Abbildung 4.15 – CSG-Objekte können durch Binärbäume dargestellt werden, wobei Blätter Primitive darstellen und Knoten Operationen darstellen. In dieser Abbildung sind Knoten für Schnittmenge, für Vereinigung und für Unterschied aufgeführt.

4.7 – Platonische Körper

Die **Platonischen Körper** (nach dem griechischen Philosophen **Platon**) sind die **Polyeder** mit größtmöglicher **Symmetrie**. Jeder von ihnen wird von mehreren deckungsgleichen (**kongruenten**) ebenen **regelmäßigen Vielecken** begrenzt. Eine andere Bezeichnung ist *reguläre Körper* (von *lat. corpora regularia*).^{[1][2]} Es gibt fünf platonische Körper. Ihre Namen enthalten die griechisch ausgedrückte Zahl ihrer begrenzenden Flächen und *eder* als Abwandlung des griechischen Wortes ἑδρα (*hedra*) (s. auch **Polyeder**), deutsch (Sitz-)Fläche. Vollkommen regelmäßige Körper. Ihre Oberflächen bestehen aus gleich großen, gleichseitigen und gleichwinkligen Vielecken.



„An architectural drawing, is as much a prospective unfolding of future possibilities as it is a recovery of a particular history, to whose intentions it testifies and whose limits it always challenges. In any case a drawing is more than the shadow of an object, more than a pile of lines, more than a resignation to the inertia of convention.“ –Daniel Libeskind

4.8 – Bildtafeln

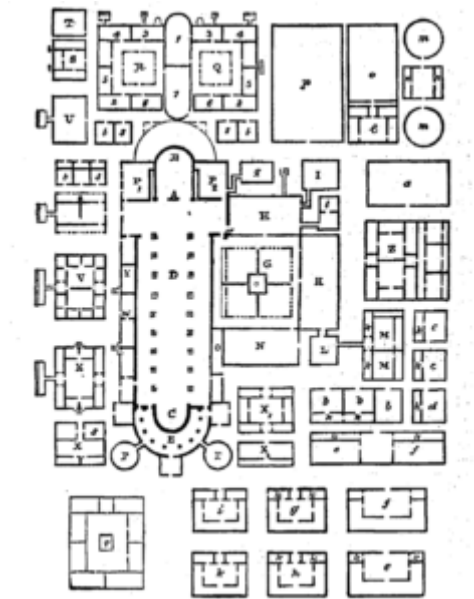


Abbildung 4.20 – St. Galler Klosterplan (Reichenau, frühes 9. Jahrhundert)



Abbildung 4.21 – Selbstportrait von M.C. Escher (1898 – 1972) in sphärischem Spiegel. Das Bild stammt aus dem Jahr 1935 und trägt den Titel "Hand mit spiegelnder Kugel".

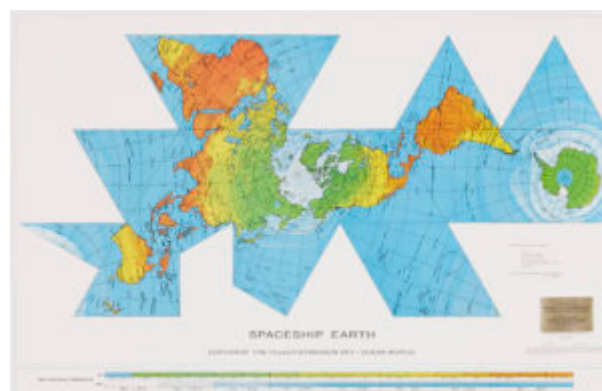


Abbildung 4.22 – Buckminster Fuller, Dymaxion Karte



Abbildung 4.23 - Louis Kahn's National Assembly Building of Bangladesh

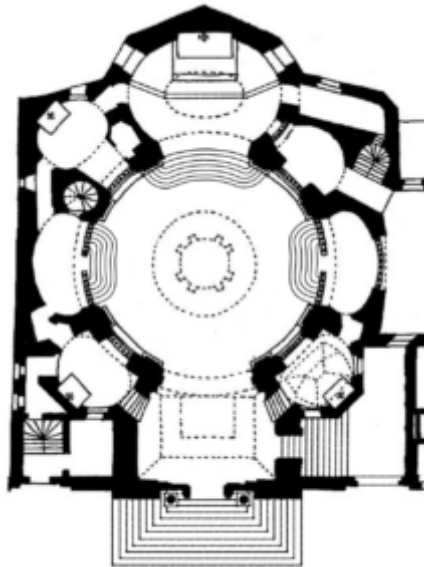


Abbildung 4.24 - Kirche Saint-Marie-de-la Visitation, Francois Mansart



Abbildung 4.25 - Zeitz Mocaa Museum 2016, Thomas Heatherwick



Abbildung 4.26 – VitraHaus, Herzog & de Meuron, 2010



Abbildung 4.27 – Peter Eisenman – Transformations, Decompositions, Critiques

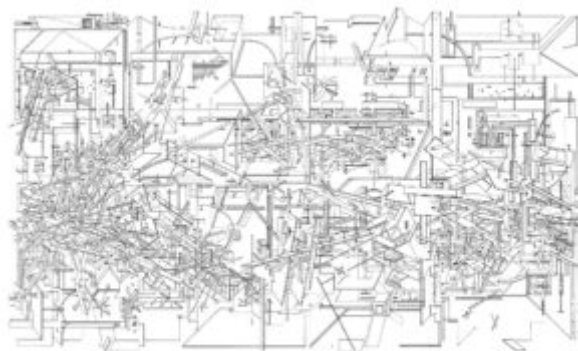


Abbildung 4.28 – Linienzeichnung, Daniel Libeskind

Parametrische Kurven und Flächen

5.1 – NURBS

Non-uniform rational B-Splines (deutsch: *nicht-uniforme rationale B-Splines*, kurz **NURBS**) sind **mathematisch** definierte **Kurven** oder **Flächen**, die im **Computergrafik**-Bereich, beispielsweise im **CGI** oder **CAD**, zur Modellierung beliebiger Formen verwendet werden. **NURBS** sind mathematische Darstellungen, die beliebige Formen von einfachen 2D-Linien, -Kreisen, -Bogen oder -Kurven bis hin zu hoch komplexen organischen 3D-Freiformflächen und -Volumenkörpern darstellen können. Aufgrund ihrer Flexibilität und Genauigkeit können NURBS-Modelle in allen Prozessen von Illustration und Animation bis hin zur Fertigung verwendet werden.

In den **1950er Jahren** wurden besonders im **Automobil**- und **Schiffbau** für die fehlerfreie Reproduzierbarkeit technischer Bauteile mathematisch exakte Beschreibungen von **Freiformflächen** benötigt. Vor dieser Zeit wurden derartige Oberflächen durch einzelne von einem **Konstrukteur** hergestellte **physikalische Modelle** beschrieben. So begann unter anderem der **Ingenieur Pierre Étienne Bézier**, zu dieser Zeit bei **Renault** in **Frankreich**, mit der Entwicklung der nach ihm benannten **Bézierkurve**. Unabhängig von Bézier arbeitete **Paul de Casteljau**, angestellt bei **Citroën**, zur gleichen Zeit auch an diesem mathematischen Problem. Weil Bézier die Ergebnisse seiner Arbeit veröffentlichte, werden heutzutage in der graphischen Datenverarbeitung **Splines**, deren Kontrollpunkte nicht auf der **Kurve** selbst liegen, als „Bézier-Spline“ bezeichnet, während der Name von de Casteljau in dem nach ihm benannten **Algorithmus** fortlebt, der für die **numerische** Verarbeitung **parametrischer Flächen** eingesetzt wird. In den 1960er Jahren wurde klar, dass **non-uniform rational B-Splines (NURBS)** eine Generalisierung von Bézier-Splines sind.

NURBS-Kurven und -Flächen verhalten sich ähnlich und haben eine gemeinsame Terminologie. Da Kurven am einfachsten zu beschreiben sind, werden wir sie genauer betrachten. Eine NURBS-Kurve wird durch vier Dinge definiert: Grad, Kontrollpunkte, Knoten und Bewertungsregel.

5.2 – Kurven

NURBS-Kurven und -Flächen sind die vorrangigen mathematischen Darstellungen, die NURBS-basierte CAD Software wie Rhino zur Geometriedarstellung verwendet. NURBS ist eine genaue mathematische Darstellung von Kurven, deren Bearbeitung sehr intuitiv geschieht. Freiformkurven können leicht unter Verwendung von NURBS dargestellt werden, und dank der Kontrollstruktur ist eine leichte und vorhersagbare Bearbeitung möglich.

5.2.1 – Kurvenparameter

Ein Parameter auf einer Kurve repräsentiert die Adresse eines Punkts auf dieser Kurve. Man kann sich die parametrische Kurve als einen innerhalb einer bestimmten Zeit zurückgelegten Weg zwischen zwei Punkten vorstellen, dies mit einer festgelegten oder variablen Geschwindigkeit. Wenn für die Wegstrecke der Zeitraum T benötigt wird, dann repräsentiert der Parameter t eine Zeit innerhalb von T , welche als ein Standort (Punkt) auf der Kurve erscheint.

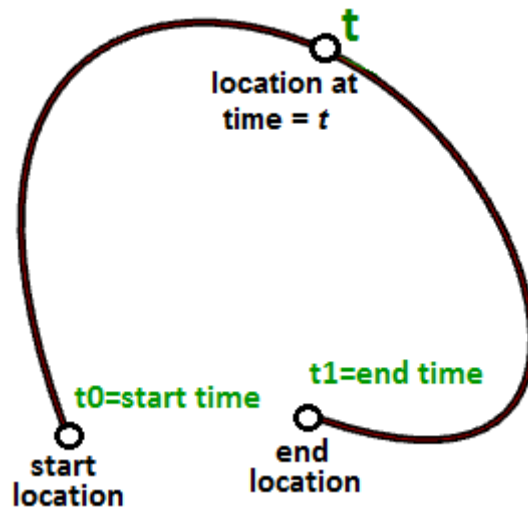


Abbildung 5.1 – Zeit ist der Parameter zur Bestimmung der Wegstrecke einer parametrischen Kurve. Der von Beginn bis Ende benötigte Zeitraum wird Kurvendomäne oder Intervall genannt.

5.2.2 – Kurvendomäne oder Intervall

Eine Kurvendomäne oder Intervall wird definiert als der Bereich von Parametern, die zu einem Punkt innerhalb dieser Kurve ausgewertet werden. Die Domäne wird gewöhnlich mit zwei reellen Zahlen beschrieben, welche die in der Form (min bis max) oder (min, max) ausgedrückten Domänenbeschränkungen definieren. Die Domänenbeschränkungen können zwei beliebige Werte sein, die auf die tatsächliche Länge der Kurve bezogen sein können oder auch nicht. In einer ansteigenden Domäne wird der Parameter der min-Domäne zum Startpunkt der Kurve ausgewertet, und max-Domäne zum Endpunkt der Kurve.

5.2.3 – Kurvenauswertung

Wir haben gelernt, dass es sich bei einem Kurvenintervall um den Bereich aller Parameterwerte handelt, die zu Punkten innerhalb der 3D-Kurve ausgewertet werden. Es besteht jedoch keinerlei Garantie, dass beispielsweise die Auswertung auf der Mitte der Domäne einen Punkt in der Mitte der Kurve ergibt.

Die gleichförmige Parametrisierung einer Kurve können wir uns vorstellen, als würden wir eine Wegstrecke mit konstanter Geschwindigkeit zurücklegen. Eine Linie von Grad 1 zwischen zwei Punkten wäre ein Beispiel, bei dem gleiche Intervalle von Parametern gleiche Intervalle einer Bogenlänge auf der Linie ergeben. In parametrischen Kurven werden gleiche Intervalle von Parametern nur selten zu gleichen Intervallen auf der 3D-Kurve ausgewertet.

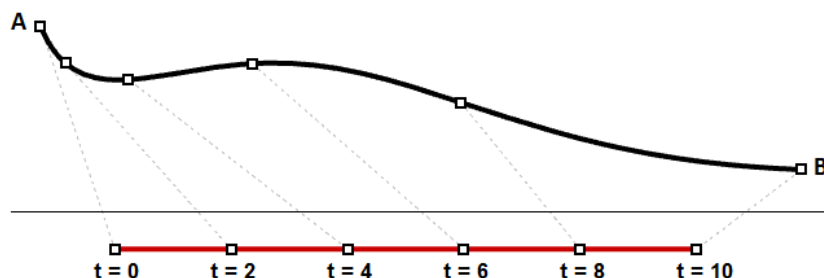


Abbildung 5.2 – Parameterspace (Domäne) von 0-10 einer Kurve. Gleiche Parameterintervalle ergeben typischerweise keine gleich langen Abstände auf einer parametrischen Kurven, etwa NURBS-Kurven.

5.2.4 – Kurvengrad

Der Kurvengrad ist eine ganze positive Zahl. Rhino ermöglicht das Arbeiten mit jeglicher Gradkurve, beginnend bei 1. Die Grade 1, 2, 3 und 5 sind die nützlichsten, Grad 4 und alle über Grad 5 hingegen finden in der wirklichen Welt wenig Anwendung. Manchmal werden die Begriffe linear (Grad 1), quadratisch (Grad 2), und kubisch verwendet (Grad 3). Es folgen einige Beispiele für Kurven und ihren Grad:

- Linien und Polylinien sind NURBS-Kurven ersten Grades.
- Kreise und Ellipsen sind Beispiele für NURBS-Kurven zweiten Grades.
- Freiform-Kurven werden gewöhnlich als NURBS-Kurven dritten oder fünften Grades dargestellt.

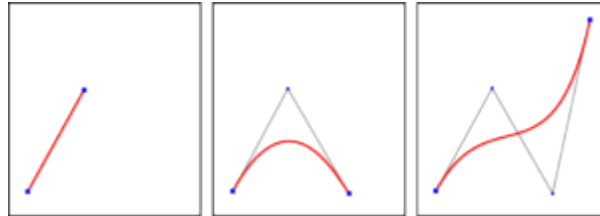


Abbildung 5.3 – Bézierkurven der Grade 1, 2 und 3 (rot) und die zugehörigen Kontrollpolygone (grau). Von links nach rechts wurde jeweils ein weiterer Kontrollpunkt (blau) hinzugefügt. Man erkennt, wie die Kurve bei Einfügen/Verändern eines Kontrollpunkts ihre Richtung und/oder Krümmung variiert.

Es ist möglich, den Grad einer NURBS-Kurve zu erhöhen, aber nicht ihre Form zu verändern. Es ist jedoch im Allgemeinen nicht möglich, den Grad einer NURBS-Kurve zu reduzieren, ohne ihre Form zu ändern.

5.2.5 – Kontrollpunkte

Die Kontrollpunkte einer NURBS-Kurve ist eine Liste von (Grad+1)-Punkten. Die intuitivste Methode, die Form einer NURBS-Kurve zu verändern, besteht in der Verschiebung ihrer Kontrollpunkte.

Die Anzahl der Kontrollpunkte, die sich auf das jeweilige Segment einer NURBS-Kurve auswirken, wird durch den Kurvengrad bestimmt. Beispielsweise unterliegt das jeweilige Segment in einer Kurve ersten Grades lediglich der Auswirkung der beiden End-Kontrollpunkte. In einer Kurve zweiten Grades wirken sich drei Kontrollpunkte auf das jeweilige Segment aus, und so weiter.

- Kurven ersten Grades gehen durch alle Kurvenkontrollpunkte. In einer NURBS-Kurve ersten Grades bestimmen zwei (Grad+1)-Kontrollpunkte jedes Segment. Bei Verwendung von fünf Kontrollpunkten hat die Kurve vier Segmente.
- Kreise und Ellipsen sind Beispiele für Kurven zweiten Grades. In einer NURBS-Kurve zweiten Grades bestimmen drei (Grad+1)-Kontrollpunkte das jeweilige Segment. Bei Verwendung von fünf Kontrollpunkten hat die Kurve drei Segmente.

Die Kontrollpunkte haben eine angegliederte Zahl, die Wichtung genannt wird. Bis auf einige Ausnahmen sind Wichtungen positive Zahlen. Intuitiv sind die Wichtungen denkbar als die Schwere jedes einzelnen Kontrollpunkts. Je höher die relative Wichtung eines Kontrollpunkts, desto näher wird die Kurve an diesen Kontrollpunkt herangezogen. Wenn die Kontrollpunkte einer Kurve alle dieselbe Wichtung (normalerweise 1) haben, wird die Kurve nicht-rational genannt. Andernfalls wird die Kurve rational genannt. Das R in NURBS steht für rational und weist darauf hin, dass für eine NURBS-Kurve die Möglichkeit besteht, rational zu sein. In der Praxis sind die meisten NURBS-Kurven nicht-rational. Manche NURBS-Kurven – Kreise und Ellipsen sind herausragende Beispiele – sind immer rational.

5.2.6 – Bewertungsregel

Die NURBS-Bewertungsregel ist eine Formel, die mit Grad, Kontrollpunkten und Knoten¹ arbeitet. Die Formel enthält sogenannte B-Spline-Basisfunktionen die einen Parameter nimmt und einen Punktstandort erzeugt. Grad, Knoten und Kontrollpunkte sind die Parameter dieser Formel.²

5.2.7 – Geometrische Kurvenstetigkeit

Stetigkeit ist ein wichtiges Konzept in der 3D-Modellierung. Stetigkeit ist wichtig, um visuelle Glätte wie auch sanftes Licht und Luftströmung zu erhalten. Die folgende Tabelle enthält verschiedene Stetigkeiten und ihre Definitionen:

G0 (Positionsstetig)	Zwei miteinander verbundene Kurvensegmente
G1 (Tangentenstetig)	Die Richtung der Tangente am Verbindungspunkt ist für beide Kurvensegmente gleich
G2 (Krümmungsstetig)	Krümmungen wie auch Tangenten stimmen für beide Kurvensegmente am gemeinsamen Endpunkt überein

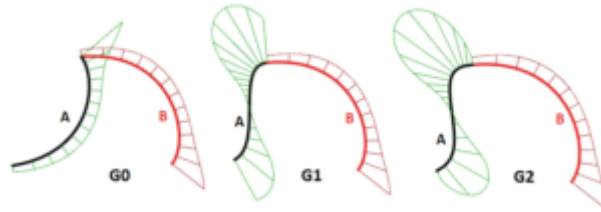


Abbildung 5.4 – Überprüfung der Kurvenstetigkeit Analyse der Krümmungsanzeige.

Krümmung ist ein bei der Modellierung von 3D-Kurven und -Flächen viel verwendetes Konzept. Krümmung wird definiert als die Änderung in der Neigung einer Tangente zu einer Kurve über eine Bogeneinheitlänge. Für einen Kreis oder eine Kugel ist dies der Kehrwert des Radius und über die gesamte Domäne hinweg konstant.

An jedem beliebigen Punkt auf einer Kurve in der Ebene ist die tangentielle Linie die Linie, die sich der Kurve durch diesen Punkt am meisten nähert. Wir können auch den besten annähernden Kreis finden, der durch diesen Punkt verläuft und tangential zur Kurve liegt. Der Kehrwert des Radius dieses Kreises ist die Krümmung der Kurve an diesem Punkt.

5.3 – Flächen

5.3.1 – Abwickelbare Fläche

Eine abwickelbare Fläche bezeichnet in der Geometrie bzw. in der Differentialgeometrie, der Kartografie und der Topologie eine Fläche, die sich ohne innere Formverzerrung aus dem Euklidischen Raum in die Euklidische Ebene transformieren/„abwickeln“ lässt. Die sich ergebende Fläche wird dann Abwicklung genannt. Anschaulich gesprochen: Ohne Stauchen und Zerren muss sich die abwickelbare Fläche glatt auf eine flache Ebene legen lassen. Bekannteste Beispiele sind die Mantelflächen bestimmter dreidimensionaler Körper wie Zylinder oder Kegel.

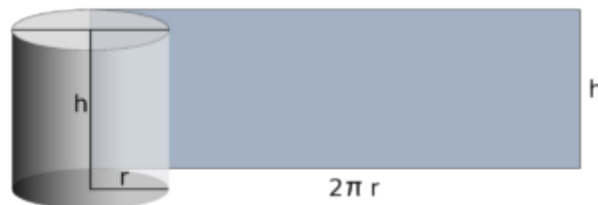


Abbildung 5.5 – Abwicklung der Mantelfläche eines Zylinders

Knoten sind eine Liste von Zahlen mit (Grad+N-1), wobei N für die Anzahl Kontrollpunkte steht. Manchmal wird diese Zahlenliste auch Knotenvektor genannt. In diesem Zusammenhang bedeutet das Wort Vektor nicht 3D-Richtung. Für den Designer sind Knoten gewöhnlich nicht hilfreich; sie werden nur für interne Berechnungen benötigt. Aus diesem Grund sind Knoten in vielen Graphik-Design-Programmen nicht änderbar oder überhaupt sichtbar. Einige CAD Programme erlauben interaktive Änderungen an Knotenpositionen, was allerdings bedeutend weniger intuitiv ist als Änderungen an Kontrollpunkten.

2.

Quelle

5.3.2 – Regelfläche

Durch jeden Punkt der Fläche geht eine Gerade, die ganz in der Fläche enthalten ist. Hyperbolische Paraboloiden sind Regelflächen.

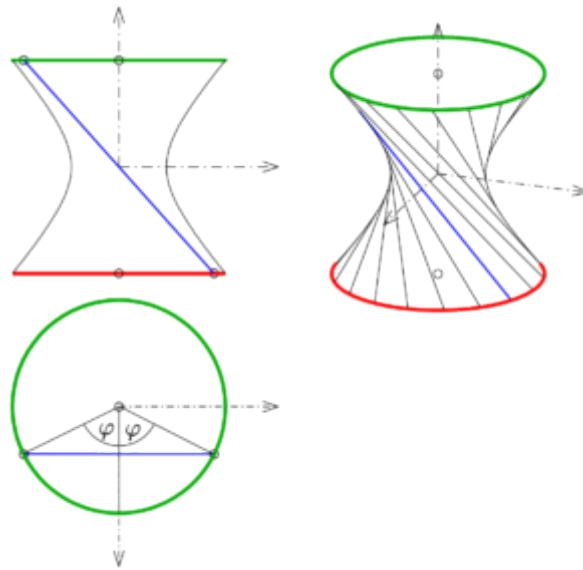


Abbildung 5.6 – Regelfläche: einschaliges Hyperboloid.

5.4 – Parametrische Flächen

5.4.1 – Flächenparameter

Eine parametrische Fläche ist eine Funktion zweier unabhängiger Parameter (meist u , v genannt) über einer zweidimensionalen Domäne.

5.4.2 – Flächendomäne

Eine Flächendomäne wird definiert als der Bereich von (u,v) -Parametern, die zu 3D-Punkten auf dieser Fläche ausgewertet werden. Die Domäne in jeder Dimension (u oder v) wird gewöhnlich als zwei reelle Zahlen beschrieben ($u_{\min}$ bis $u_{\max}$) und ($v_{\min}$ bis $v_{\max}$).

5.4.3 – Flächenauswertung

Die Auswertung einer Fläche auf einem Parameter innerhalb der Flächendomäne ergibt einen Punkt auf der Fläche. Bedenken Sie, dass die Mitte der Domäne (mid_u , mid_v) nicht unbedingt zum Mittelpunkt der 3D-Fläche ausgewertet wird. Auch wird die Auswertung der u - und v -Werte, die sich außerhalb der Flächendomäne befinden, kein gutes Ergebnis erbringen.

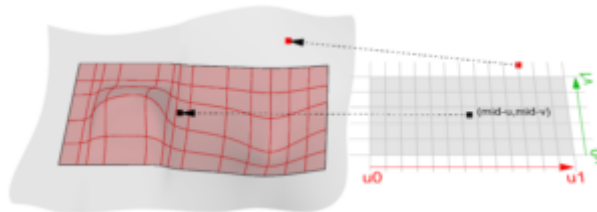


Abbildung 5.7 – NURBS-Fläche in 3D-Modellierungsraum (links). Das Flächenparameterrechteck mit Domäne, die von u_0 bis u_1 in der ersten Richtung und von v_0 bis v_1 in der zweiten Richtung reicht (rechts). Flächenauswertung.

5.4.4 – Geometrische Flächenstetigkeit

Viele Modelle können nicht anhand einer einzigen Fläche erzeugt werden. Stetigkeit zwischen verbundenen Flächen ist von Bedeutung für visuelle Glätte, Lichtreflektierung und Luftströmung.

Die folgende Tabelle enthält verschiedene Stetigkeiten und ihre Definitionen:

G0 (Positionsstetig)	Zwei miteinander verbundene Flächen.
G1 (Tangentenstetig)	Die einander entsprechenden Tangenten der beiden Flächen entlang ihrer Verbindungskante sind sowohl in u- als auch v-Richtung parallel.
G2 (Krümmungsstetig)	Krümmungen wie auch Tangenten stimmen für beide Flächen an der gemeinsamen Kante überein.

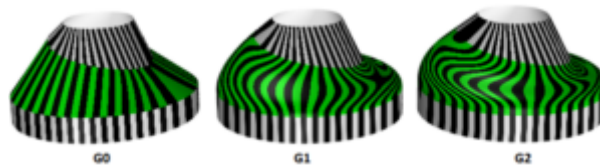


Abbildung 5.8 – Prüfung der Flächenstetigkeit mit der Lichtlinienanalyse.

5.5 – NURBS Flächen

Sie können sich NURBS-Flächen als ein Raster von NURBS-Kurven vorstellen, die in zwei Richtungen verlaufen. Die Form einer NURBS-Fläche wird durch eine Zahl von Kontrollpunkten bestimmt, sowie den Grad dieser Fläche in jede dieser beiden Richtungen (u- und v-Richtungen). NURBS-Flächen sind effizient bei der Speicherung und Darstellung von Freiformflächen mit hohem Genauigkeitsgrad. Die mathematischen Gleichungen und Details von NURBS-Flächen gehen über den Rahmen dieses Texts hinaus. Stattdessen konzentrieren wir uns lieber auf die für Designer relevantesten Eigenschaften.

5.5.1 – Getrimmte NURBS-Flächen

NURBS-Flächen können getrimmt oder ungetrimmt sein. Getrimmte Flächen verwenden eine darunterliegende NURBS-Fläche und geschlossene Kurven, um einen Teil dieser Fläche herauszutrimmen. Jede Fläche hat eine geschlossene Kurve, welche die äußere Kante (äußere Schleife) bestimmt, und sie kann nicht- überschneidende, geschlossene innere Kurven zur Bestimmung von Öffnungen (innere Schleifen) haben. Eine Fläche mit einer äußeren Schleife, welche die gleiche wie die ihrer darunterliegenden NURBS-Fläche ist und keine Öffnungen hat, ist was wir eine ungetrimmte Fläche nennen.

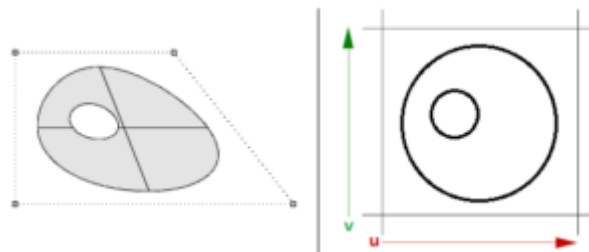


Abbildung 5.9 – Getrimmte Fläche in Modellierungsraum (links) und in Parameterrechteck (rechts).

5.5.2 – Flächenverbände

Ein Flächenverband besteht aus zwei oder mehr (möglicherweise getrimmten) miteinander verbundenen NURBS-Flächen. Jede Fläche hat ihre eigene Struktur, Parametrisierung und Iso-kurven-Richtungen, die nicht übereinstimmen müssen. Flächenverbände werden unter Verwendung des Begrenzungsflächenmodells (**BRep** engl. Abk. für Boundary Representation)

dargestellt. Die BRep-Struktur beschreibt Flächen, Kanten und Scheitelpunkte mit Trimmdaten und Verbindung unterschiedlicher Teile. Getrimmte Flächen werden auch unter Verwendung von BRep-Datenstruktur dargestellt.

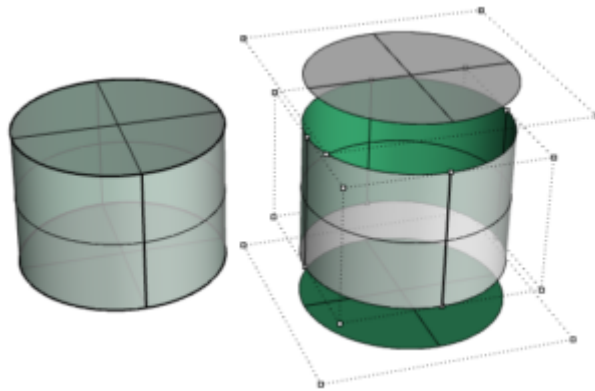


Abbildung 5.10 – Die oberen und unteren Seiten des Zylinders im folgenden Beispiel sind aus planaren Flächen getrimmt. Die Abbildung zeigt die Kontrollpunkte der darunterliegenden Flächen.

5.6 – Bildtafeln

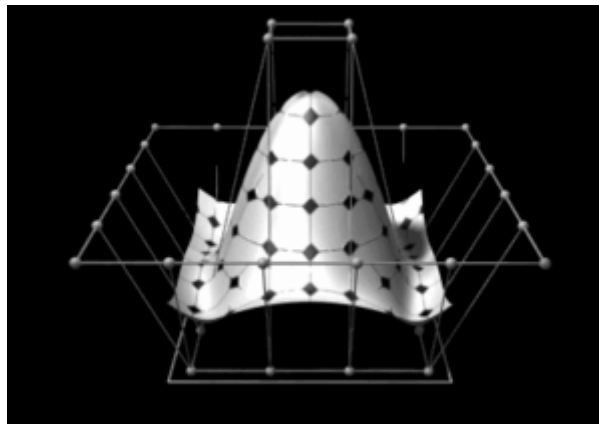


Abbildung 5.11 – Dreidimensionale NURBS-Flächen können komplexe, organische Formen aufweisen. Kontrollpunkte beeinflussen die Richtungen der Oberfläche.

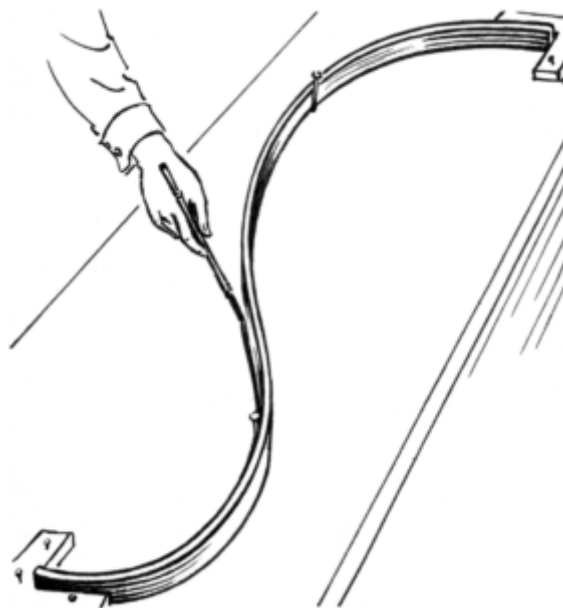


Abbildung 5.12 – Zeichnen einer Spline

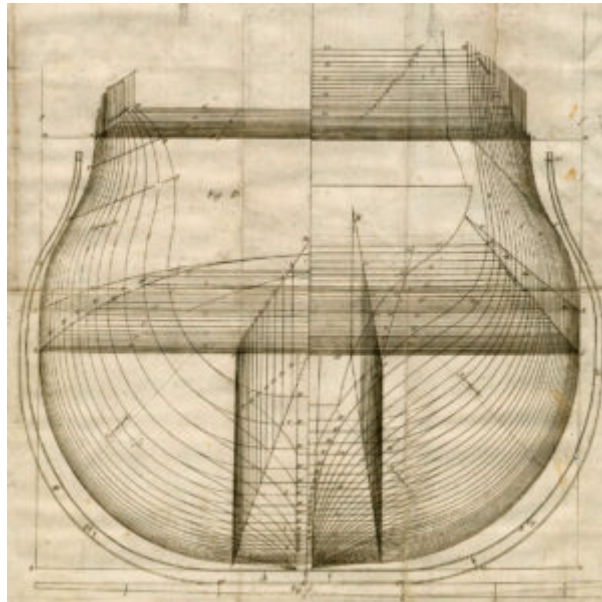


Abbildung 5.13 – A ship's body plan comprised of superimposed construction geometry. Cross sections such as these recorded vessels on contractual order. The outer member (marked Fig. E/F) is the midships mould. Portions of each cross section are indexed to corresponding pieces on the reconfigurable mould. Source: William Sutherland, *The Shipbuilders Assistant: or, Some Essays Towards Completing the Art of Marine Architecture* (London, 1711), 82.

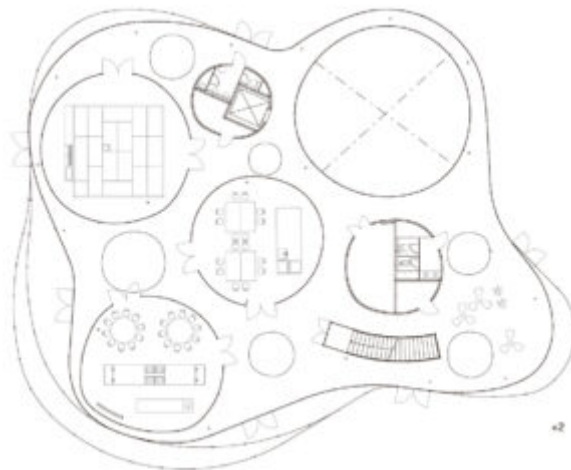


Abbildung 5.14 – Toyota Aizuma Hall, Aichi – Kazuyo Sejima & Associates



Abbildung 5.15 – Kurven einer Achterbahn



Abbildung 5.16 – Autobahnkreuz, Los Angeles



Abbildung 5.17 – BMW Gina, Design Chris Bangle



Abbildung 5.18 – Enric Ruiz-Geli created the Villa Nurbs in the coastal town Empuriabrava at the Costa Brava, Spain.

Geometrische Strukturen

6.1 – Punktwolken

Eine **Punktwolke** oder ein **Punkthaufen** (englisch point cloud) ist eine Menge von Punkten eines **Vektorraums**, die eine unorganisierte räumliche Struktur („Wolke“) aufweist.^[1] Eine Punktwolke ist durch die enthaltenen Punkte beschrieben, die jeweils durch ihre Raumkoordinaten erfasst sind. Punktwolken mit **Georeferenzierung** enthalten Punkte in einem erdbezogenen Koordinatensystem. Zu den Punkten können zusätzlich Attribute, wie z. B. geometrische Normalen, Farbwerte, Aufnahmezeitpunkt oder Messgenauigkeit, erfasst sein.

Die Erzeugung kann grundsätzlich über Scanning-Verfahren (z. B. terrestrisches oder flugzeuggestütztes **Laserscanning**) oder **photogrammetrische Verfahren**^[2] erfolgen sowie allgemein mittels Abtastung von Objektoberflächen durch Systeme wie Koordinatenmessmaschinen oder tastende **3D-Scanner**. Optische Scanner untergliedert man in **Lasertechnologie**, die nach dem Triangulationsprinzip arbeiten, und Normallicht-Scanner, die nach dem Streifenlichtverfahren („coded-light“) arbeiten. Einen zusammenfassenden Überblick über die Vielfalt und Leistungsfähigkeit aktueller optischer Scanning-Methoden und die Weiterverarbeitung der resultierenden 3D-Daten/Punktwolke gibt beispielsweise C. Teutsch.^[3] Durch die mehrfache Erfassung eines räumlichen Ausschnitts zu unterschiedlichen Zeitpunkten lässt sich ein vierdimensionales (zeitvariantes) diskretes räumliches Modell einer Umgebung aufbauen.^[4] Jeder Punkt der Wolke wird dabei zeitlich und räumlich (**XYZ-Koordinaten**) lokalisiert und kann in weiterer Folge auch georeferenziert werden.^[5]

Es gibt viele Möglichkeiten um aus einer Punktwolke eine geschlossene 3D-Oberfläche zu erstellen.

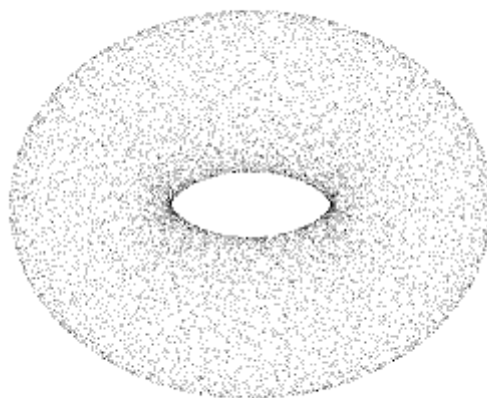


Abbildung 6.1 – Animiertes Punktwolkenmodell eines Torus

6.2 – Mesh – Diskrete Flächennetze

Untereinander mit Kanten verbundene **Punkte** bilden in der **Computergrafik** ein **Polygonnetz oder Mesh**, das die Gestalt eines **Polyeders** definiert. **Dreiecksnetze** und **Vierecksnetze** sind hier am geläufigsten.

In der 3D-Computergrafik und der Volumenmodellierung ist ein Polygonnetz eine Sammlung von Scheitelpunkten, Kanten und Flächen, die die Form eines polyedrischen Objekts definieren. Die Flächen bestehen normalerweise aus Dreiecken (Triangle Mesh), Vierecken (Quads) oder anderen einfachen konvexen Polygonen (n-Gons), da dies das Rendern vereinfacht, können aber auch allgemeiner aus konkaven Polygonen oder sogar Polygonen mit Löchern zusammengesetzt sein.

Die Untersuchung von Polygonnetzen ist ein großes Teilgebiet der Computergrafik (insbesondere der 3D-Computergrafik) und der geometrischen Modellierung. Für unterschiedliche Anwendungen und Ziele werden unterschiedliche Darstellungen von Polygonnetzen verwendet. Die Vielfalt der an Netzen durchgeführten Operationen kann umfassen: Boolesche Logik (konstruktive Festkörpergeometrie), Glättung, Vereinfachung und viele andere. Es gibt auch Algorithmen für Raytracing, Kollisionserkennung und Starrkörperdynamik mit Polygonnetzen. Wenn die Kanten des Netzes anstelle der Flächen gerendert werden, wird das Modell zu einem Drahtgittermodell.

Volumetrische Netze unterscheiden sich von Polygonnetzen dadurch, dass sie explizit sowohl die Oberfläche als auch das Volumen einer Struktur darstellen, während Polygonnetze nur explizit die Oberfläche darstellen (das Volumen ist implizit).



Abbildung 6.2 – Unterschiedliche Drahtgittermodelle, die einfachste Möglichkeit, ein Polygonnetz darzustellen.

6.3 – Voxel – Diskrete Volumen

Bei einem räumlichen Datensatz, der in **diskretisierter** Form in **kartesischen Koordinaten** vorliegt, bezeichnet Voxel den **diskreten Wert** an einer XYZ-Koordinate des Datensatzes.

Voxel (zusammengesetzt aus dem englischen *volume* *vox* und *el* von *elements*^[1]) bezeichnet in der **Computergrafik** einen Gitterpunkt („Bild“punkt, Datenelement) in einem dreidimensionalen **Gitter**. Dies entspricht einem **Pixel** in einem 2D-Bild, einer **Rastergrafik**. Wie bei Pixeln wird bei Voxeln üblicherweise die Position nicht explizit gespeichert, sondern implizit aus der Position zu anderen Voxeln hergeleitet. Im Gegensatz dazu werden bei **Punkten** oder **Polygonen** die Positionen der Eckkoordinaten gespeichert.

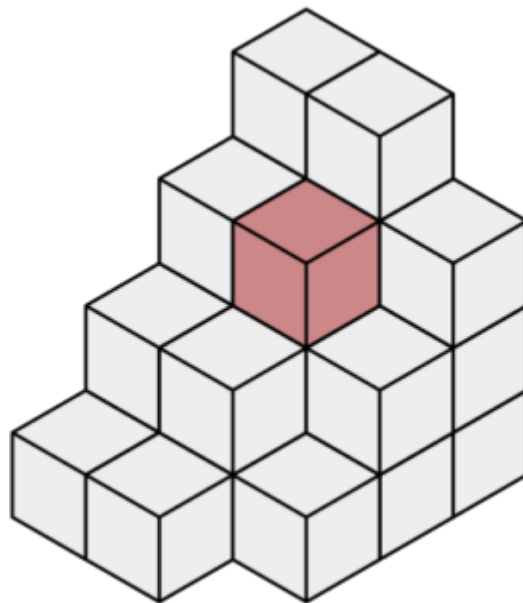


Abbildung 6.3 - Eine Menge von gestapelten Voxeln. Ein einziges ist hervorgehoben.



6.4 - Implizite Flächen

Implizite Flächen sind Flächen, die im Raum benachbarte Punkte gleicher Merkmale oder Werte einer bestimmten Größe wie zum Beispiel Temperatur oder Dichte miteinander verbinden. Sie sind das dreidimensionale Gegenstück zu [Isolinien](#), die Punkte auf einer Fläche verbinden. Die Bedeutung von Isoflächen liegt in der computergraphischen Visualisierung von [Skalarfeldern](#) bzw. Gittern.

Isolinien werden für die wissenschaftliche Visualisierung verwendet. In geografischen Karten wird die Topographie üblicherweise mit Hilfe von Höhenstufen (d. h. Kurven konstanter Höhe über dem Meeresspiegel) visualisiert (Abbildung 12.41). In Wetterkarten wird die Visualisierung oft durch Isolinien wichtiger Messdaten wie Temperatur oder Luftdruck ergänzt.

Pottmann, Helmut; Asperl, Andreas ; Höfer, Michael; Kilian, Axel. Architekturgeometrie. Bentley Institute Press. Kindle-Edition.

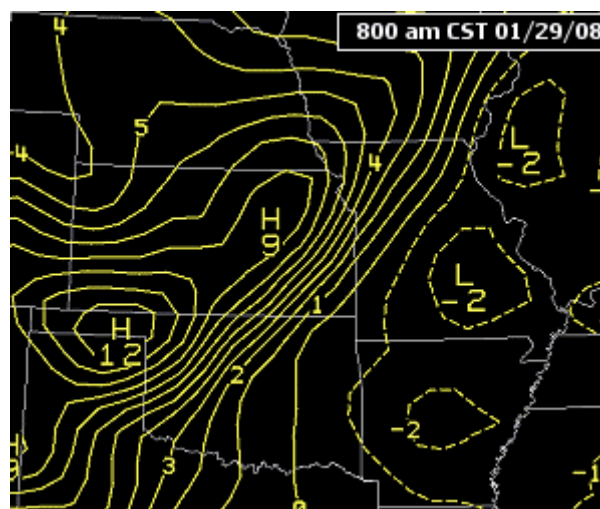


Abbildung 6.5 - Video loop of isallobars showing the motion of a cold front

Es gibt viele Wege, 3d Isoflächen zu **rendern**, die beiden gebräuchlichsten sind **Raycasting** und der **Marching-Cubes**-Algorithmus der die implizite Fläche zuerst in ein Mesh / Polygonnetz übersetzt.

6.4.1 – Signed Distance Functions

Signed Distance Function oder kurz SDFs geben, wenn sie die Koordinaten eines Punktes im Raum übergeben, den kürzesten Abstand zwischen diesem Punkt und einer Oberfläche zurück. Das Vorzeichen des Rückgabewerts gibt an, ob sich der Punkt innerhalb oder außerhalb dieser Oberfläche befindet (daher vorzeichenbehaftete Abstandsfunktion).

Beispiel

Schauen wir uns ein Beispiel an. Stellen Sie sich eine Kugel vor, die im Ursprung zentriert ist. Punkte innerhalb der Kugel haben einen Abstand vom Ursprung, der kleiner als der Radius ist, Punkte auf der Kugel haben einen Abstand, der gleich dem Radius ist, und Punkte außerhalb der Kugel haben einen Abstand, der größer als der Radius ist. Also sieht unsere erste SDF für eine Kugel mit Radius 1 am Ursprung zentriert so aus:

$f(1,0,0) = 0$
 $f(0,0,0.5) = -0.5$
 $f(0,3,0) = 2$

Großartig, (1, 0, 0) befindet sich auf der Oberfläche, (0, 0, 0.5) befindet sich innerhalb der Oberfläche, wobei der nächste Punkt auf der Oberfläche 0.5 Einheiten entfernt ist, und (0, 3, 0) liegt außerhalb der Oberfläche, wobei der nächste Punkt auf der Oberfläche 2 Einheiten entfernt ist.

6.4.2 – Metaballs

Metakugeln. Metabälle, Metaballs, auch bekannt als „Blobs“ oder „weiche Objekte“, folgen dem gleichen Prinzip. Ihre Form wird durch Eingabeformen gesteuert, bei denen es sich in der Regel um einfache Objekte wie Punkte, Linien und einfache Oberflächen handelt. Unter Verwendung des Abstands zu den Eingabeformen wird jede Eingabeform einer lokalen Feldfunktion zugeordnet. Diese lokalen Feldfunktionen werden summiert, was eine Funktion $f(x,y,z)$ ergibt. Seine Niveaumengen $f(x,y,z) = k$ sind die interessierenden Flächen. Ein Beispiel wird gezeigt.¹

Ein **Metaball** ist das Ergebnis eines **Algorithmus**, der eine dehnbare Oberfläche erzeugt, die die Form einer Kugel (bei genau einem) oder eine Menge von ineinander gehenden Kugeln erzeugt. Der Algorithmus wurde in den frühen 1980ern von **Jim Blinn** entwickelt. Ein Metaball ist als eine **Funktion** in n Dimensionen definiert, für die üblichen drei Dimensionen also entsprechend $f(x, y, z)$. Um ein **Volumen** zu erzeugen wird ein **Schwellenwert** gewählt.

$$\sum_{i=0}^n \text{Metaball}_i(x, y, z) \leq \text{Schwellenwert}$$

definiert dann, ob der von n Metaballs definierte Körper am Punkt (x, y, z) gefüllt ist. Eine typische Metaball-Funktion ist

$$f(x, y, z) = \frac{1}{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}$$

1.

Pottmann, Helmut; Asperl, Andreas; Höfer, Michael; Kilian, Axel. Architekturgeometrie. Bentley Institute Press. Kindle-Edition.

wobei (x_0, y_0, z_0) das Zentrum des Balles angibt und (x, y, z) den zu untersuchenden Punkt. $f(x, y, z)$ gibt dann also die Stärke des Balles an diesem Punkt zurück, ist die Summe der Stärken aller Bälle an diesem Punkt größer als der Schwellenwert, so ist der Körper dort gefüllt. Da die Funktion auf Grund der [Division](#) rechenintensiv ist, werden auch [polynomiale](#) Annäherungen verwendet.

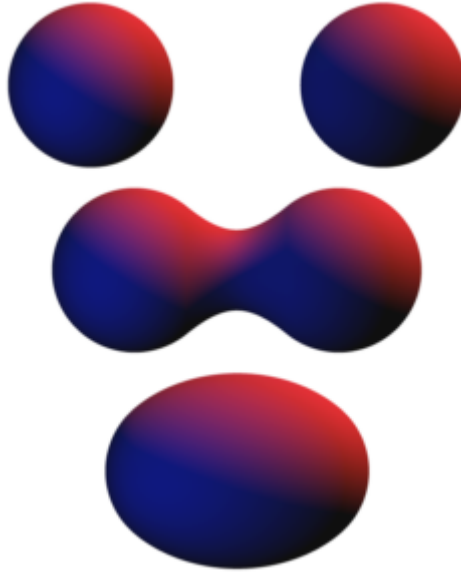


Abbildung 6.15 - Zwei positive Metaballs verschmelzen ineinander.

6.5 – Bildtafeln



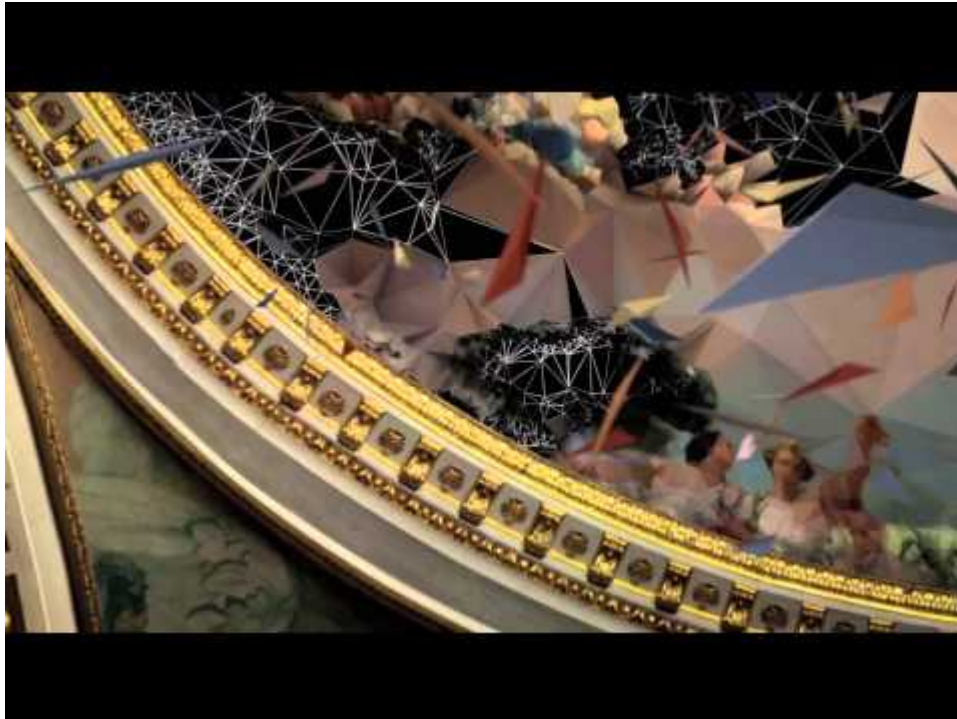
Abbildung 6.16 – Jakob + Macfarlane Architects, 2013



Abbildung 6.17 – Höhenlinien (Topographie)



Abbildung 6.18 – Galaxy Soho von Zaha Hadid in Peking



Ein YouTube Element wurde aus dieser Textversion entfernt. Sie können es hier online ansehen: <https://wp-prd.let.ethz.ch/computationaldesigniii/?p=444>



Ein Vimeo Element wurde aus dieser Textversion entfernt. Sie können es hier online ansehen: <https://wp-prd.let.ethz.ch/computationaldesigniii/?p=444>

Topologie, Graphen und Netzwerke

7.1 – Topologie

Die **Topologie** ([griechisch](#) τόπος *tópos*, deutsch ‚Ort, Platz, Stelle‘ und [-logie](#)) ist die Lehre von der Lage und Anordnung geometrischer Gebilde im Raum und ein fundamentales [Teilgebiet der Mathematik](#). Sie beschäftigt sich mit den Eigenschaften mathematischer Strukturen, die unter stetigen Verformungen erhalten bleiben, wobei der Begriff der [Stetigkeit](#) durch die Topologie in sehr allgemeiner Form definiert wird. Die Topologie ging aus den Konzepten der [Geometrie](#) und [Mengenlehre](#) hervor. Die **geometrische Topologie** ist ein [Teilgebiet der Mathematik](#), das sich mit [Mannigfaltigkeiten](#) (engl. *Manifold*) und deren [Einbettungen](#) beschäftigt.



Abbildung 7.1 – Tasse und Volltorus sind zueinander homöomorph. Anmerkung: Ein Homöomorphismus ist eine direkte Abbildung zwischen den Punkten der Tasse und des Volltorus, die Zwischenstufen im zeitlichen Verlauf dienen nur der Illustration der Stetigkeit dieser Abbildung.

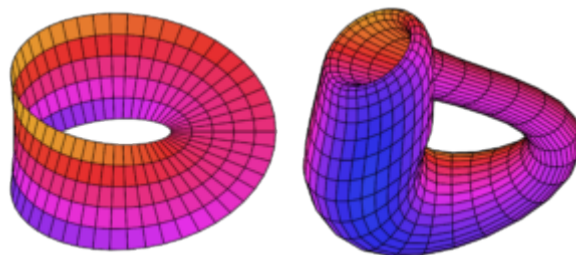


Abbildung 7.2 – Mathematisch gesehen ist das Möbiusband (links) eine nicht-orientierbare Mannigfaltigkeit. Eine weitere Fläche, die in diese Kategorie gehört, ist die Kleinsche Flasche (rechts)

Unter dem **Geschlecht** einer [kompakten orientierbaren Fläche](#) versteht man in der [Topologie](#) die Anzahl der „Löcher“ (oder der „Henkel“) der Fläche.



Abbildung 7.3 – Flächen mit Geschlecht 1, 2 und 3.

Die Mannigfaltigkeit einer Topologie erlaubt Rückschlüsse darauf, ob sich das Objekt 3D drucken lässt, denn diese gibt an, ob ein klar definiertes Innen und Aussen existiert.

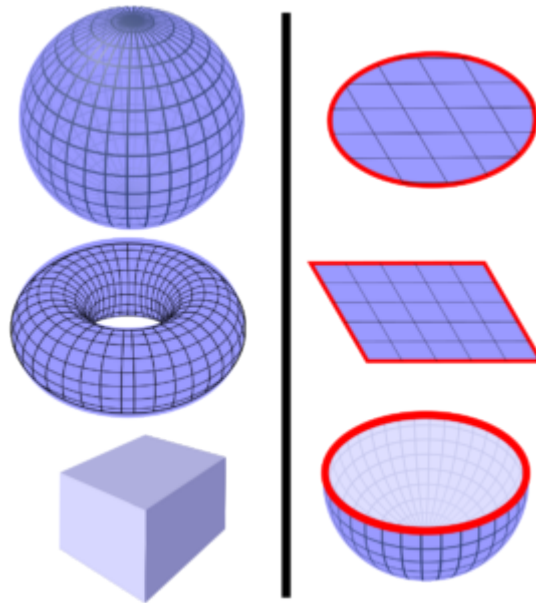


Abbildung 7.4 – Auf der linken Seite sind Mannigfaltigkeiten ohne Rand und auf der rechten Seite sind Mannigfaltigkeiten mit Rand abgebildet. Diese sind nicht direkt 3D druckbar, weil kein Innen und Aussen definiert ist.

7.2 – Graphen

Ein **Graph** ist in der [Graphentheorie](#) eine abstrakte Struktur, die eine Menge von Objekten zusammen mit den zwischen diesen Objekten bestehenden Verbindungen repräsentiert. Die mathematischen Abstraktionen der Objekte werden dabei [Knoten](#) des Graphen genannt. Die paarweisen Verbindungen zwischen Knoten heißen [Kanten](#). Die Kanten können [gerichtet](#) oder [ungerichtet](#) sein. Häufig werden Graphen anschaulich gezeichnet, indem die Knoten durch Punkte und die Kanten durch Linien dargestellt werden.^[2]

Beispiel

Anschauliche Beispiele für Graphen sind ein [Stammbaum](#) oder das [U-Bahn-Netz](#) einer Stadt (siehe Abbildungen). Bei einem Stammbaum stellt jeder Knoten ein Familienmitglied dar und jede Kante ist eine Verbindung zwischen einem Elternteil und einem Kind. In einem U-Bahn-Netz stellt jeder Knoten eine U-Bahn-Station dar und jede Kante eine direkte Zugverbindung zwischen zwei Stationen.

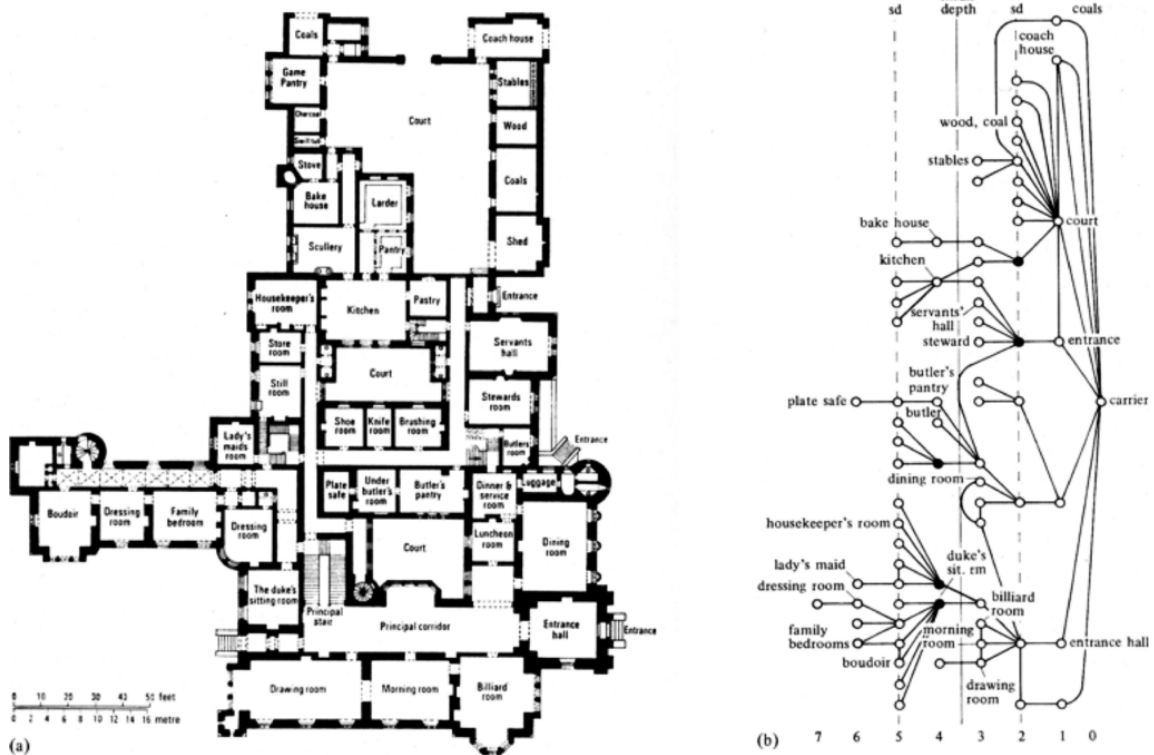


Abbildung 7.5 – Steadman, J. P. 1983. *Architectural Morphology an Introduction to the Geometry of Building Plans*

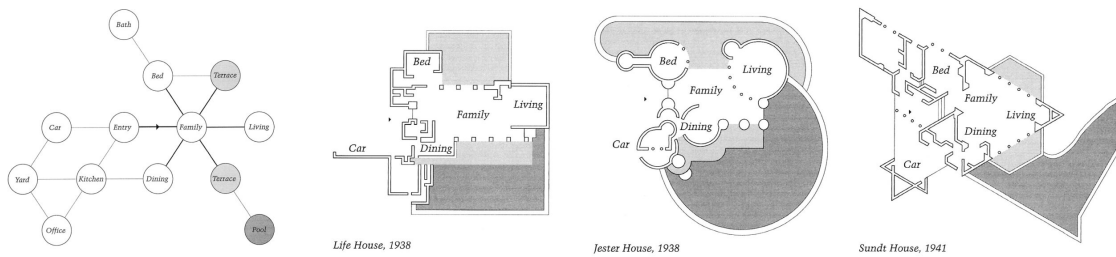


Abbildung 7.6 – Frank Lloyd Wright in Mitchell, William John. 1990. *The Logic of Architecture*. Mit Press



Abbildung 7.7 – Peter Eisenmann, *THE MAX REINHARDT HAUS*, Location: Berlin, Germany, Year: 1992

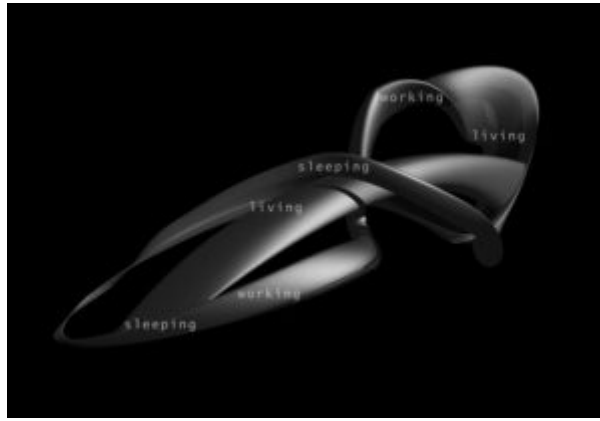


Abbildung 7.8 – Un Studio, Möbius House

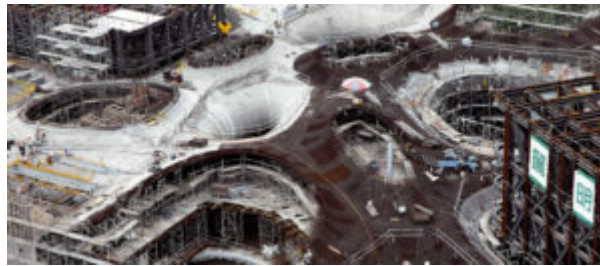


Abbildung 7.9 – National Taichung Theater von Toyo Ito & Associates



Abbildung 7.10 – NonLin/Lin von Marc Fornes und The Very Many



Abbildung 7.11 – OMA — Jussieu Library

Digitale Ausgabe

Digitale Bilder

8.1 – Bilddaten

8.1.1 – Raster / Vektor

Eine **Vektorgrafik** ist eine **Computergrafik**, die aus **grafischen Primitive** wie Linien, **Kreisen**, **Polygonen** oder allgemeinen Kurven (**Splines**) zusammengesetzt ist. Meist sind mit Vektorgrafiken Darstellungen gemeint, deren Primitive sich **zweidimensional** in der **Ebene** beschreiben lassen, ähnlich wie bei einem **Vektor** in der **analytischen Geometrie**. Eine Bildbeschreibung, die sich auf **dreidimensionale** Primitive stützt, wird eher **3D-Modell** oder **Szene** genannt und die Erzeugung zweidimensionaler Linien- und Flächengrafiken sowie fotorealistischer Bilder aus solchen 3D-Modellen wird **Bildsynthese** oder **Rendern** genannt.

Beispiel

Um beispielsweise das Bild eines Kreises zu speichern, benötigt eine Vektorgrafik mindestens zwei Werte: die Lage des Kreismittelpunktes und den Kreisdurchmesser. Neben der Form und Position der grafischen Primitive werden eventuell auch die **Farbe**, **Strichstärke**, diverse Füllmuster und weitere, das Aussehen bestimmende, Daten angegeben.^[1]

Im Vergleich zu Rastergrafiken lassen sich Vektorgrafiken oft mit deutlich geringerem Platzbedarf speichern. Eines der wesentlichen Merkmale und Vorteile gegenüber der Rastergrafik ist die stufenlose und verlustfreie **Skalierbarkeit**. Die Stärke von Vektorgrafiken allgemein ist die Auflösungsunabhängigkeit, d. h., sie sind für eine Wiedergabe (**Bildschirm**, Drucken) in beliebiger Auflösung geeignet. Dies erfordert jedoch immer ein aufwändiges **Rendern** der Vektorgrafik in eine **Rastergrafik**.

Eine **Rastergrafik**, auch **Pixelgrafik** (englisch *raster graphics image, digital image, bitmap* oder *pixmap*), ist eine Form der Beschreibung eines Bildes in Form von **computerlesbaren Daten**. Rastergrafiken bestehen aus einer rasterförmigen Anordnung sogenannter **Pixel**, denen jeweils eine **Farbe** zugeordnet ist. Die Hauptmerkmale einer Rastergrafik sind daher die Bildgröße, umgangssprachlich auch **Bildauflösung** genannt, sowie die **Farbtiefe**. Die Erzeugung und Bearbeitung von Rastergrafiken fällt in den Bereich der **Computergrafik** und **Bildbearbeitung**.

Ausgewählte Rasterformate	Ausgewählte Vectorformate
• Joint Photographic Experts Group (JPEG) • Portable Network Graphics (PNG) • Graphics Interchange Format (GIF) • Bitmap Image File (BMP) • Tagged Image File Format (TIFF) • Adobe Photoshop File (PSD)	• Scalable Vector Graphics (SVG) • Encapsulated PostScript File (EPS) • Adobe Illustrator File (AI) • Collaborative Design Activity (COLLADA) • PostScript (PS) • Enhanced MetaFile (EMF)

8.1.2 – Auflösung

Die **Bildauflösung**, kurz auch **Auflösung** genannt, ist ein umgangssprachliches Maß für die Bildgröße einer **Rastergrafik**. Sie wird durch die Gesamtzahl der **Bildpunkte** oder durch die Anzahl der Spalten (Breite) und Zeilen (Höhe) einer Rastergrafik angegeben.

Der Begriff **Auflösung** wird in der Praxis mehrdeutig und in vielen Bereichen verwendet, wodurch es zu Missverständnissen kommen kann. Auflösung im physikalischen Sinn (*Bildelemente pro Länge*) bezeichnet die **Punktdichte** einer *Wiedergabe* oder Bildabtastung und ist damit – neben der **Farbtiefe** – ein Maß für die Qualität.

Bei Rastergrafiken selbst, die z. B. als Datei vorliegen, kann diese Qualität nicht angegeben werden, da zunächst unklar ist, wie die Wiedergabe erfolgt. So kann eine kleinere, beispielsweise nur 200 Byte große **Favicon**-Grafikdatei eine durchaus „exzellente und 100 % perfekte“ Wiedergabequalität liefern. *Auflösung* im technischen Sinn ist wiedergabebezogen. Solange die Wiedergabe auf physikalisch immer gleichen Medien erfolgt, beispielsweise einem 9 × 13 cm großen **Fotoabzug** oder identischen **Fernsehern**, hängt die dort erreichte Qualität auch von der Größe der ursprünglichen Rastergrafik ab. Da im Allgemeinen jedoch nicht klar ist, wie die Ausgabe in allen späteren Fällen genau erfolgt, kann die „Bildauflösung“ nicht als direktes Maß für eine allgemeine Wiedergabequalität dienen.

Für technische Prozesse, die eine Rastergrafik wiedergeben, gilt: Je größer die Grafik ist („Bildauflösung“ bzw. Bildgröße in Pixeln),

- desto besser *kann* die erreichte Wiedergabequalität sein
- oder desto größer *kann* mit identischer Qualität wiedergegeben werden.

Die **Farbtiefe** gibt die *Feinheit* der Abstufungen an, mit der die Farbe einzelner Bildelemente einer Rastergrafik wiedergegeben werden kann. Neben der Pixelanzahl ist sie eine der bestimmenden Größen einer Rastergrafik.

8.1.3 – Farbmodelle

Alle **Farben** eines **Farbmodells**, die durch eine farbgebende Methode tatsächlich ausgegeben werden können, werden in dem dreidimensionalen **Farbraum** dargestellt. Jede farbgebende Methode hat ihren eigenen Farbraum. Eine Darstellung aller *Farbörter* (Singular: **Farbort**) eines Farbmodells bildet den **Farbkörper**.

Alle Verfahrensweisen und damit verbundenen Geräte und Materialien, die Farbe zur Darstellung bringen können, werden farbgebende Methoden genannt. Solche Verfahren sind **Drucker**, **Monitore**, **Ausbelichtungen**, **Kunstdrucke**, Lackierungen oder manueller Farbauftrag.

- **RGB-Farbraum** – Ein RGB-Farbraum ist ein additiver Farbraum, der Farbwahrnehmungen durch das additive Mischen dreier Grundfarben (Rot, Grün und Blau) nachbildet. Verwendung: Computer**monitore**, Internetstandard.
- **CMYK-Farbmodell** – Das CMYK-Farbmodell ist ein subtraktives Farbmodell, das die technische Grundlage für den modernen Vierfarbdruck (Cyan, Magenta, Yellow und den Schwarzanteil Key) bildet. Verwendung: **Desktoppublishing**, **Druck-Endstufe**.
- **HSV-Farbraum** In Fragen der Farbnachstellung wird das HSV-Modell (Farbton, Sättigung, Helligkeit) gegenüber den Alternativen RGB und CMYK bevorzugt, weil es der menschlichen Farbwahrnehmung stärker entspricht (beispielsweise ist es im RGB-Modell schwer, sich Gelb als Mischung von Rot und Grün vorzustellen). Verwendung: **Design**, Dokumentation von **Malerei**, **Videokunst**.

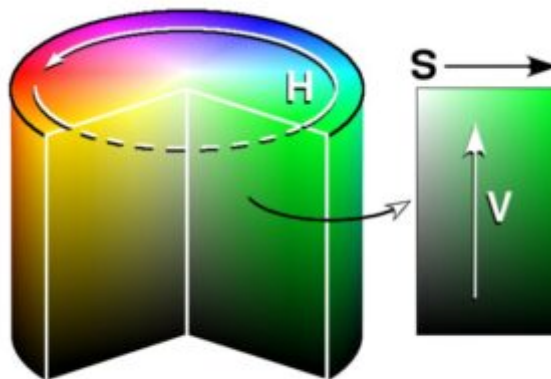


Abbildung 8.1 – HSV Farbzyylinder

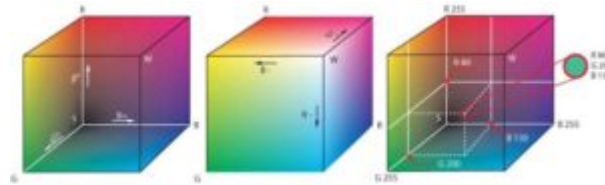


Abbildung 8.2 – RGB Farbwürfel

8.2 – Bildsynthese – Rendering

Raytracing (dt. **Strahlverfolgung**^[1] oder **Strahlenverfolgung**^[2] in englischer Schreibweise meist *ray tracing*) ist ein auf der Aussendung von **Strahlen** basierender **Algorithmus** zur **Verdeckungsrechnung**, also zur Ermittlung der Sichtbarkeit von dreidimensionalen Objekten von einem bestimmten Punkt im Raum aus. Ebenfalls mit Raytracing bezeichnet man mehrere Erweiterungen dieses grundlegenden Verfahrens, die den weiteren Weg von Strahlen nach dem Auftreffen auf Oberflächen berechnen.

Prominenteste Verwendung findet Raytracing in der 3D-**Computergrafik**. Hier ist der grundlegende Raytracing-Algorithmus eine Möglichkeit zur Darstellung einer **3D-Szene**. Erweiterungen, die den Weg von **Lichtstrahlen** durch die Szene simulieren, dienen, ebenso wie das **Radiosity**-Verfahren, der Berechnung der Lichtverteilung.

Globale Beleuchtung (**englisch Global Illumination**, abgekürzt **GI**) bezeichnet in der **3D-Computergrafik** allgemein die Simulation der Licht-Wechselwirkungen von Objekten. Im Speziellen sind damit nur diejenigen Verfahren gemeint, die alle Möglichkeiten der Ausbreitung von Lichtstrahlen in einer **3D-Szene** berücksichtigen. Dadurch werden die Gesetze der **geometrischen Optik** sowie der Energieerhaltung vollständig erfüllt und ein relativ realistisches Bild erzeugt.

Eine andere Möglichkeit, echte globale Beleuchtung zu simulieren, besteht in der Verwendung von **High Dynamic Range Images** (HDRIs), auch *Environment Maps* genannt, die die Szene umschließen und diese ausleuchten. Dieses Verfahren wird als **Image-based Lighting** bezeichnet.

8.3 – Bildkomposition

8.3.1 – Architekturfotographie

Everyone will have noticed how much easier it is to get hold of a painting, more particularly a sculpture, and especially architecture, in a photograph than in reality.

Walter Benjamin

<https://davidcampany.com/architecture-as-photography-document-publicity-commentary/>

Warum ist Architekturfotografie für Architekten relevant?

- Architekten erstellen Pläne und Bilder, keine Gebäude.
- Pictural turn: Das Bild ist das Hauptmedium unserer Zeit.
- Architekten erfahren durch Architekturfotografie, wie Menschen Architektur wahrnehmen
- Szenografisches Design: Architekten lernen beim Entwerfen die wichtigsten Perspektiven zu berücksichtigen (Man möchte Gebäude entwerfen, von denen man gerne Fotos machen würde).
- Man lernt über Licht, Schattenwirkung sowie Perspektiven, Sichtbarkeiten und Blickbeziehungen.

8.3.2 – Regel der Drittel

Bei der Drittel-Regel wird das Bild gedanklich in neun Teile geschnitten. Man zieht zwei waagerechte und zwei senkrechte Linien, so dass alle neun Teile gleich groß sind. Das zu fotografierende **Motiv** wird an einem der vier Schnittpunkte angelegt, man kann es aber auch längs einer Linie platzieren. Nach der Proportionsregel des **Goldenen Schnitts** sollte das Motiv das Gesamtbild im Goldenen Schnitt teilen beziehungsweise der Abstand des Motivs vom Bildrand zur Bildlänge im Verhältnis $1:\Phi \approx 0,618$ stehen. Da eine exakte Platzierung jedoch aufwendig ist, benutzt man stattdessen als grobe aber einfach zu verwendende Näherung $\frac{2}{3}:1 = 2:3$.

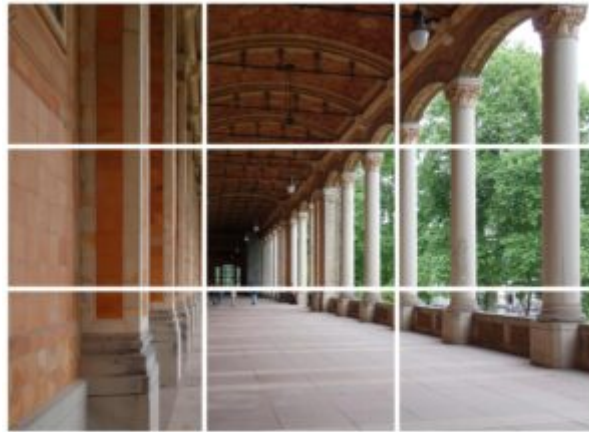


Abbildung 8.3 – Der Ausgang der Wandelhalle im Kurpark von Baden Baden liegt an einem der vier Schnittpunkte.

8.3.3 – Konvergierende Vertikalen

Konvergierende Vertikalen stören uns nicht, wenn wir eine Straße entlang gehen, aufgrund der engen Zusammenarbeit zwischen unserem Gleichgewichtssinn und unserer visuellen Wahrnehmung. Anders sieht es jedoch aus, wenn wir in zweidimensionalen Bildern aufeinander zulaufende Vertikalen betrachten, denn unser Gehirn signalisiert sofort, dass etwas nicht stimmt. Nähern sich konvergierende Vertikalen in einem leichten Winkel, kann ein Bild einen unruhigen Eindruck erwecken und die Ästhetik eines abgebildeten Gebäudes beeinträchtigen. Dies kann den Betrachter verwirren und einen erschütternden Eindruck eines Gebäudes erzeugen, das manchmal so streng ist, dass es den Anschein erweckt, als würde es umkippen nach hinten. Eine ungeschriebene Regel schreibt vor, dass Architekturfotografen im Allgemeinen versuchen, vertikale Linien in ihren Bildern vertikal zu halten.

Da ein Objekt oft nicht aus einem größeren Abstand fotografiert werden kann, sollte die Neigung der Senkrechten zueinander auf das als natürlich empfundene Maß reduziert werden. Oft möchte man die Senkrechten auch ganz parallel machen, da eine Neigung gegenüber dem Bildrand auch als störend empfunden wird. Die dazu notwendige Maßnahme wird als **Shiften** bezeichnet. Bevorzugt kann dies unter Verwendung geeigneter Kameras und Objektive schon bei der Aufnahme geschehen. Alternativ, falls solche Möglichkeiten nicht zur Verfügung stehen, ist dies auch mittels digitaler Nachbearbeitung möglich.

Bei **Großformatkameras** bzw. dem **Tilt-und-Shift-Objektiv** von Kleinbild- und Mittelformatkameras werden stürzende Linien vermieden, indem zur Wahl des Bildausschnitts nicht die Kamera verschwenkt, sondern der Bildausschnitt verschoben wird. Für die digitale Nachbearbeitung stehen entsprechende Funktionen in den meisten Bildbearbeitungsprogrammen zur Verfügung.



Abbildung 8.4 – Korrektur von stürzenden Linien.

8.3.4 – Variation of distance / focal length

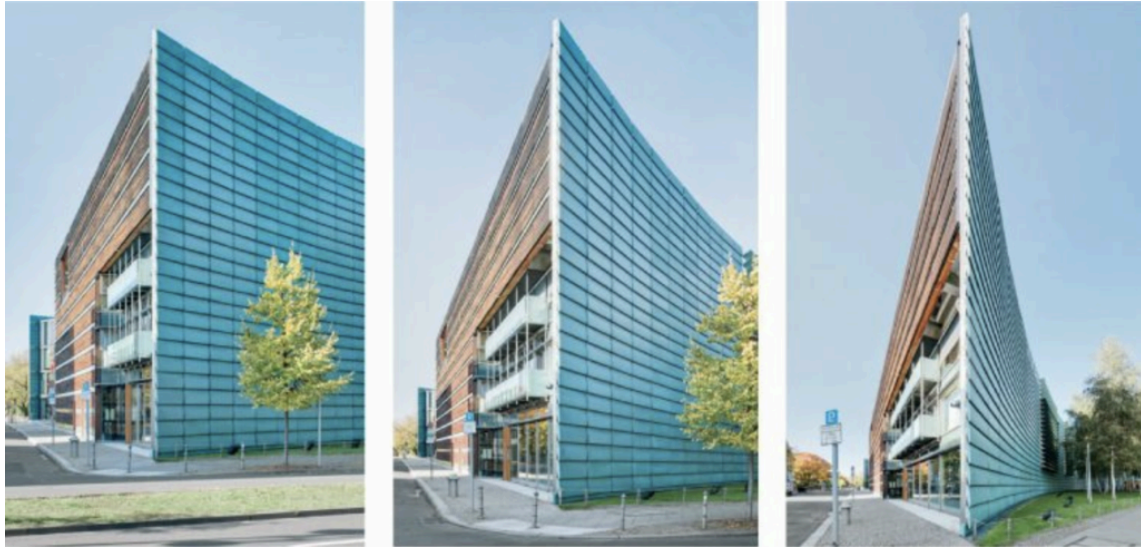


Abbildung 8.5 – Auf der linken Seite erzeugt ein mittlerer Motivabstand ein natürlich wirkendes Bild mit einem realistischen Eindruck von Tiefe. In der Mitte und rechts führt die zunehmende Nähe zum Motiv zu immer dramatischeren Raumeffekten [Brennweiten: 50mm, 24mm, 12mm].

8.3.5 – Belichtung

Dreipunktbeleuchtung ist eine Standardmethode, die in visuellen Medien wie Theater, Video, Film, Standfotografie, computergenerierten Bildern und 3D-Computergrafik verwendet wird.

- **Das Führungslicht.** Die Sehgewohnheiten des Menschen sind von jeher mit der Lichtsituation einer nahezu punktförmigen Lichtquelle (Sonne oder Mond) vertraut. Die klassische Ausleuchtung greift darauf zurück und setzt auch ein dominantes Licht, das (oft als einziges Licht in der Szene) auch Schatten wirft. Dies ist das Führungslicht. Oft wird die Szene in Richtung der Kamera oder von links bzw. rechts oberhalb der Kamera ausgeleuchtet. Ein Führungslicht von genau oberhalb der Szene verursacht tiefe Schatten in den Augenhöhlen, ein Licht von unten wirkt ebenfalls ungewöhnlich und kann sehr drastische, unheimliche Szenen liefern. Dies gilt auch für ein Gegenlicht als Führungslicht. Das Führungslicht kann nicht nur die im Mittelpunkt der Szene stehenden Objekte und Personen ausleuchten, es ist das bevorzugte Mittel, um den Blick des Betrachters auf das Wesentliche zu lenken. Die Lichtqualität kann hart oder weich sein.
- **Die Aufhellung.** Insbesondere harte Führungslichter oder Gegen- und Seitenlichter führen zu hohen Lichtkontrasten, die vom **Dichteumfang** des Filmes oder Sensors nicht mehr verarbeitet werden könnten. Manchen Motiven wird auch durch geringere Kontraste geschmeichelt (z. B. das **Beauty-Light** in der Porträtfotografie). Dies ist die Aufgabe der Aufhellung, die die vom Führungslicht weniger getroffenen Stellen stärker beleuchten soll. Dafür ist ein weiches, nicht Schatten werfendes Licht gut geeignet, um dem Führungslicht nicht seine Dominanz zu nehmen. Als Aufhellung können künstliche Lichtquellen oder Reflektoren zum Einsatz kommen.
- **Effektlicht (oder Kante, Haarlicht, siehe auch **Spitzlicht**).** Dies ist ein zumeist von hinten auf das Motiv gerichtetes Licht, das zum Beispiel helfen soll, einen Menschen vom Hintergrund abzuheben (Glanz im Haar). Die Lichtstärke ist oft hoch, die Lichtqualität hart.
- **Weitere Lichtquellen.** Zum Beispiel zur Beleuchtung des Hintergrundes; kann auch mit Durchlicht durch den Hintergrund erfolgen (siehe auch **Hohlkehle**) oder zur Akzentuierung von weiteren Teilen des Motivs.

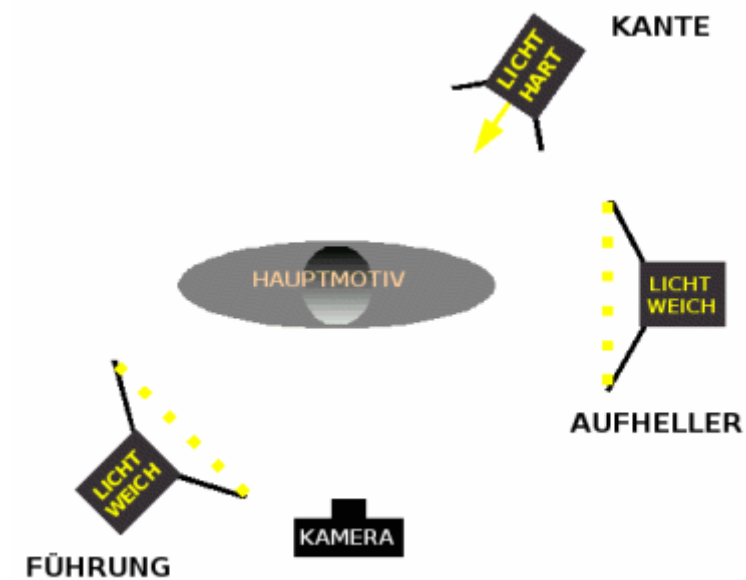


Abbildung 8.6 – Einfache Lichtführung aus einem Führungslicht, Aufheller und einem Effektlucht. Statt einer künstlichen Lichtquelle als Aufheller kann auch ein Reflektor verwendet werden

Ein Videoelement wurde aus dieser Textversion entfernt. Sie können es hier online ansehen: <https://wp-prd.let.ethz.ch/computationaldesigniii/?p=23>

8.4 – Bildtafeln

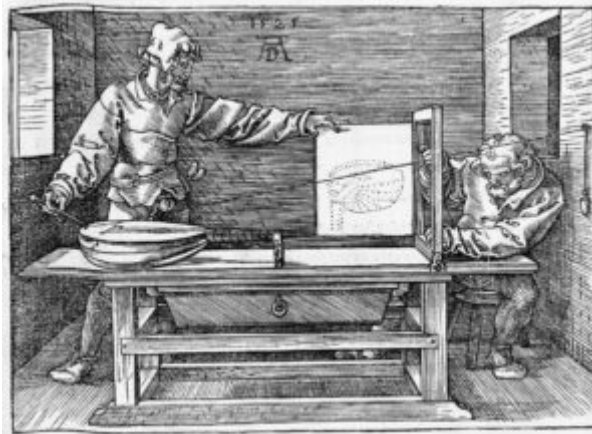


Abbildung 8.7 – Albrecht Dürer, *Unterweysung der messung mit dem zirckel un richt scheyt*, 1525

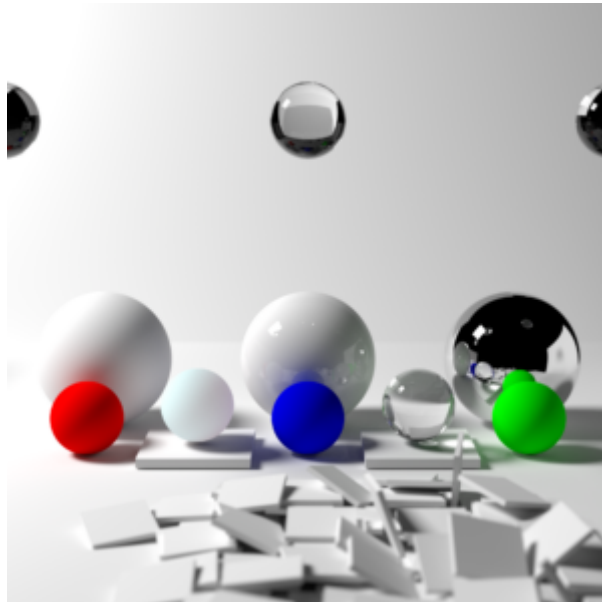


Abbildung 8.8 – An image rendered using path tracing, demonstrating notable features of the technique



Abbildung 8.9 – Abgewickelte Textur um einen Kopf zu UV-Mappen

Digitale Fabrikation

9.1 – Modelle

Ein **Modell** ist ein vereinfachtes Abbild der Wirklichkeit. Das Abbild kann die Form konkreter Gegenstände haben (Modell-Eisenbahn, Computersimulation u. ä.) oder rein abstrakt dargestellt sein (Theorien; Gleichungen). Nach [Herbert Stachowiak](#) kennzeichnen ein Modell mindestens drei Merkmale:^[1]

1. **Abbildung:** Ein Modell steht immer *für etwas* anderes – nämlich für ein natürliches oder ein künstliches Original, welches es somit abbildet oder repräsentiert. Von Modellen lassen sich ihrerseits Modelle anfertigen, in denen somit das Original z. B. um noch einen Schritt weiter vereinfacht repräsentiert ist. Beispiel für einen partiellen Komplexitätszuwachs ist die [Ausschnittsvergrößerung](#) einer [Technischen Zeichnung](#).
2. **Verkürzung:** Ein Modell erfasst nicht alle Attribute des Originals, sondern nur diejenigen, die dem Modellschaffer bzw. Modellnutzer relevant erscheinen.
3. **Pragmatismus:** Modelle sind ihren Originalen nicht eindeutig zugeordnet. Sie erfüllen ihre Ersetzungsfunktion
 - für bestimmte Subjekte (*für wen?*)
 - innerhalb bestimmter Zeitintervalle (*wann?*)
 - unter Einschränkung auf bestimmte gedankliche oder tatsächliche Operationen (*wozu?*).

In der [Architektur](#) versteht man unter einem Architekturmodell oder kurz Modell die [maßstäbliche](#) Darstellung eines [Entwurfes](#). Dabei kann sowohl ein [reales](#) als auch ein [virtuelles](#) Gebäudemodell gemeint sein. Wichtig ist bei diesen Modellen nicht unbedingt eine hohe Detailtreue, sondern vor allem die Darstellung der zentralen Ideen und des [Konzeptes](#), des Entwurfs. Architekturmodelle sind daher oft recht [abstrakt](#) und unterscheiden sich deutlich von Modellen, die bestehende [Bauwerke](#) abbilden. (Siehe dazu [Modellbau](#)).

Das Architekturmodell ist ein Arbeits- und Entwurfswerkzeug des [Architekten](#) und ein Teil der [Architekturdarstellung](#) wie die [Präsentationszeichnung](#). Anhand von Entwurfsmodellen können Architekten und Bauherren sehr schnell die [Kubatur](#) eines Entwurfes sowie die [räumlichen](#) Zusammenhänge erfassen und bewerten. Gerade für [Laien](#), die im Lesen von Architekturzeichnungen ungeübt sind, ist ein Modell oft anschaulicher als zweidimensionale Darstellungen.

Mit Hilfe von plastischen Architekturmodellen kann man verschiedene [Simulationen](#) durchführen. Die [Belichtung](#) und [Verschattung](#) von Gebäuden kann mittels einfacher Massenmodelle simuliert werden. Im [Windkanal](#) kann man die [Aerodynamik](#) bestimmter [Kubaturen](#) testen.

Typen von Architekturmodellen können vor allem nach ihrer Funktion unterschieden werden:

- *Arbeitsmodell:* Modell als Arbeitsmittel des [Architekten](#) zur Überprüfung eines [Entwurfes](#), oft sehr grob und schnell in preiswerten und leicht verarbeitbaren Materialien wie [Pappe](#) hergestellt.
- *Entwurfsmodell:* Modell zur [Darstellung](#) eines Entwurfes, stellt oft eine Zwischenstufe und Diskussionsgrundlage dar.
- *Wettbewerbsmodell:* Modell für die [Präsentation](#) eines Entwurfes bei einem [Wettbewerb](#).
- *Präsentationsmodell:* Sehr sorgfältig ausgearbeitetes Modell zur Präsentation eines [Entwurfes](#) für (potentielle) [Bauherren](#) oder Käufer einer [Immobilie](#) oder für die [Öffentlichkeit](#). Meist aufwändig, detailgetreu und aus hochwertigen Materialien.
- *Städtebauliches Modell:* Modell, das ein [Bauwerk](#) im [städtebaulichen Kontext](#) zeigt. Oft ausgeführt als *Massenmodell*
- *Massenmodell:* Stellt die [Baumassen](#) von [Baukörpern](#) dar. Einfache, massive Kuben repräsentieren die Bauwerke.

9.2 – CNC

CNC-Maschinen (Computerized Numerical Control) sind **Werkzeugmaschinen**, die durch den Einsatz von **Steuerungstechnik** in der Lage sind, Werkstücke mit hoher **Präzision** auch für komplexe Formen **automatisch** herzustellen.

Bei Verwendung eines **CAM-Systems** (Computer-aided manufacturing) können die Daten aus dem **CAD-Programm**, mit dem in der Regel die Bauteile konstruiert werden, unter Berücksichtigung einiger weiterer Faktoren wie **Geometrie der Werkzeuge**, Drehzahlen, Vorschüben mit Hilfe eines **Postprozessors** in ein CNC-Programm umgewandelt werden. Neben der manuellen Eingabe der Steuerinformationen gibt es weiterhin die sogenannte Teach-In-Methode, die für wenige dafür geeignete CNC-Maschinen zur Verfügung steht, und die **Werkstattprogrammierung** mit Eingabemasken.

Vorgänger der CNC-Maschinen waren die **NC-Maschinen**, also Werkzeugmaschinen mit **numerischer Steuerung** (*Numerical control*), z. B. durch Lochstreifen oder Lochkarten.

Taxonomie der CNC – Prozesse

- Subtraktive Prozesse z.B. Fräsen
- Transformative Prozesse (z.B. Gussprozesse)
- Deformations Prozesse (z.B. Biegen)
- Oberflächen Behandlung (z.B. Polieren oder Lackieren)
- Verbindungs Prozesse (z.B. Schweißen)
- Integrierte Prozesse (Automation und Robotik) (z.B. Pick&Place)
- Additive Prozesse / 3D Druck

9.3 – Rapid Prototyping

Rapid Prototyping (übersetzt *schneller Modellbau*) ist der Überbegriff über verschiedene Verfahren zur schnellen Herstellung von Musterbauteilen ausgehend von den Konstruktionsdaten.^[2]

Die meisten Verfahren des Rapid Prototyping – außer z. B. der Polyamidguss – sind der additiven Fertigung bzw. dem **3D-Druck** zuzurechnen, weil sie mit schichtweisem Materialauftrag und ohne Verwendung einer Form arbeiten. Rapid Prototyping bezeichnet die Art der Anwendung näher: Während **Rapid Tooling** die Herstellung von Werkzeugen und **Rapid Manufacturing** die Herstellung von Bauteilen und Fertigprodukten bezeichnet, bezeichnet Rapid Prototyping die Erstellung von Prototypen und Modellen.

9.3.1 – 3D Druck

Der **3D-Druck** (auch **3-D-Druck**),^[1] auch bekannt unter den Bezeichnungen **additive Fertigung** (englisch Additive Manufacturing, AM), **generative Fertigung** oder **Rapid-Technologien**,^[2] ist eine umfassende Bezeichnung für alle **Fertigungsverfahren**, bei denen Material Schicht für Schicht aufgetragen und so **dreidimensionale** Gegenstände (**Werkstücke**) erzeugt werden.

Dabei erfolgt der schichtweise Aufbau computergesteuert aus einem oder mehreren flüssigen oder festen **Werkstoffen** nach vorgegebenen Maßen und Formen (siehe **CAD/CAM**). Beim Aufbau finden **physikalische** oder **chemische** Härtings- oder **Schmelzprozesse** statt.

Typische Werkstoffe für das 3D-Drucken sind **Kunststoffe**, **Kunstharze**, **Keramiken**^[3] und speziell aufbereitete **Metalle**.^[4] Inzwischen wurden auch Carbon- und Graphitmaterialien für den 3D-Druck von Teilen aus Kohlenstoff entwickelt. Obwohl es sich oft um formende Verfahren handelt, sind für ein konkretes Erzeugnis keine speziellen Werkzeuge erforderlich, die die jeweilige Geometrie des Werkstückes gespeichert haben (zum Beispiel Gussformen).

3D-Drucker werden in der Industrie, im Modellbau und der Forschung eingesetzt und dort zur **Fertigung** von Modellen, Mustern, **Prototypen**, **Werkzeugen**, Endprodukten und für private Nutzung verwendet. Daneben gibt es Anwendungen im Heim- und Unterhaltungsbereich, dem Baugewerbe sowie in der Kunst und Medizin.^[5]

Digitale Prozesse

Daten und Modelle

10.1 – Technische Entwicklungen

10.1.1 – Datenwachstum

Mengen von Daten wachsen typischerweise **exponentiell**. Berechnungen aus dem Jahr 2011 zufolge verdoppelt sich das weltweite erzeugte **Datenvolumen** alle 2 Jahre.^[15] Diese Entwicklung wird vor allem getrieben durch die zunehmende maschinelle Erzeugung von Daten z. B. über Protokolle von Telekommunikationsverbindungen (**Call Detail Record**, CDR) und Webzugriffen (**Logdateien**), automatische Erfassungen von **RFID**-Lesern, **Kameras**, **Mikrofonen** und sonstigen Sensoren. Big Data fallen auch in der Finanzindustrie an (Finanztransaktionen, Börsendaten) sowie im Energiesektor (Verbrauchsdaten) und im Gesundheitswesen (Abrechnungsdaten der **Krankenkassen**). In der **Wissenschaft** fallen ebenfalls große Datenmengen an, z. B. in der **Geologie**, **Genetik**, **Klimaforschung** und **Kernphysik**.

10.1.2 – Big Data

Der aus dem englischen Sprachraum stammende Begriff **Big Data** (von **englisch** big ‚groß‘ und **data** ‚Daten‘, deutsch auch **Massendaten**) steht in engem Zusammenhang mit dem umfassenden Prozess der **Datafizierung** und bezeichnet **Datenmengen**, welche beispielsweise zu groß, zu komplex, zu schnelllebig oder zu schwach **strukturiert** sind, um sie mit manuellen und herkömmlichen Methoden der **Datenverarbeitung** auszuwerten.^[1]

„Big Data“ wird häufig als **Sammelbegriff** für digitale Technologien verwendet, die in technischer Hinsicht für eine neue Ära **digitaler Kommunikation und Verarbeitung** und in sozialer Hinsicht für einen gesellschaftlichen Umbruch verantwortlich gemacht werden.^[2] Dabei unterliegt der Begriff als **Schlagwort** einem kontinuierlichen Wandel; so wird damit ergänzend auch oft der Komplex der **Technologien** beschrieben, die zum Sammeln und Auswerten dieser Datenmengen verwendet werden.

10.1.3 – Cloud Computing

Cloud Computing (deutsch *Rechnerwolke* oder *Datenwolke*^[1]) beschreibt ein Modell, das bei Bedarf – meist über das Internet und geräteunabhängig – zeitnah und mit wenig Aufwand geteilte **Computerressourcen** als Dienstleistung, etwa in Form von **Servern**, **Datenspeicher** oder **Applikationen**, bereitstellt und nach Nutzung abrechnet. Angebot und Nutzung dieser Computerressourcen ist definiert und erfolgt in der Regel über eine **Programmierschnittstelle** (API) bzw. für Anwender über eine **Website** oder **App**.^{[2][3]}

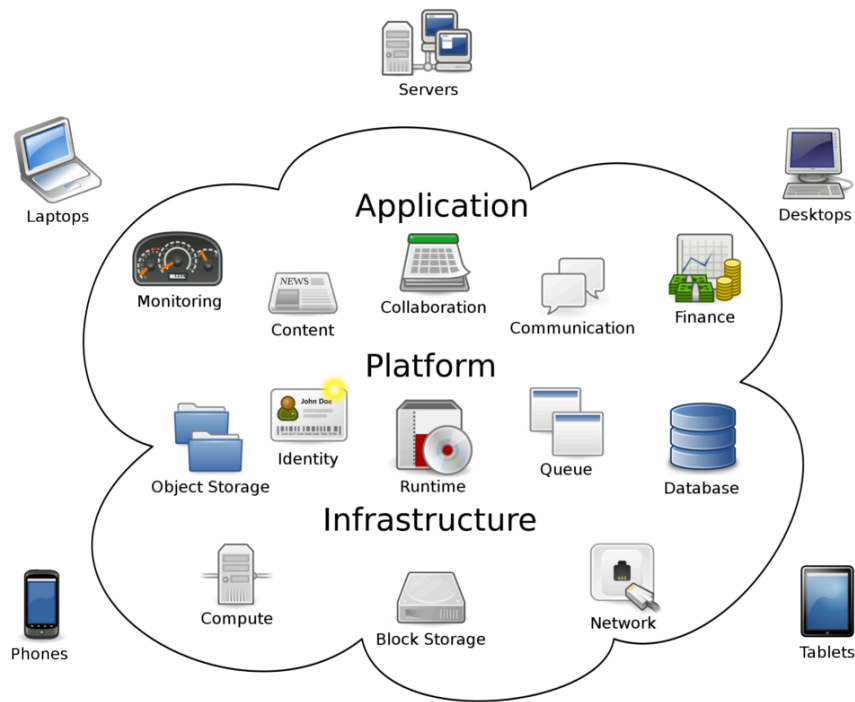


Abbildung 10.1 – Diagramm, das einen Überblick über das Cloud Computing mit den typischen Anwendungstypen zeigt, die von diesem Computing-Modell unterstützt werden.

10.1.4 – Physische Präsenz von Daten

Ein Rechenzentrum oder Datenzentrum ist ein Gebäude, ein bestimmter Bereich innerhalb eines Gebäudes oder eine Gruppe von Gebäuden, die zur Unterbringung von Computersystemen verwendet werden. Da der IT-Betrieb für die Geschäftskontinuität entscheidend ist, umfasst er im Allgemeinen redundante oder Backup-Komponenten und Infrastruktur für Stromversorgung, Datenkommunikationsverbindungen, Umgebungskontrollen und verschiedene Sicherheitsvorrichtungen. Ein großes Rechenzentrum ist ein industrieller Betrieb, der so viel Strom verbraucht wie eine Kleinstadt. Der Energieverbrauch ist ein zentrales Thema für Rechenzentren. Die Leistungsaufnahme reicht von wenigen Kilowatt für ein Serverrack in einem Schrank bis zu vielen Megawatt für große Einrichtungen. Einige Einrichtungen haben eine Leistungsdichte, die mehr als das 100-fache der eines typischen Bürogebäudes beträgt. Bei Einrichtungen mit höherer Leistungsdichte sind die Stromkosten eine dominierende Betriebsausgabe und machen über 10 % der Gesamtbetriebskosten eines Rechenzentrums aus.

Energieverbrauch Rechenzentren

Im Jahr 2020 verbrauchten Rechenzentren (ohne Kryptowährungs-Mining) und Datenübertragung jeweils etwa 1 % des weltweiten Stroms. Obwohl ein Teil dieses Stroms kohlenstoffarm war, forderte die IEA (International Energy Agency) mehr „Anstrengungen von Regierung und Industrie in Bezug auf Energieeffizienz, Beschaffung erneuerbarer Energien und RD&D (Research, Development and Demonstration)“, da einige Rechenzentren immer noch Strom aus fossilen Brennstoffen nutzen. Es wird geschätzt, dass Rechenzentren im Jahr 2018 für 0,5 % der Treibhausgasemissionen in den USA verantwortlich waren.



Abbildung 10.2 – Innenaufnahme eines Datenzentrums

10.2 – Theoretische Grundlagen

10.2.1 – Daten

Daten bezeichnet als Plural von Datum Fakten, Zeitpunkte oder kalendarische Zeitangaben und als Pluralwort gemeinsprachlich die durch Beobachtungen, Messungen u. ä. gewonnenen Zahlenwerte und darauf beruhenden Angaben oder formulierbaren Befunde.[1] Während Daten in der Umgangssprache Gegebenheiten, Tatsachen oder Ereignisse sind, sind Daten in der Fachsprache Zeichen, die eine Information darstellen.[2]

In verschiedenen [Fachgebieten](#) wie z. B. der [Informatik](#), der [Mathematik](#), der [Wirtschaftstheorie](#), der [Neurowissenschaft](#) oder den [Biowissenschaften](#) sind unterschiedliche – meist ähnliche – Definitionen gebräuchlich. Eine einheitliche Definition gibt es bisher nicht.[3][4] Das liegt auch daran, dass die verschiedenen Fachgebiete dem Begriff Daten einen unterschiedlichen [Begriffsinhalt](#) zuordnen, der nur dieses Fachgebiet betrifft. Daten sind nur Darstellungen/Angaben über Sachverhalte und Vorgänge, die in der Form bestimmter Zeichen/Symbole auf bestimmten Datenträgern existieren. Aus ihnen kann (bei Menschen durch kognitive Tätigkeiten des Empfängers) „Information“ werden, zweckbezogenes Wissen, das man beim Handeln im Hinblick auf gesetzte Ziele benötigt.

10.2.2 – Information

Information kann bewusst als [Nachricht](#) oder Botschaft von einem Sender an einen Empfänger übermittelt oder auch unbewusst transportiert werden und durch die [Wahrnehmung](#) von Form und Eigenschaft eines Objektes auffallen. Information erhält ihren Wert durch die [Interpretation](#) des Gesamtgeschehens auf verschiedenen Ebenen durch den Empfänger der Information. Sender oder Empfänger können nicht nur Personen/Menschen, sondern auch (höher-entwickelte) Tiere oder künstliche Systeme (wie Maschinen oder Computer/Computerprogramme) sein.

Information wird auch synonym für Nachricht, Auskunft, Belehrung, Aufklärung verwendet, die aber i. e. S. nur die „Träger von Informationen“, nicht die Information selbst sind.

Kommunikation Die Kommunizierbarkeit gilt als eine wesentliche Eigenschaft von Information, und jegliche Kommunikation setzt Information voraus.



Abbildung 10.3 – Claude Elwood Shannon (1916 – 2001) war ein Amerikanischer Mathematiker, Elektroingenieur und Kryptograph, bekannt als ein “Vater der Informationstheorie”

10.2.3 – Wissen

Der Begriff der Information ist eng verknüpft mit Fragestellungen im Themenkomplex Wissen. Dazu gehört insbesondere das Problem der Definition von Komplexität. Weiterhin zählen hierzu Betrachtungen über den Unterschied zwischen Zufall und Ordnung sowie der Begriff der Unterscheidbarkeit und der Relevanz.

10.2.4 – Daten, Information, Wissen, Weisheit

Die Wissenshierarchie (Informationshierarchie, Informationspyramide) bezieht sich lose auf eine Klasse von Modellen zur Darstellung von Strukturen und/oder Funktionen bzw. Beziehungen zwischen Daten, Informationen, Wissen und Weisheit. „Typischerweise wird Information in Form von Daten definiert, Wissen in Form von Information und Weisheit in Form von Wissen“.

10.2.5 – Data Mining

Unter Data-Mining (von [englisch](#) data mining, aus [englisch](#) data ‚Daten‘ und [englisch](#) mine ‚graben‘, ‚abbauen‘, ‚fördern‘)[1] versteht man die systematische Anwendung statistischer Methoden auf große Datenbestände (insbesondere „[Big Data](#)“ bzw. Massendaten) mit dem Ziel, neue [Querverbindungen und Trends](#) zu erkennen. Solche Datenbestände werden aufgrund ihrer Größe mittels computergestützter Methoden verarbeitet. In der Praxis wurde der Unterbegriff Data-Mining auf den gesamten Prozess der sogenannten „[Knowledge Discovery in Databases](#)“ ([englisch](#) für Wissensentdeckung in Datenbanken; KDD) übertragen, der auch Schritte wie die Vorverarbeitung und Auswertung beinhaltet, während Data-Mining im engeren Sinne nur den eigentlichen Verarbeitungsschritt des Prozesses bezeichnet.[2]

10.2.6 – Ontologie

Die Ontologie (im 16. Jahrhundert als griechisch ὀντολογία ontología gebildet aus [altgriechisch](#) ὄν ὄν ‚seiend‘ bzw. [altgriechisch](#) τὸ ὄν ‚das Sein‘ und λόγος lógos ‚Lehre‘, also ‚Lehre vom Seienden‘ bzw. ‚Lehre des Seins‘) ist eine [Disziplin](#) der (theoretischen) [Philosophie](#), die sich mit der Einteilung des Seienden und den Grundstrukturen der [Wirklichkeit](#) befasst, z.

B. mit Begriffen wie [Existenz](#), [Sein](#), [Werden](#) und [Realität](#).

Ontologien in der Informatik sind meist sprachlich gefasste und formal geordnete Darstellungen einer Menge von Begriffen und der zwischen ihnen bestehenden Beziehungen in einem bestimmten Gegenstandsbereich (in Anlehnung an den klassischen Begriff der Ontologie). Sie werden dazu genutzt, „Wissen“ in digitalisierter und formaler Form zwischen Anwendungsprogrammen und Diensten auszutauschen. Wissen umfasst dabei sowohl Allgemeinwissen als auch Wissen über sehr spezielle Themengebiete und Vorgänge.

Im Bauwesen existieren bereits mehrere Ontologien verschiedener Domänen und Fachrichtungen, die unter anderem durch die Linked Building Data Community Group und durch buildingSMART erarbeitet werden.

10.2.7 – Semantik

Semantik (von altgriechisch σημαίνειν sēmaínein, deutsch ‚bezeichnen, ein Zeichen geben‘), auch Bedeutungslehre, nennt man die Theorie oder Wissenschaft von der Bedeutung der Zeichen. Zeichen können hierbei beliebige Symbole sein, insbesondere aber auch Sätze, Satzteile, Wörter oder Wortteile. Soweit sich die Semantik mit Zeichen aller Art befasst, ist sie ein Teilbereich der Semiotik. Sofern sie sich allein mit sprachlichen Zeichen befasst, ist sie eine Teildisziplin der Linguistik. Allgemeinsprachlich wird unter Semantik auch einfach die Bedeutung eines bestimmten Wortes, Satzes oder Textes verstanden.[1]

Ein Semantisches Datenmodell (SDM, englisch auch conceptual schema) ist im Rahmen der Datenmodellierung eine abstrakte, formale Beschreibung und Darstellung eines Ausschnittes der in einem bestimmten Zusammenhang (z. B. eines Projekts) „wahrgenommenen Welt“. Für Gebäudedaten existiert zum Beispiel der IFC Standard als sematisches Datenmodell.

10.3 – Digitale Datenstrukturen

10.3.1 – Geographische Informations Systeme

Geoinformationssysteme, Geographische Informationssysteme (GIS) oder Räumliche Informationssysteme (RIS) sind [Informationssysteme](#) zur Erfassung, Bearbeitung, Organisation, Analyse und Präsentation [räumlicher Daten](#). Geoinformationssysteme umfassen die dazu benötigte Hardware, Software, Daten und Anwendungen.

10.3.2 – Gebäude Informations Modelle

Der Begriff Building Information Modeling (kurz: BIM; deutsch: Bauwerksdatenmodellierung) beschreibt eine Arbeitsmethode für die vernetzte Planung, den Bau und die Bewirtschaftung von [Gebäuden](#) und anderen Bauwerken mithilfe von [Software](#). Dabei werden alle relevanten Bauwerksdaten digital modelliert, kombiniert und erfasst. Das Bauwerk ist als virtuelles Modell auch geometrisch visualisiert ([Computermode](#)ll). Building Information Modeling findet Anwendung sowohl im [Bauwesen](#) zur [Bauplanung](#) und [Bauausführung](#) ([Architektur](#), [Ingenieurwesen](#), [Haustechnik](#), [Tiefbau](#), [Städtebau](#), [Eisenbahnbau](#), [Straßenbau](#), [Wasserbau](#), [Geotechnik](#)) als auch im [Facilitymanagement](#)[1].

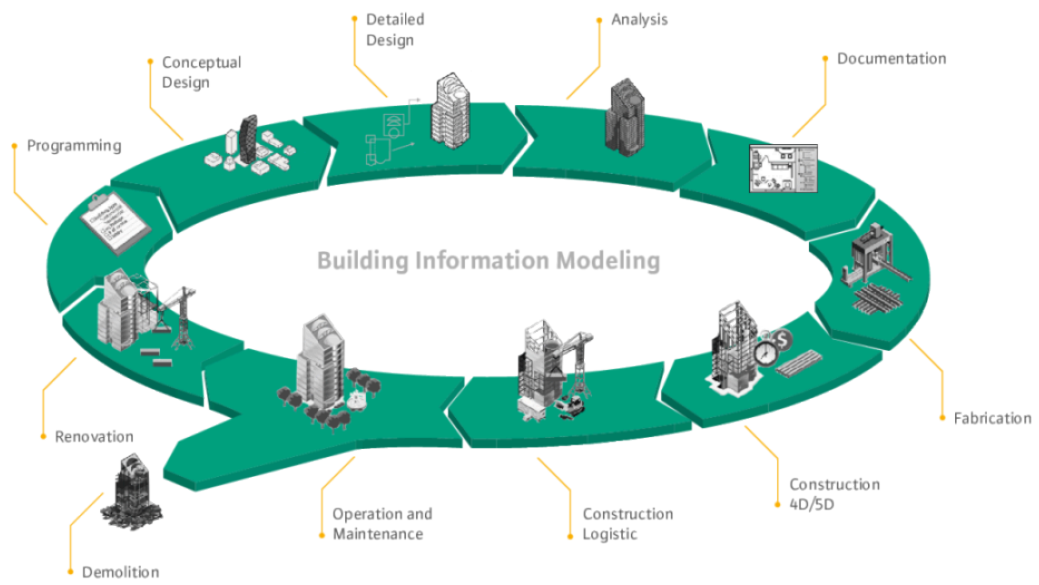
Die vielleicht grundlegendste und wichtigste Feststellung ist, dass BIM keine Software und kein Werkzeug oder gar eine Reihe von Werkzeugen ist. Building Information Modeling umfasst drei Hauptaspekte:

- Ein semantisches Datenmodell, das durchaus aus mehreren Modellen bestehen kann, die über ein Referenzmodell miteinander verknüpft sind.
- Der Modellierungsprozess, der sich über längere Zeiträume erstrecken kann und an dem Mitwirkende und Mitarbeiter aus vielen verschiedenen Bereichen beteiligt sind.
- Management des Modells: die Planung und Koordination von Daten- und Informationsflüssen.

10.3.3 – Industry Foundation Classes

Die Industry Foundation Classes sind ein offener Standard im Bauwesen zur digitalen Beschreibung von Gebäudemodellen. Industry Foundation Classes (IFC). Sie haben Ihre Wurzeln im ISO 10303 standard, aber basieren auf mitte-1990er Jahre auf STEP, Standard for the Exchange of Product Model Data.

10.4 – Digitale Prozessketten



Code

11.1 – Algorithmus

Ein **Algorithmus** ist eine eindeutige Handlungsvorschrift zur Lösung eines Problems oder einer Klasse von Problemen. Algorithmen bestehen aus endlich vielen, wohldefinierten Einzelschritten.^[1] Damit können sie zur Ausführung in ein Computerprogramm implementiert, aber auch in menschlicher Sprache formuliert werden. Bei der Problemlösung wird eine bestimmte Eingabe in eine bestimmte Ausgabe überführt.^[2]

1. Der Algorithmus muss bei denselben Voraussetzungen das gleiche Ergebnis liefern (**Determiniertheit**).
2. Die nächste anzuwendende Regel im Verfahren ist zu jedem Zeitpunkt eindeutig definiert (**Determinismus**).

11.2 – Programmiersprachen

Eine **Programmiersprache** ist eine formale Sprache zur Formulierung von Datenstrukturen und Algorithmen, d. h. von Rechenvorschriften, die von einem Computer ausgeführt werden können.^[1] Sie setzen sich üblicherweise aus schrittweisen Anweisungen aus erlaubten (Text-)Mustern zusammen, der sogenannten **Syntax**.

Während die ersten Programmiersprachen noch unmittelbar an den Eigenschaften der jeweiligen Rechner ausgerichtet waren, werden heute meist problemorientierte oder auch (allgemeiner) höhere Programmiersprachen verwendet, die eine maschinenunabhängigere^[2] und somit für den Menschen leichter verständliche Ausdrucksweise erlauben. In diesen Sprachen geschriebene Programme können automatisiert in Maschinensprache übersetzt werden, welche unmittelbar von einem Prozessor ausgeführt werden können.

Programmiersprachen bieten meist mindestens

- Ein-/Ausgabe-Befehle, damit das Programm Daten entgegennehmen und wieder ausgeben kann;
- Deklaration von Variablen und Feldern, um Informationen zwischenspeichern zu können;
- mathematische Grund- und Standardfunktionen;
- Grundfunktionen zur Zeichenkettenverarbeitung;
- Steueranweisungen für bedingte Ausführung, Wiederholung, Programmunterteilung (z. B. in Unterfunktionen) sowie Einbinden von Bibliotheken.

11.3 – Variablen

Mit Isaac Newton setzte sich die Vorstellung von „fließenden Größen“ (Fluenten) durch, die aber erst durch Leonhard Euler ausdrücklich formuliert wurde. David Hilbert entwickelte die moderne Vorstellung einer Variablen, die einfach nur ein Symbol für ein beliebiges Objekt jeglicher Art ist und nicht nur für ein mathematisches Objekt (Zahlen, Punkte, Geraden, Ebenen, Vektoren, Vektorräume etc.)

- **Mathematik:** Variablen sind Platzhalter für Unbekannte/Unbestimmte/Veränderliche in Formulierungen oder Formeln der Mathematik;^[1] Rechengröße, die unterschiedliche Werte annehmen kann
- **Informatik:** Variablen sind Bezeichnung eines Speicherplatzes, der verschiedene Werte aufnehmen kann

Eine **Variable** ist ein Name für eine Leerstelle in einem logischen oder mathematischen Ausdruck.^[1] Der Begriff leitet sich vom lateinischen Adjektiv *variabilis* (veränderlich) ab. Gleichwertig werden auch die Begriffe Platzhalter oder Veränderliche benutzt. Als „Variable“ dienten früher Wörter oder Symbole, heute verwendet man zur mathematischen Notation in der Regel

Buchstaben als Zeichen. Wird anstelle der Variablen ein konkretes Objekt eingesetzt, so ist „darauf zu achten, dass überall dort, wo die Variable auftritt, auch das gleiche Objekt benutzt wird.“^[1]

Ein **Formelzeichen** steht in der **Physik** und den **Ingenieurwissenschaften** für eine nicht notwendig numerisch festgelegte oder für eine zumindest anfangs noch veränderliche **physikalische Größe** oder **Zahl**. Die Formelzeichen für Größen sind im Allgemeinen einzelne Buchstaben, bei Bedarf ergänzt durch Indices oder andere modifizierende Zeichen.^{[2][3]}

Die Variablen, die in einer Gleichung vorkommen, nannte man in den Schulbüchern der Mathematik bis in die 1960er Jahre auch *Unbekannte* oder *Unbestimmte*.^[4] Beim Zusammentreffen mehrerer Variabler unterscheidet man abhängige und unabhängige Variable, aber nur, wenn ein Zusammenhang zwischen den Variablen besteht. Alle unabhängigen Variablen gehören zu einer **Definitions Menge** oder einem Definitionsbereich, die davon abhängigen zu einer **Wertemenge** oder einem Wertebereich.^{[5][6]}

In der **Programmierung** ist eine **Variable** ein abstrakter Behälter für eine Größe, welche im Verlauf eines **Rechenprozesses** auftritt. Im Normalfall wird eine Variable im **Quelltext** durch einen Namen bezeichnet und hat eine **Adresse** im Speicher einer Maschine.

Der durch eine Variable repräsentierte Wert und gegebenenfalls auch die Größe kann – im Unterschied zu einer **Konstante** – zur **Laufzeit** des Rechenprozesses verändert werden.

11.3.1 – Listen, Felder, Arrays

Ein **Feld** (**englisch** field, array) ist in der **Informatik** eine **Datenstruktur**-Variante, mit deren Verwendung „viele gleichartig strukturierte Daten [...] verarbeitet werden sollen“. ^[1] Der Zugriff auf bestimmte Inhalte eines Felds erfolgt mit Hilfe von **Indizes**, die dessen Position bezeichnen.

11.4 – Wiederholung

Eine **Schleife** (auch „Wiederholung“ oder **englisch** loop) ist eine **Kontrollstruktur** in **Programmiersprachen**. Sie wiederholt einen **Anweisungs-Block** – den sogenannten *Schleifenrumpf* oder *Schleifenkörper* –, solange die Schleifenbedingung als Laufbedingung^[Anm 1] gültig bleibt bzw. als **Abbruchbedingung** nicht eintritt. Schleifen, deren Schleifenbedingung immer zur Fortsetzung führt oder die keine Schleifenbedingung haben, sind **Endlosschleifen**.

FOR *Iterator:=Anfangszahl* **TO** *Endezahl* **STEP** *Schrittweite* **DO** *Schleifenrumpf*.

Bei einer **For-Schleife** zählt der **Computer** von einer Anfangszahl bis zu einer Endzahl und wiederholt dabei jedes Mal den Codeblock („Schleifenrumpf“). Die aktuelle Zahl wird in eine Variable („Iterator“) gesetzt, damit sie bei Bedarf in dem Codeblock Verwendung finden kann. Häufig ist die Zählschleife auf Ganzzahlen beschränkt. Das Ändern der Iterator-Variablen im Schleifenkörper ist bei vielen Programmiersprachen verboten und gilt als schlechter Programmierstil, da es oft zu schwer verständlichem Code führt – es läuft der Denkweise zuwider, direkt am Schleifenkopf die Anzahl der Durchläufe ablesen zu können.

11.5 – Bedingungen

Eine **bedingte Anweisung** ist eine **Kontrollstruktur** in der **Programmierung**. Ein Programmabschnitt wird dabei nur unter einer bestimmten Bedingung ausgeführt. Eine **Verzweigung** legt fest, welcher von zwei oder mehreren Programmabschnitten, abhängig von einer oder mehreren Bedingungen, ausgeführt wird. Bedingungen werden in Programmiersprachen meist mit Konstrukten wie „if...then“ (deutsch: Wenn.. Dann...) eingeführt.

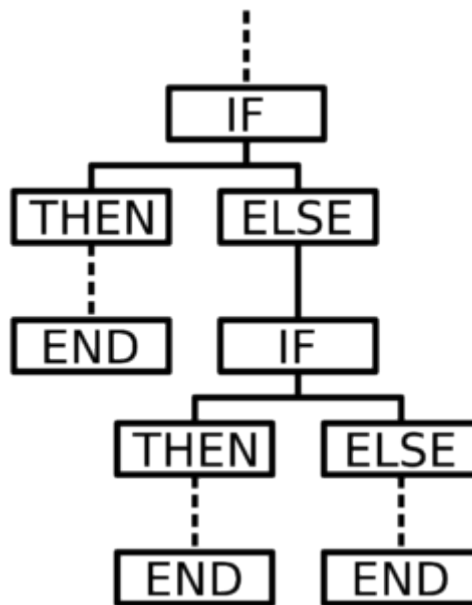


Abbildung 11.1 – Detailliertes Diagramm zur Verwendung von „Wenn-Dann“ Kontrollstrukturen

11.6 – Bildtafeln

Mother Tongues

Tracing the roots of computer languages through the ages

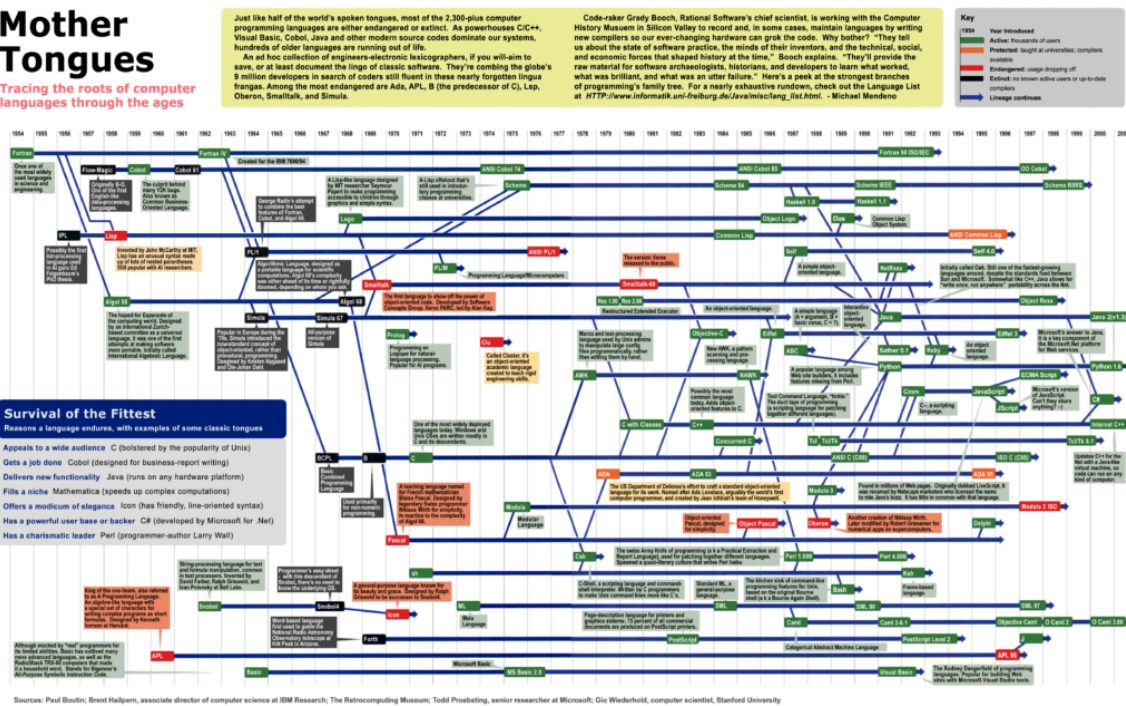


Abbildung 11.2 – Computersprachen durch den Verlauf der Zeit

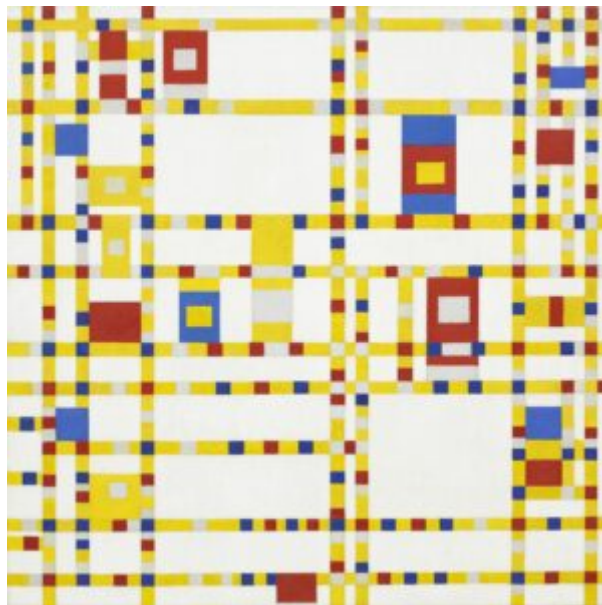


Abbildung 11.3 – Piet Mondrian – Boogie Woogie

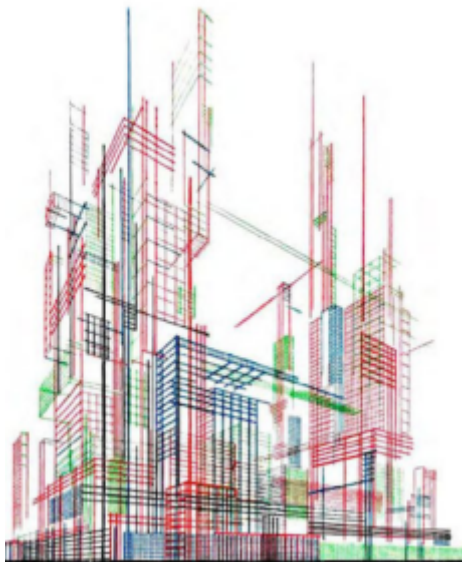


Abbildung 11.4 – Chernikhov – Fantasy and Construction

Parametrisches und Procedurales Design

12.1 – Parameter

Ein Parameter oder auch eine Formvariable ist eine an sich unabhängige Variable, die aber zumindest in einer gegebenen Situation eher als eine festgehaltene Größe aufgefasst wird.

12.2 – Parametrisches Design

Parametrisches Design ist eine Designmethode, bei der Merkmale (wie Gebäudeelemente und technische Komponenten) nach algorithmischen Prozessen geformt werden, anstatt direkt modelliert zu werden. Bei dieser Methode bestimmen Parameter und Regeln die Beziehung zwischen Designabsicht und Designantwort. Der Begriff parametrisch bezieht sich auf Eingabeparameter, die in die Algorithmen eingespeist werden.

Das Potential von parametrischem Design ist vielfältig:

- Fehler vermeiden (präzises, numerisches Modellieren)
- Schnelles Anpassen von Design
- Modellieren von komplexer Geometrie
- Generieren von individualisierten Elementen
- Generieren von Design-variationen
- Generieren von Familien von Elementen
- Generieren von Exportdaten z.B. zur Fabrikation

Gebäude können von der ersten Entwurfsphase an auf ein Optimum (z. B. Zeit, Kosten, Qualität) in der Planung, Vorfertigung, Ausführung und Nutzung hin konzipiert werden. Parametrische Planung ist bei komplexen Projekten mit vielen Akteuren besonders sinnvoll, da Änderungen im Planungsverlauf problemlos in das parametrische Modell eingearbeitet werden können. Verschiedene Entwurfsvarianten erleichtern somit grundlegende Entscheidungen in der Anfangsphase von Planungen. Die durchgängige Parametrisierung eines einzelnen Produkts bzw. Bauteils inklusive seines „Zubehörs“ (Formwerkzeuge, Fertigungsverfahren, Materialeigenschaften etc.) stellt eine geschlossene digitale Prozesskette sicher.

Parametrismus ist ein Stil innerhalb der zeitgenössischen Avantgarde-Architektur, der als Nachfolger der modernen und postmodernen Architektur gefördert wird. Der Begriff wurde 2008 von Patrik Schumacher, einem Architekturpartner von Zaha Hadid (1950–2016), geprägt. Parametrismus hat seinen Ursprung im parametrischen Design, das auf den Einschränkungen in einer parametrischen Gleichung basiert. Parametrismus stützt sich auf Programme, Algorithmen und Computer, um Gleichungen für Designzwecke zu manipulieren.

Aspekte des Parametrismus wurden in Stadtplanung, Architekturdesign, Innenarchitektur und Möbeldesign verwendet. Befürworter des Parametrismus haben erklärt, dass eines der bestimmenden Merkmale darin besteht, dass „Parametrismus impliziert, dass alle Elemente des Designs parametrisch variabel und wechselseitig anpassungsfähig werden.“ Laut Schumacher ist Parametrismus eine Autopoiesis oder ein selbstreferenzielles System bei dem alle Elemente miteinander verbunden sind und ein äußerer Einfluss, der eines verändert, alle anderen verändert.

Der Parametrismus lehnt sowohl die Homogenisierung (serielle Wiederholung) als auch die reine Differenz (Agglomeration nicht verwandter Elemente) zugunsten von Differenzierung und Korrelation als zentrale Kompositionswerte ab. Ziel ist es, mehr räumliche Komplexität bei gleichzeitiger Lesbarkeit aufzubauen, d. h. Beziehungen zwischen Räumen (oder Elementen einer Komposition) zu intensivieren und sich so an Kontexte anzupassen, dass lesbare Verbindungen hergestellt werden. Dies ermöglicht der Architektur, die Komplexität zeitgenössischer Lebensprozesse in die globale postfordistische Netzwerkgesellschaft zu übersetzen.

12.3 – Mapping

Ein zentrales Konzept der Mathematik ist die *Abbildung*, die auch *Funktion* genannt wird (engl. Mapping). Abbildungen sind eindeutige Zuordnungen zwischen zwei Mengen **A** und **B**. Dies bedeutet, dass jedem Element $x \in A$ durch die Abbildung f genau ein Element $f(x) \in B$ zugeordnet wird. Ein Beispiel hierfür ist die Quadratfunktion von der Menge $\mathbb{R}$ in die Menge $\mathbb{R}$, die jeder reellen Zahl $x \in \mathbb{R}$ ihre Quadratzahl $x^2 \in \mathbb{R}$ zuordnet. Die Schreibweise für Abbildungen von **A** nach **B** ist: $f : A \rightarrow B : x \mapsto f(x)$

Ausgesprochen wird dieser Ausdruck so:

f : $A \rightarrow B$: $x \mapsto f(x)$
 Abbildung f aus A nach B , die x auf $f(x)$ abbildet

Hinweis

Die Begriffe „Abbildung“ und „Funktion“ sind beide in der Mathematik üblich und bedeuten genau dasselbe.

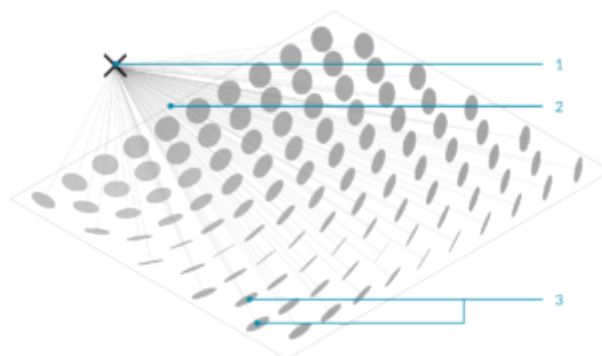
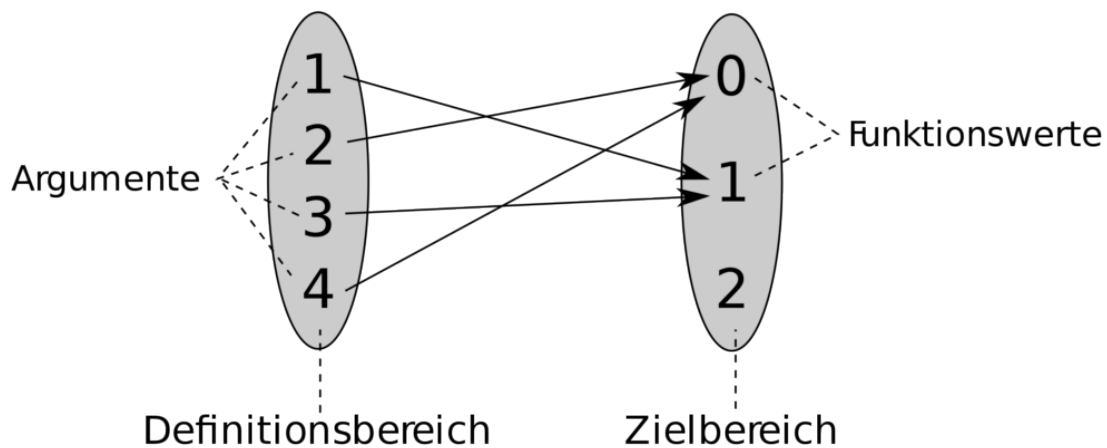


Abbildung 12.13 – 1: Attraktor Punkt, 2.:Vektoren, 3: Kreise zu dem Attraktor orientiert, basierend auf deren Normalen. In diesem Beispiel wird die Orientierung zum Attraktor auf die Rotation der Kreise abgebildet.

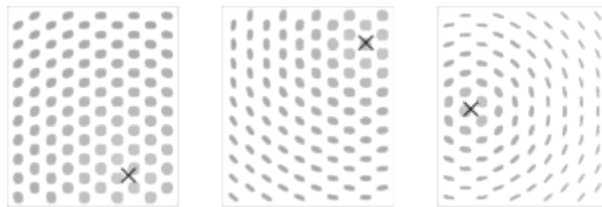


Abbildung 12.14 - Verschiedene Positionen des Attraktors

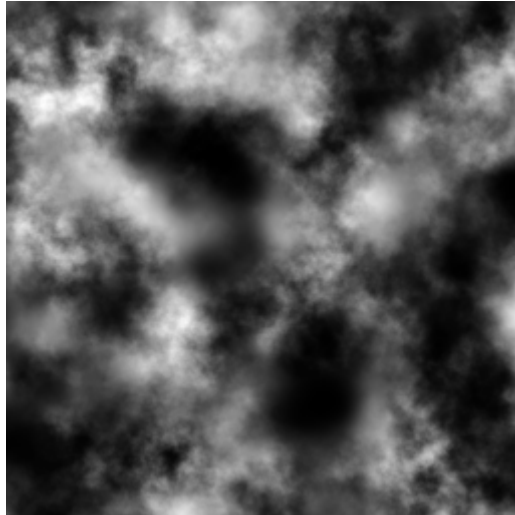


Abbildung 12.15 - Höhenfeld (Heightmap)

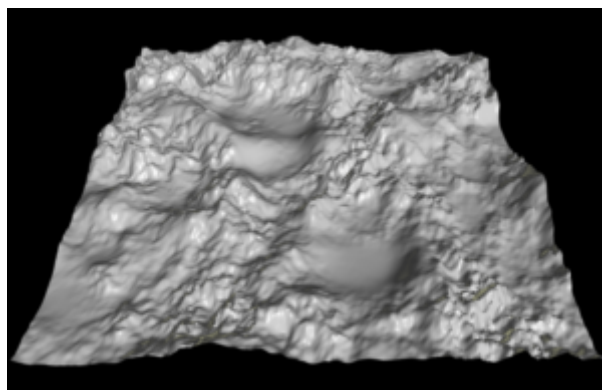


Abbildung 12.16 - Mesh, deformiert durch Höhenfeld (Heightmap)

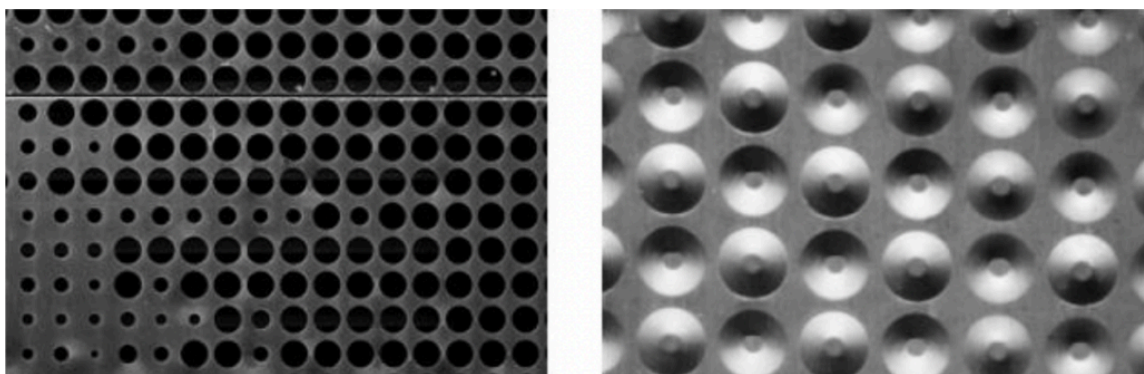


Abbildung 12.17 - Lochgrösse und Extrusionstiefe, De Young Museum.

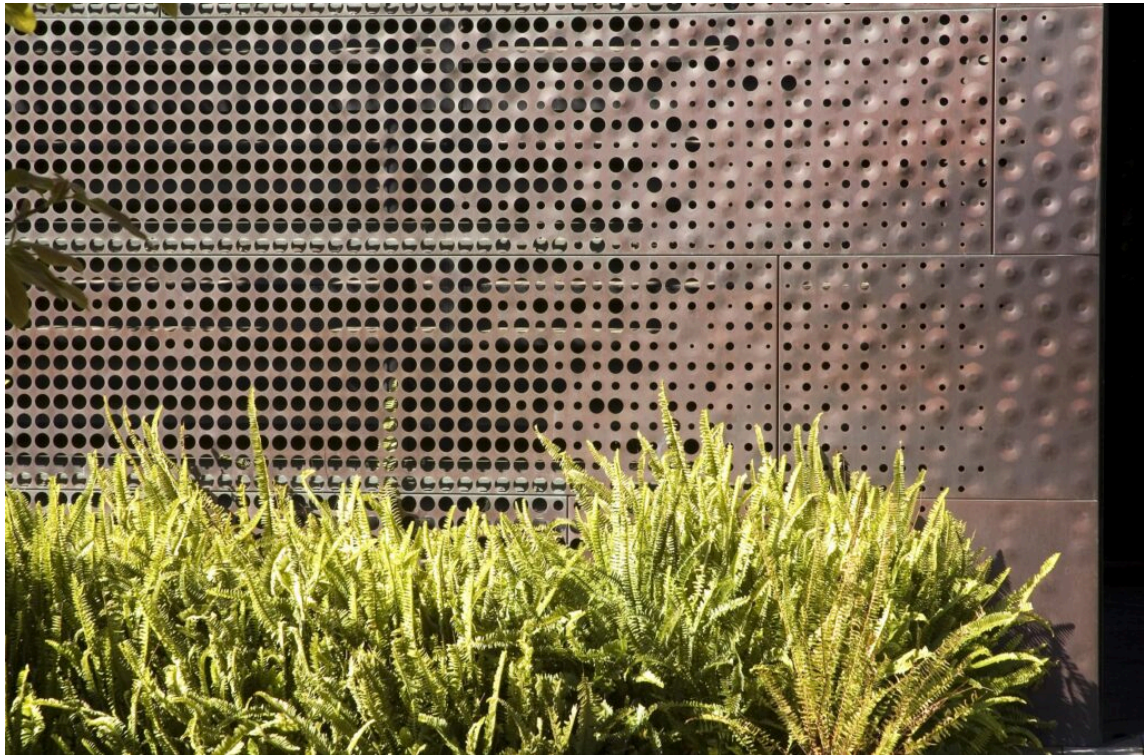


Abbildung 12.18 – De Young Museum, Herzog & DeMeuron

12.4 – Rekursion

Als Rekursion ([lateinisch](#) recurrere ‚zurücklaufen‘) wird ein prinzipiell unendlicher Vorgang, der sich selbst als Teil enthält oder mithilfe von sich selbst definierbar ist, bezeichnet.^[1] Üblicherweise sind rekursive Vorgänge relativ kurz beschreibbar bzw. können durch eine relativ kurze Anweisung ausgelöst werden.^{[2][3]} Die bei Rekursion aufeinander folgenden Teilvorgänge oder die nacheinander erzeugten Objekte sind nicht unabhängig voneinander, sondern zwischen jedem Schrittpaar oder Objektpaar besteht eine besondere, die rekursive Beziehung.

12.5 – Fraktal

Fraktal ist ein vom Mathematiker [Benoît Mandelbrot](#) 1975 geprägter Begriff ([lateinisch](#) fractus ‚gebrochen‘, von [lateinisch](#) frangere, (in Stücke zer-),brechen‘), der bestimmte natürliche oder künstliche Gebilde oder geometrische [Muster](#) bezeichnet. Gebilde weisen einen hohen Grad von [Skaleninvarianz](#) bzw. [Selbstähnlichkeit](#) auf. Das ist beispielsweise der Fall, wenn ein Objekt aus mehreren verkleinerten Kopien seiner selbst besteht. Geometrische Objekte dieser Art unterscheiden sich in wesentlichen Aspekten von gewöhnlichen glatten Figuren.

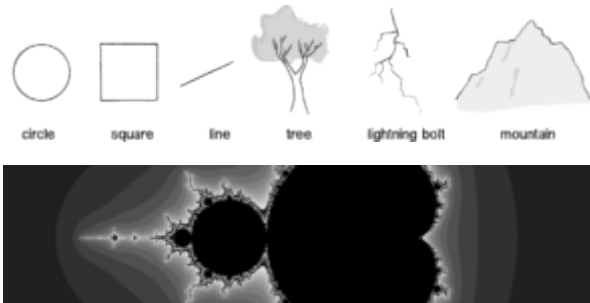


Abbildung 12.20 – Mandelbrot Menge



Abbildung 12.21 – Börsendaten im Zeitraum von 6 Monaten



Abbildung 12.22 – Börsendaten im Zeitraum von 6 Stunden

12.6 – L-System

Bei den **Lindenmayer-** oder **L-Systemen** handelt es sich um einen [mathematischen Formalismus](#), der 1968 von dem ungarischen [theoretischen Biologen Aristid Lindenmayer](#) als Grundlage einer [axiomatischen Theorie biologischer Entwicklung](#) vorgeschlagen wurde. In jüngerer Zeit fanden L-Systeme Anwendung in der [Computergrafik](#) bei der Erzeugung von [Fraktalen](#) und in der realitätsnahen [Modellierung](#) von [Pflanzen](#).

Das wesentliche Prinzip von L-Systemen besteht in der sukzessiven Ersetzung von Einzelteilen eines einfachen Objektes mittels sogenannter [Produktionsregeln](#). Diese Ersetzungen können [rekursiv](#) durchgeführt werden. Damit gehören L-Systeme zu den sogenannten [Ersetzungssystemen](#).

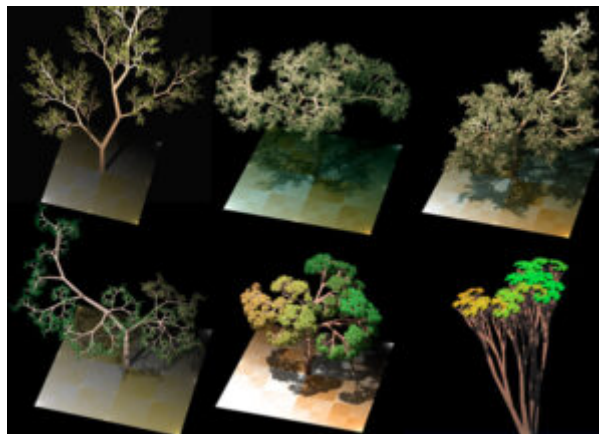


Abbildung 12.23 – Fraktale Bäume, mit einem L-System generiert.

12.7 – Shape Grammar

Shape grammars, deutsch **Formen-Grammatiken**, sind eine Art von [Produktionssystem](#) in der Informatik zur Generierung von geometrischen Formen. Üblicherweise werden Shape grammars zur Erzeugung von zwei- bzw. dreidimensionalen Formen eingesetzt, heutzutage vor allem in den Bereichen der [Architektur](#) und [Computergrafik](#). Die Grundlage für Shape grammars wurde in einem Seminar-Artikel von [George Stiny](#) und James Gips im Jahr 1971 geschaffen.^[1]

Eine Shape grammar besteht aus Regeln sowie einer Generierungs-[Engine](#), welche die Regeln auswählt und verarbeitet bzw. berechnet. Eine Regel definiert wie eine existierende Figur (bzw. ein Teil davon) im Geometrischen Raum transformiert werden kann. Die Definition einer Shape grammar folgt der Standarddefinition einer [Phrasenstrukturgrammatik](#) von Chomsky, wobei statt Symbolen, Figuren („shapes“) verwendet werden.

Eine Shape grammar generiert eine Form durch rekursives Anwenden der Shape rules, beginnend mit der Startfigur. Das Ergebnis der angewendeten Regel auf eine vorhandene Form, ist immer eine neue Form, die aus der vorhandenen Form besteht, wobei ein Vorkommnis der linken Seite der Regel in der neuen Form durch die Form auf der rechten Seite der Regel ersetzt wurde.

Eine Shape grammar besteht minimal aus drei Produktionsregeln (=Shape rules). Einer Anfangsregel, zumindest einer Transformationsregel und einer Terminierungsregel. Die Startregel ist erforderlich um den Generierungsprozess zu starten, während die Terminierungsregel notwendig ist um die Generierung abzuschließen. Der einfachste Weg den Prozess zu stoppen, ist es den Marker (=Nonterminale) zu entfernen. Im Gegensatz zu Chomsky-Grammatiken können Produktionsregeln in Shape grammars nicht nur seriell, sondern auch parallel angewendet werden, ähnlich dem Ablauf von [L-Systemen](#).

Shape grammars wurden ursprünglich für Gemälde und Skulpturen präsentiert,^[1] haben seither jedoch vor allem in der Architektur ([Computer-aided architectural design](#)) Anwendung gefunden. Shape grammars eignen sich dabei vor allem für kleine, klar definierte Probleme, wie etwa Struktur und Layout von Innenräumen oder Fassaden von Gebäuden. Shape grammars bestehen dabei sehr oft aus einer großen Anzahl an Regeln. So besteht die von William Mitchell^[2] präsentierte Shape grammar zur Generierung einer Villa im Stile des italienischen Architekten [Andrea Palladio](#) aus 69 Regeln, die in acht Durchführungsschritten angewendet werden.

Ähnlich zur Anwendung in der Architektur haben Shape grammars auch in der [Computergrafik](#) in den letzten Jahrzehnten an Bedeutung gewonnen. Shape grammars werden dabei vor allem bei der prozeduralen Modellierung von Gebäuden oder Städten eingesetzt (z. B.: für [Filme](#) oder Videospiele). Shape Grammars bilden dabei die Grundlage für zahlreiche entwickelter Systeme die anhand von Produktionsregeln eine Variation von unterschiedlichen 3D-Modellen generieren. Sowohl realistisch aussehende Straßenplänen, als auch Fassaden oder Innenräume von Gebäuden können dabei mit Hilfe von Shape Grammars prozedural erstellt werden.^{[3][4][5]} Zu weiteren Bereichen in denen Shape grammars angewendet wurden, zählen Industriedesign und Ingenieurwissenschaften.^[6]

12.8 – Bildtafeln

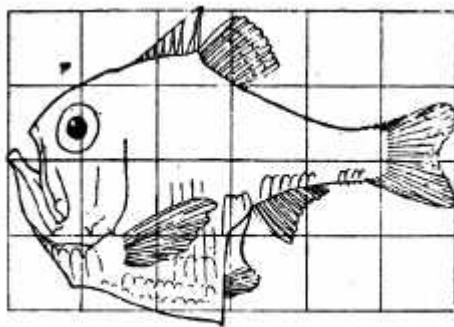


Fig. 517. *Argyropelecus Olfersi*.

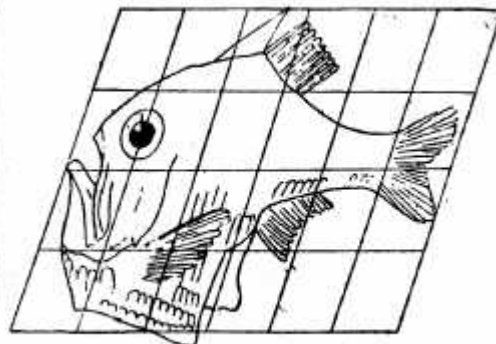


Fig. 518. *Sternoptyx diaphana*.

Abbildung 12.24 – Thompson illustrierte die Transformation von *Argyropelecus olfersi* in *Sternoptyx diaphana* durch die Anwendung einer 20° Scherungsabbildung (engl. shear mapping).

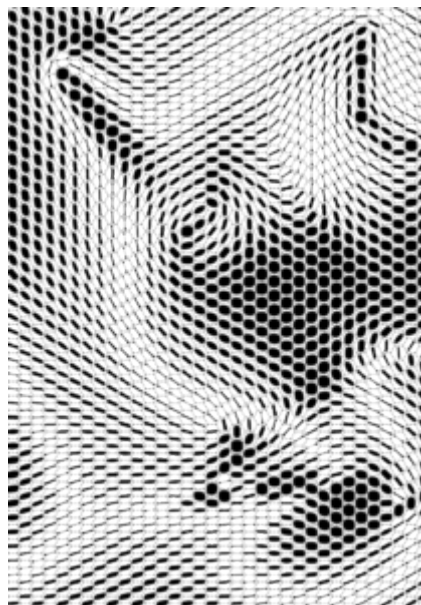
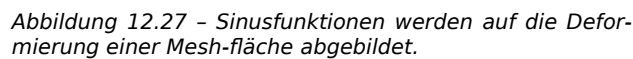
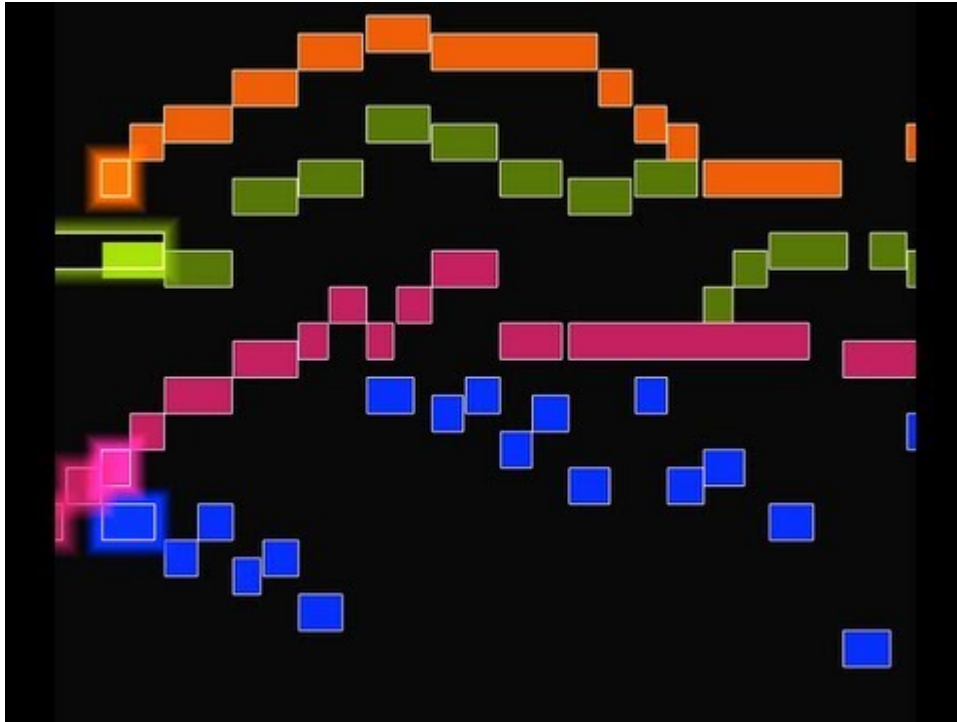


Abbildung 12.25 – Die Orientierung von Attraktoren auf verschiedene geometrische Transformationen abgebildet.

Abbildung 12.26 – ASCII Zeichen nach Helligkeitswerten abgebildet.





Ein YouTube Element wurde aus dieser Textversion entfernt. Sie können es hier online ansehen: <https://wp-prd.let.ethz.ch/computationaldesigniii/?p=647>

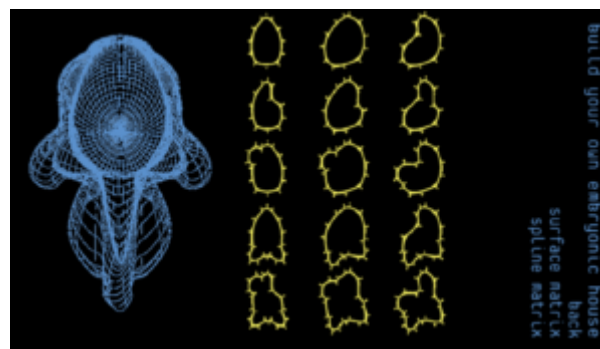


Abbildung 12.28 – Greg Lynn – Embryonic House

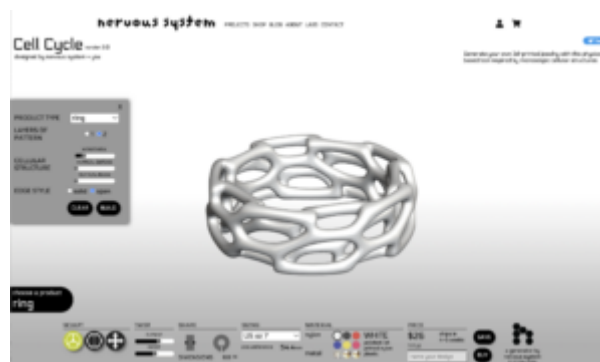


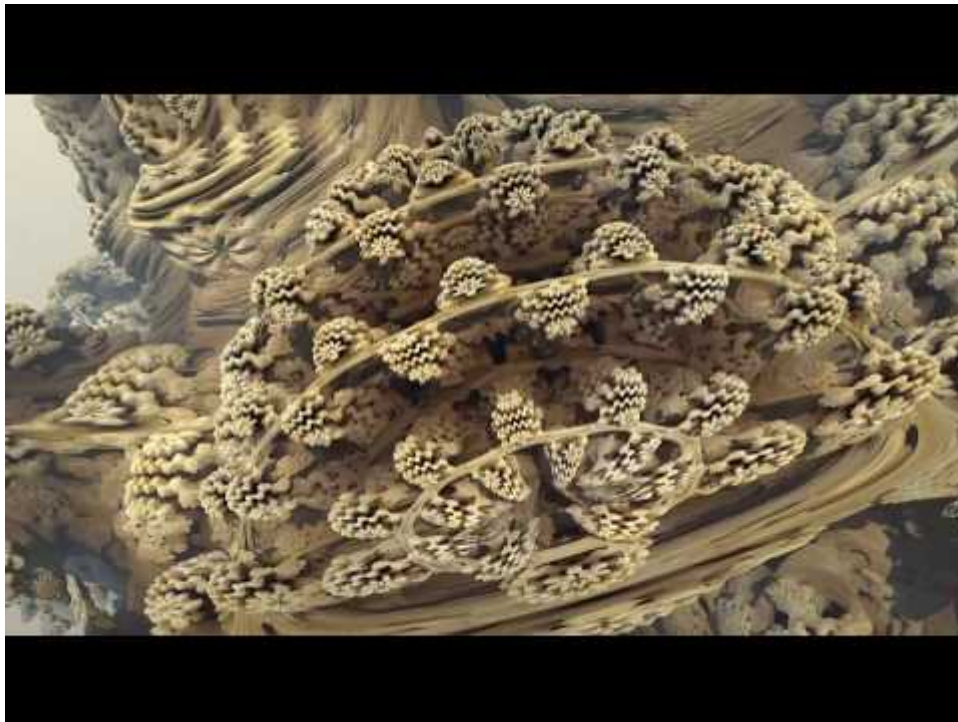
Abbildung 12.29 – Nervous Systems



Abbildung 12.30 – 3D-gedruckte parametrische Schuhsohle



Abbildung 12.31 – City Engine





Ein YouTube Element wurde aus dieser Textversion entfernt. Sie können es hier online ansehen: <https://wp-prd.let.ethz.ch/computationaldesigniii/?p=647>

City-Engine: Procedural City Generator basierend auf Prinzipien von Shape Grammar

Analyse und Simulation

13.1 – Geometrische Analysen

13.1.1 – Sichtbarkeit

Eine einzelne Isoviste ist das Raumvolumen, das von einem bestimmten Punkt im Raum aus sichtbar ist, zusammen mit einer Angabe der Position dieses Punktes. Isovisten sind von Natur aus dreidimensional, können aber auch in zwei Dimensionen untersucht werden: entweder im horizontalen Schnitt („Plan“) oder in anderen vertikalen Schnitten durch die dreidimensionale Isoviste. Jedem Punkt im physikalischen Raum ist eine Isoviste zugeordnet. Die Isoviste ist ein Ansatz, Raum aus der Sicht einer Person in einer Umgebung zu beschreiben. Es bezieht sich auf das gezeichnete Polygon, das einen Bereich abdeckt, der gesehen oder erreicht werden kann, wenn er von einer bestimmten Position aus in einer geraden Linie geht.

Beispiel

Die Grenzform eines Isovisten kann je nach Standort in einem Raum variieren oder nicht. Wenn der Raum konvex ist (z. B. ein Rechteck oder ein Kreis), dann ist die Grenzform aller Isovisten in diesem Raum gleich; ebenso deren Volumen (oder deren Fläche, wenn wir im Plan denken). Wenn der Raum jedoch nicht konvex wäre (z. B. ein L-förmiger oder unterteilter Raum), dann gäbe es viele Isovisten, deren Volumen (Fläche) kleiner wäre als das des gesamten Raums, und vielleicht einige, die das ganze Zimmer wären; und viele hätten unterschiedliche, vielleicht einzigartige Formen: groß und klein, schmal und breit, zentrisch und exzentrisch, ganz und nur teilweise.

Man kann sich die Isoviste auch als das von einer Punktlichtquelle beleuchtete Raumvolumen vorstellen. Es kann auch in der digitalen 3D-Umgebung als der Bereich betrachtet werden, der nicht im Schatten einer einzelnen Punktlichtquelle liegt.

Es wird im Bereich der Architektur zur Analyse von Gebäuden und Stadtgebieten verwendet (u.a. als eine von einer Reihe von Methoden, die in der Space Syntax verwendet werden).



Abbildung 13.1 – Die blaue Fläche ist der Isovist der vom Zentrum des Kreises gesehen werden kann.

13.1.2 – Sonne & Schatten

In der **Darstellenden Geometrie** verwendet man **Schattenkonstruktionen** um die Darstellung räumlicher Objekte attraktiver zu machen oder um Beleuchtungsprobleme schon in der Planungsphase diskutieren zu können. Wenn eine Parallelprojektion oder eine Zentralprojektion eines Objektes, z. B. eines Hauses, vorliegt, so ist es relativ einfach, **Schatten** auf eine horizontale Ebene (Standebene) sowohl von parallelem (Sonne) als auch zentralem (Lampe) Licht zu konstruieren. In allgemeineren Fällen (Schatten auf Ebenen, die von der Standebene verschieden sind) kann man den Schatten zunächst in Grund- und Aufriss, eventuell mit Hilfe der **Durchstoßpunktmethode**, konstruieren und dann in das Bild einzeichnen.

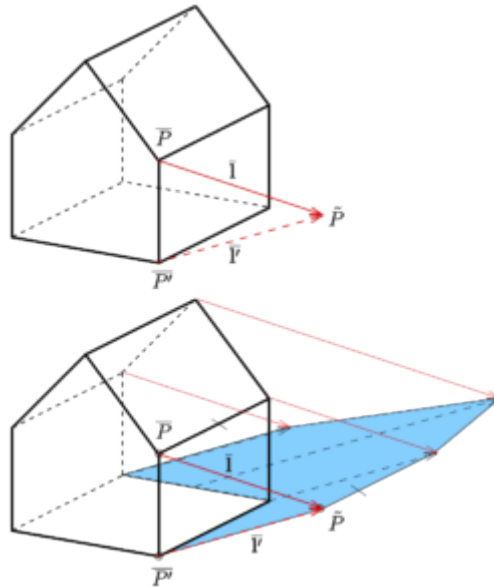


Abbildung 13.2 – Parallelprojektion: Schatten bei parallelem Licht

13.1.2.1 – Kanton Zurich: Schattenwurf von Hochhäusern

Hochhäuser dürfen ihre Nachbarschaft insbesondere durch Schattenwurf nicht wesentlich beeinträchtigen. So dürfen bewohnte Gebäude oder bebaute sowie überbaubare Flächen in Wohnzonen an den mittleren Wintertagen nicht länger als drei Stunden beschattet werden.

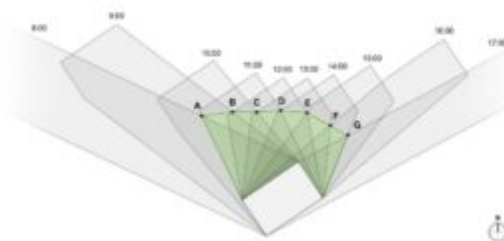


Abbildung 13.3 – Das gesuchte Schattendiagramm ergibt sich durch das Verbinden der Schnittpunkte der jeweils drei Stunden auseinanderliegenden Schatten.

13.2 – Netzwerk Analysen

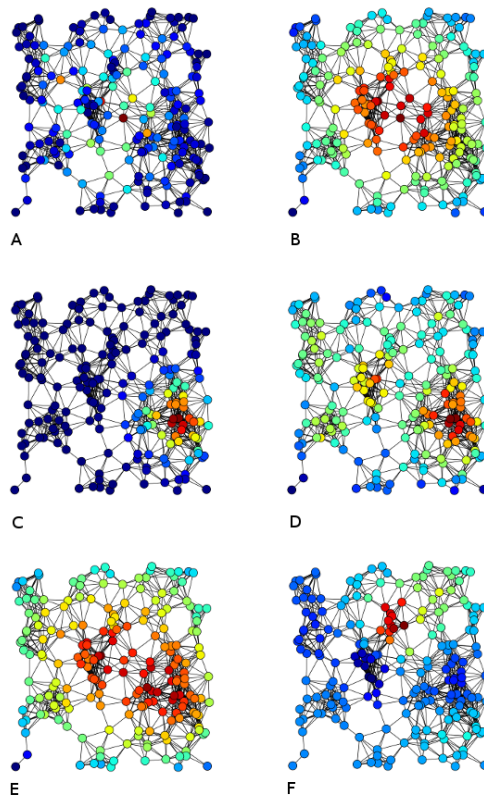


Abbildung 13.4 – Zentralitätsberechnung A) Betweenness, B) Closeness, C) Eigenvector, D) Degree, E) Harmonic und F) Katz.

Der Begriff **Space Syntax** umfasst eine Reihe von Theorien und Techniken zur Analyse räumlicher Konfigurationen. Es wurde von Bill Hillier, Julienne Hanson und Kollegen am The Bartlett, University College London in den späten 1970er bis frühen 1980er Jahren konzipiert, um Einblicke in die gegenseitig konstruktive Beziehung zwischen Gesellschaft und Raum zu gewinnen. Mit der Entwicklung der Space Syntax hat sich herausgestellt, dass bestimmte Maße mit dem menschlichen Raumverhalten korrelieren. Space Syntax wurde daher verwendet, um wahrscheinliche Auswirkungen des architektonischen und städtischen Raums auf die Benutzer vorherzusagen.

13.3 – Simulation

Die Simulation oder Simulierung bezeichnet die Nachbildung von realen Szenarien oder der Analyse von **Systemen**, deren Verhalten für die theoretische, formelmäßige Behandlung zu **komplex** sind. Bei der Simulation werden Experimente oder Trainings an einem **Modell** durchgeführt, um Erkenntnisse über das reale System zu gewinnen. Dabei wird das Modell unterschiedlichen Einflussgrößen ausgesetzt. Letzteres stellt eine **Abstraktion** des zu simulierenden Systems dar (Struktur, Funktion, Verhalten).

13.4 – Physics Engine

Eine **Physik-Engine** ist eine **Funktionseinheit der Informationstechnologie**, die zur **Simulation** physikalischer Prozesse sowie der Berechnung objektimmanenter Eigenschaften (z. B. **Impuls**) dient. Ziele sind eine Vereinfachung der Programmierung und die Vermittlung von realistischen Bewegungsabläufen in einer 3D-Ansicht, beispielsweise soll eine realistische „Spielphysik“ erzielt werden. Hauptanwendungsgebiete sind moderne **Computerspiele** und Simulationssoftware, sowie in Filmstudios zur Erzeugung von **Spezialeffekten**.

Dabei geht es nicht immer unbedingt um physikalische Exaktheit. Dies liegt oft an der immensen Komplexität der auf physikalischen Prozessen basierenden Berechnungen. Da in solchen Fällen meist ein Anspruch auf Echtzeitfähigkeit besteht, geht Effizienz vor Exaktheit.

Physikseingines lassen sich je nach Methode in folgende Kategorien unterteilen:

- Physik **starrer Körper** (rigid body mechanics)
- nicht elastische Physik mit Deformationen (soft body dynamics)
- Masse-Feder Modelle (spring dynamics) u. a. als Grundlage der Simulation von Seilen und Stoff (rope bzw. cloth)
- Partikelsysteme u. a. als Grundlage der Simulation von Flüssigkeiten (fluid dynamics) und Feuer

13.5 – Partikel

Ein **Partikelsystem** ist eine Funktion im Bereich der **Computeranimation**, mit der sich eine große Anzahl von Objekten animieren lässt. Partikelsysteme werden beispielsweise eingesetzt, um Feuer-, Rauch- oder Explosionseffekte zu simulieren.

Bei einem Partikelsystem werden von einem so genannten Emitter Partikel ausgestoßen, deren Bewegung über zahlreiche **Parameter** beeinflusst werden kann^[1]. Diese sind unter anderem:

- Ausstoßgeschwindigkeit
- Lebensdauer
- Dämpfung (das Partikel wird mit der Zeit langsamer)
- Anzahl der Partikel im Gesamtsystem
- Aufteilung von Partikeln in eine bestimmte Zahl weiterer Partikel nach einer bestimmten Zeitspanne (Multiplikation)
- eine Zufälligkeit des Verhaltens
- Krafteinflüsse auf die Partikel

Die Partikel sind zunächst nur logische Elemente, denen grafische Eigenschaften zugewiesen werden müssen, damit sie sichtbar werden. Zum Beispiel kann jedes Partikel durch ein geometrisches Objekt ersetzt werden, wodurch sich **Schwärme**, **Asteroidenfelder** und Ähnliches darstellen lassen (Mitte im Bild). Mit entsprechenden **Materialzuweisungen** können Partikelsysteme auch eingesetzt werden, um Rauch, Nebel oder Feuer darzustellen (links im Bild).

Weitergehende Partikelsysteme können auch auf andere Objekte reagieren, von denen sie je nach Einstellung angezogen oder abgestoßen werden oder an denen sie abprallen (physikalische/ereignisbasierende Verhaltensweisen)^[2]. Durch die Darstellung der Bewegungspfade der einzelnen Partikel werden Haare, Plüsch, Grasflächen und ähnliche Strukturen realisiert^[2].

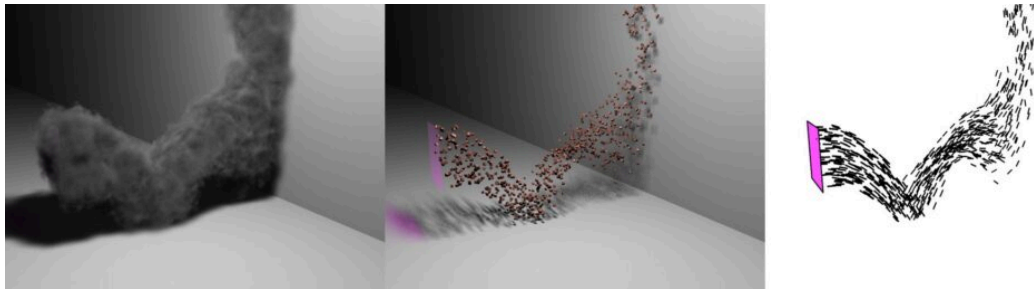


Abbildung 13.5 – Stellt Verschiedene Anwendungen von Partikelsystemen bei 3D-Programmen dar.

13.6 – Agenten

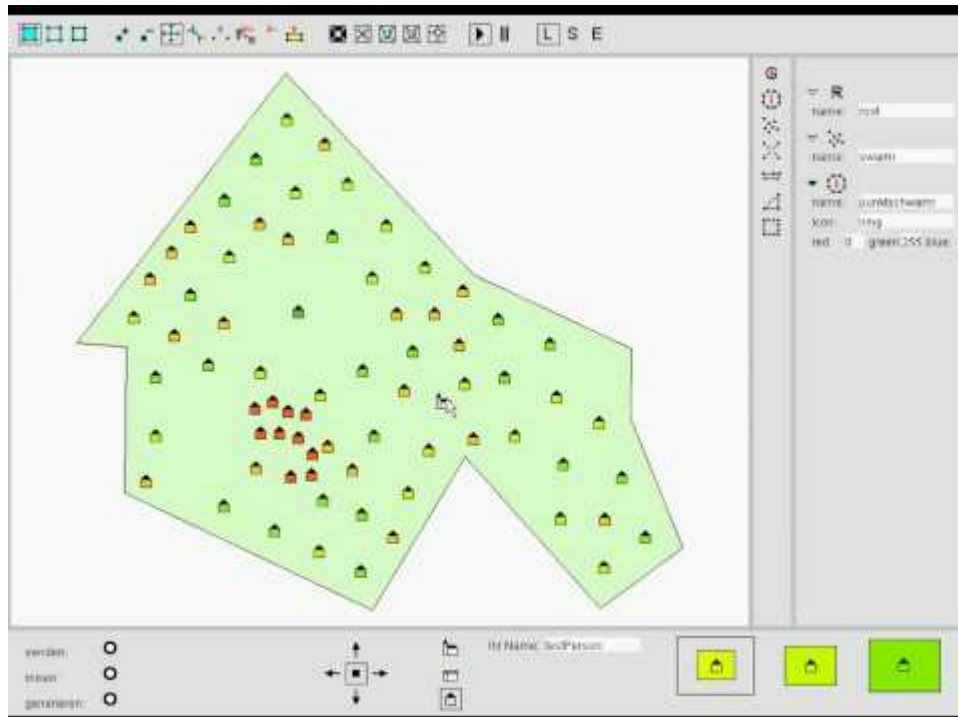
Bei einem **Multiagentensystem** oder **MAS** handelt es sich um ein **System** aus mehreren gleichartigen oder unterschiedlich spezialisierten handelnden Einheiten, **Software-Agenten**, die kollektiv ein Problem lösen.

Multiagentensysteme existieren sowohl in der Biologie (natürliche Multiagentensysteme) als auch in der Technik. Eine Beispielfamilie biologischer Multiagentensysteme stellen Ameisenstaaten dar. Einige der in Ameisenstaaten ablaufenden Algorithmen (**Ameisenalgorithmen**) stellen heuristische Lösungsverfahren für komplexe Optimierungsaufgaben dar und sind neben ihrem grundsätzlichen Interesse innerhalb der **Theoretischen Biologie** auch ein Vorbild bei der Optimierung technischer Prozesse. Man spricht auch von **verteilter**, im Bereich der Technik von **künstlicher Intelligenz** oder DAI (*distributed artificial intelligence*).

Außerhalb Europas, insbesondere in den USA, hat sich für MAS die Bezeichnung Agent-based Modeling bzw. Simulation (ABM) durchgesetzt.

13.7 – Emergenz

Emergenz(lateinisch emergere „Auftauchen“, „Herauskommen“, „Emporsteigen“) bezeichnet die Möglichkeit der Herausbildung von neuen Eigenschaften oder Strukturen eines Systems infolge des Zusammenspiels seiner Elemente. Dabei lassen sich die emergenten Eigenschaften des Systems nicht – oder jedenfalls nicht offensichtlich – auf Eigenschaften der Elemente zurückführen, die diese isoliert aufweisen. Das Phänomen der emergenten Selbstorganisation wird als *Autopoiesis* bezeichnet.



Ein YouTube Element wurde aus dieser Textversion entfernt. Sie können es hier online ansehen: <https://wp-prd.let.ethz.ch/computationaldesigniii/?p=37>

Suche und Optimierung

...that we build machines equipped with at least those devices that humans employ to design. Let us build machines that can learn, can grope, and can fumble, machines that will be architectural partners, architecture machines. Negroponte, Nicholas. *Architecture Machine - Toward a More Human Environment*. Cambridge, Mass.: MIT Press 1970

14.1 – Entwerfen als Variantenerzeugung und Einschränkung

Engineering, medicine, business, architecture, and painting are concerned not with the necessary but with the contingent — not with how things are but with how they might be — in short, with design.

Herbert Simon, *The science of the artificial*

Ich entwerfe nicht, ich suche

Frei Otto

14.2 – Probleme

Gemeinsam mit Melvin Webber prägte Horst Rittel ab Mitte der 1960er Jahre den Begriff „Wicked Problems“. Der im deutschen bisweilen als „böartige Probleme“ übersetzte Begriff zielt auf die große Bandbreite undefinierter Probleme, die sich beispielsweise in Planungs- oder Gestaltungsprozessen ergeben. Horst Rittel, der als einer der ersten eine Theorie der *wicked problems* formuliert hat, nennt zehn Merkmale^[2] dieser komplexen Sachverhalte:

1. *Wicked problems* haben keine definitive Formulierung. Das Problem der Armut in Texas ist dem Problem der Armut in Nairobi zwar ähnlich, unterscheidet sich aber dennoch von dem in Nairobi, so dass keine praktischen Merkmale „Armut“ beschreiben.
2. Es ist schwer, vielleicht sogar unmöglich, den Erfolg von *wicked problems* zu messen oder zu beanspruchen, weil sie ineinander fließen, einander bedingen, im Gegensatz zu den Grenzen traditioneller Gestaltungsprobleme, die vergleichsweise klar artikuliert oder definiert werden können.
3. Lösungen für *wicked problems* können nur gut oder schlecht, nicht wahr oder falsch sein. Es gibt keinen idealisierten Endzustand, den es zu erreichen gilt, und deshalb sollten Ansätze für *wicked problems* tragfähige Wege sein, um eine Situation zu verbessern, anstatt sie zu lösen.
4. Es gibt keine Schablone, an die man sich halten kann, wenn man ein *wicked problem* angeht, wenngleich die Geschichte als Leitfaden dienen kann. Teams, die sich mit *wicked problems* auseinandersetzen, müssen buchstäblich Dinge erfinden, während sie voranschreiten.
5. Es gibt immer mehr als eine Erklärung für ein *wicked problem*, wobei die Angemessenheit der Erklärung stark von der individuellen Perspektive des/r Gestaltenden abhängt.
6. Jedes *wicked problem* ist ein Symptom für ein anderes Problem. Die vernetzte Qualität sozioökonomischer politischer Systeme veranschaulicht, wie z.B. eine Veränderung in der Bildung zu einem neuen Ernährungsverhalten führt.
7. Keine Abmilderungsstrategie für ein *wicked problem* verfügt über einen definitiven wissenschaftlichen Test, weil wicked Problems mit Menschen zu tun haben und die

- (Natur-) Wissenschaft insbesondere darauf ausgerichtet ist, natürliche Phänomene zu verstehen.
8. Das Anbieten einer "Lösung" für ein *wicked problem* ist häufig ein "One-Shot", weil eine signifikante Intervention den Gestaltungsraum so weit verändert, dass die Fähigkeit zum Ausprobieren minimiert wird.
 9. Jedes *wicked problem* ist einzigartig.
 10. Konstrukteure, die versuchen, ein *wicked problem* zu lösen, müssen die volle Verantwortung für ihre Handlungen übernehmen.

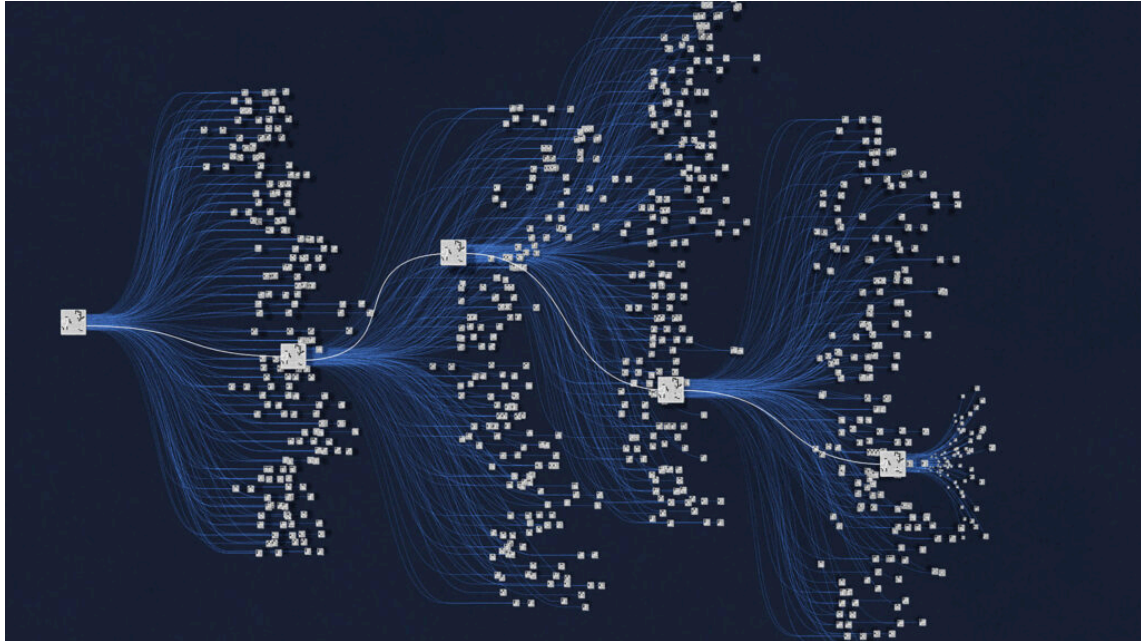


Abbildung 14.1 – Ausschnitt aus dem Entscheidungsbaum des Spiels GO

14.3 – Heuristik

Heuristik (von [altgriechisch](#) εὐρίσκω *heurísko* (ich finde); von εὐρίσκειν *heurískein* (auffinden entdecken)) bezeichnet die Kunst, mit begrenztem Wissen ([unvollständigen Informationen](#)) und wenig Zeit dennoch zu wahrscheinlichen Aussagen oder praktikablen Lösungen zu kommen.[1] Es bezeichnet ein analytisches Vorgehen, bei dem mit begrenztem Wissen über ein System mit Hilfe [mutmaßender Schlussfolgerungen](#) Aussagen über das System getroffen werden. Die damit gefolgerten Aussagen weichen oftmals von der optimalen Lösung ab. Durch Vergleich mit einer optimalen Lösung kann die Güte der Heuristik bestimmt werden.

Bekannte Heuristiken sind zum Beispiel [Versuch und Irrtum](#) (trial and error), statistische Auswertung von [Zufallsstichproben](#) und das [Ausschlussverfahren](#). Heuristische Verfahren basieren auf Erfahrungen; sie können auch auf „falschen“ Erfahrungen (z. B. [verzerrte Wahrnehmung](#), [Scheinkorrelation](#)) basieren.[2]

14.4 – Evolutionäre Algorithmen

Evolutionäre Algorithmen (EA) sind eine Klasse von [stochastischen, metaheuristischen Optimierungsverfahren](#), deren Funktionsweise von der [Evolution](#) natürlicher Lebewesen inspiriert ist.

In Anlehnung an die [Natur](#) werden Lösungskandidaten für ein bestimmtes Problem künstlich evolviert, EA sind also [naturanalogue Optimierungsverfahren](#). Die Zuordnung zu den stochastischen und metaheuristischen Algorithmen bedeutet vor allem, dass EA meist nicht die beste Lösung für ein Problem finden, aber bei Erfolg eine hinreichend gute, was in der Praxis vor allem bei [NP-vollständigen](#) Problemen bereits wünschenswert ist. Die Verfahren verschiedener EA unterscheiden sich untereinander in erster Linie durch die genutzten [Selektions-, Rekombinations-](#) und [Mutationsoperatoren](#), das [Genotyp-Phänotyp](#)-Mapping sowie die [Problemrepräsentation](#).

Die ersten praktischen **Implementierungen** evolutionärer Algorithmen wurden Ende der 1950er Jahre veröffentlicht,^[2] allerdings äußerten sich bereits in den vorhergehenden Jahrzehnten Wissenschaftler zum Potenzial der Evolution für **maschinelles Lernen**.^[3]

Die Anwendungen von EA gehen über Optimierung und Suche hinaus und finden sich auch in **Kunst**, **Modellierung** und **Simulation**, insbesondere auch bei der Untersuchung **evolutionsbiologischer Fragestellungen**.

Das grobe Verfahren evolutionärer Algorithmen besteht meist aus einer Initialisierung und einer **Generationsschleife**, die solange durchlaufen wird, bis ein **Abbruchkriterium** erfüllt ist:^[9]

1. **Initialisierung**: Die erste **Generation** von Lösungskandidaten wird (meist zufällig) erzeugt.
2. **Evaluation**: Jedem Lösungskandidaten der Generation wird entsprechend seiner Güte ein Wert der Fitnessfunktion zugewiesen.
3. Durchlaufe die folgenden Schritte, bis ein Abbruchkriterium erfüllt ist:
 - a. **Selektion**: Auswahl von Individuen meist basierend auf ihrer Fitness, die die Eltern für die Rekombination bilden
 - b. **Rekombination**: Erzeugung von Nachkommen durch zufällige Kombination der Genome der Eltern
 - c. **Mutation**: Zufällige Veränderung aller oder eines Teils der Nachkommen
 - d. **Evaluation**: Jedem Nachkommen wird entsprechend seiner Güte ein Wert der Fitnessfunktion zugewiesen.
 - e. **Selektion**: Bestimmung einer neuen Generation aus der alten und/oder den in dieser Generation gebildeten Nachkommen

14.5 – Fitnesslandschaft

Eine **Fitnessfunktion** ist die **Zielfunktion** eines **evolutionären (Optimierungs-)Algorithmus** (EA). Gelegentlich wird eine Fitnessfunktion auch als Teil einer Zielfunktion beschrieben^[1] oder umgekehrt. Wie auch evolutionäre Algorithmen haben Fitnessfunktionen ein biologisches Vorbild, die **biologische Fitness**, die den Grad der Anpassung eines **Organismus** an seine Umgebung angibt und einen wesentlichen Faktor für seine Reproduktionswahrscheinlichkeit darstellt. Bei evolutionären Algorithmen beschreibt die Fitness eines Lösungskandidaten, wie gut er das zugrunde liegende Optimierungsproblem löst. Die Fitnessfunktion berechnet aus den Eigenschaften eines Lösungsversuchs, wie gut sich dieses „Individuum“ bzgl. des gestellten Problems als Lösung eignet.

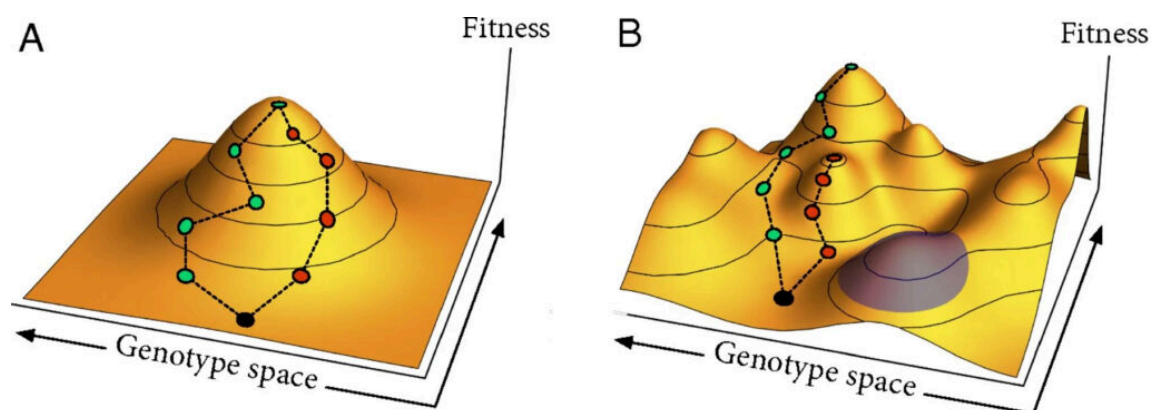


Abbildung 14.2 – Fitnesslandschaft

Eine Fitnessfunktion muss nicht zwangsläufig einen absoluten Wert berechnen können, da es oft reicht, Kandidaten zu vergleichen, um den besseren auszuwählen. Eine relative Angabe der Fitness (Kandidat a ist besser als b) genügt in manchen Fällen^[2], wie z. B. bei der **Pareto-Optimierung**. Ein **Pareto-Optimum** (auch **Pareto-effizienter Zustand**) ist ein (bestmöglicher) Zustand, in dem es nicht möglich ist, eine (Ziel-)Eigenschaft zu verbessern, ohne zugleich eine andere verschlechtern zu müssen.

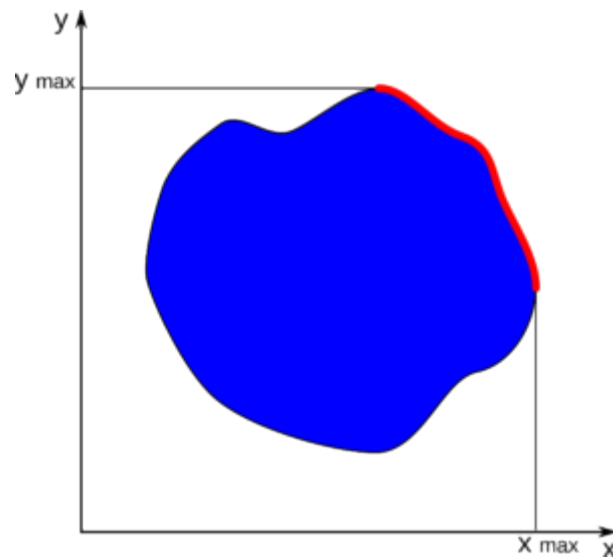


Abbildung 14.3 – y-Achse: Festigkeit x-Achse: „Leichtigkeit“ (=Kehrwert der Masse) Ein Bauteil soll sowohl belastbar als auch leicht werden. Es sei also gekennzeichnet durch die zwei Eigenschaften Festigkeit und Masse. Je höher die Festigkeit und je geringer die Masse, desto besser sei das Bauteil. Trägt man die Wertepaare für viele verschiedene Bauteile in ein Diagramm ein, das Festigkeit und Leichtigkeit (Kehrwert der Masse) gegenüberstellt, so erhält man die blau markierte Menge.

Glossary

3D Druck

Der 3D-Druck (auch 3-D-Druck), auch bekannt unter den Bezeichnungen additive Fertigung (englisch Additive Manufacturing, AM), generative Fertigung oder Rapid-Technologien, ist eine umfassende Bezeichnung für alle Fertigungsverfahren, bei denen Material Schicht für Schicht aufgetragen und so dreidimensionale Gegenstände (Werkstücke) erzeugt werden.

Agenten

Bei einem Multiagentensystem oder MAS handelt es sich um ein [System](#) aus mehreren gleichartigen oder unterschiedlich spezialisierten handelnden Einheiten, [Software-Agenten](#), die kollektiv ein Problem lösen.

Algorithmus

Ein Algorithmus ist eine eindeutige Handlungsvorschrift zur Lösung eines Problems oder einer Klasse von Problemen. Algorithmen bestehen aus endlich vielen, wohldefinierten Einzelschritten.

API

Eine Programmierschnittstelle, häufig nur kurz API genannt, ist ein Programmteil, der von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung gestellt wird.

Auflösung

Die Bildauflösung, kurz auch Auflösung genannt, ist ein umgangssprachliches Maß für die Bildgröße einer Rastergrafik. Sie wird durch die Gesamtzahl der Bildpunkte oder durch die Anzahl der Spalten (Breite) und Zeilen (Höhe) einer Rastergrafik angegeben.

Bedingung

Eine bedingte Anweisung ist eine Kontrollstruktur in der Programmierung. Ein Programmabschnitt wird dabei nur unter einer bestimmten Bedingung ausgeführt. Eine Verzweigung legt fest, welcher von zwei oder mehreren Programmabschnitten, abhängig von einer oder mehreren Bedingungen, ausgeführt wird.

BIM

Der Begriff Building Information Modeling (kurz: BIM; deutsch: Bauwerksdatenmodellierung) beschreibt eine Arbeitsmethode für die vernetzte Planung, den Bau und die Bewirtschaftung von [Gebäuden](#) und anderen Bauwerken mithilfe von [Software](#). Dabei werden alle relevanten Bauwerksdaten digital modelliert, kombiniert und erfasst. Das Bauwerk ist als virtuelles Modell auch geometrisch visualisiert ([Computermode](#)ll). Building Information Modeling findet Anwendung sowohl im [Bauwesen](#) zur [Bauplanung](#) und [Bauausführung](#) ([Architektur](#), [Ingenieurwesen](#), [Haustechnik](#), [Tiefbau](#), [Städtebau](#), [Eisenbahnbau](#), [Straßenbau](#), [Wasserbau](#), [Geotechnik](#)) als auch im [Facilitymanagement](#)[1].

Binärcode

Ein Binärcode ist ein Code, in dem Informationen durch Sequenzen von zwei verschiedenen Symbolen (zum Beispiel 1/0 oder wahr/falsch) dargestellt werden.

BRep

Ein Flächenverband besteht aus zwei oder mehr (möglicherweise getrimmten) miteinander verbundenen NURBS-Flächen. Jede Fläche hat ihre eigene Struktur, Parametrisierung und Isokurven-Richtungen, die nicht übereinstimmen müssen. Flächenverbände werden unter Verwendung des Begrenzungsflächenmodells (BRep engl. Abk. für Boundary Representation) dargestellt.

CAD

CAD (von engl. computer-aided design, zu Deutsch rechnerunterstütztes Konstruieren) bezeichnet die Unterstützung von konstruktiven Aufgaben mittels EDV zur Herstellung eines Produkts (Beispielsweise Auto, Flugzeug, Bauwerk, Kleidung). In einem engeren Sinn versteht man unter CAD das rechnerunterstützte Erzeugen und Ändern des geometrischen Modells. In einem weiteren Sinn versteht man darunter sämtliche rechnerunterstützten Tätigkeiten in einem Konstruktionsprozess, einschließlich der geometrischen Modellierung, des Berechnens, des Simulierens und sonstiger Informationsgewinnung und Informationsbereitstellung, von der Konzeptentwicklung bis zur Übergabe an die Herstellung bzw. Fertigung (Arbeitsvorbereitung).

Cloud Computing

Cloud Computing (deutsch Rechnerwolke oder Datenwolke) beschreibt ein Modell, das bei Bedarf – meist über das Internet und geräteunabhängig – zeitnah und mit wenig Aufwand geteilte Computerressourcen als Dienstleistung, etwa in Form von Servern, Datenspeicher oder Applikationen, bereitstellt und nach Nutzung abrechnet. Angebot und Nutzung dieser Computerressourcen ist definiert und erfolgt in der Regel über eine Programmierschnittstelle (API) bzw. für Anwender über eine Website oder App.

CNC

CNC-Maschinen (Computerized Numerical Control) sind Werkzeugmaschinen, die durch den Einsatz von Steuerungstechnik in der Lage sind, Werkstücke mit hoher Präzision auch für komplexe Formen automatisch herzustellen.

Digitale Signale

Ein digitales Signal (digitus: Finger, lat.) ist eine physikalische Größe, die nur bestimmte diskrete Werte annehmen kann. Die Werte entsprechen der Anzahl der vereinbarten Zustände. Werden zwei Zustände vereinbart, dann handelt es sich um binäre (digitale) Signale.

Dreipunktbeleuchtung

Dreipunktbeleuchtung ist eine Standardmethode, die in visuellen Medien wie Theater, Video, Film, Standfotografie, computergenerierten Bildern und 3D-Computergrafik verwendet wird.

Emergenz

Emergenz (lateinisch emergere „Auftauchen“, „Herauskommen“, „Emporsteigen“) bezeichnet die Möglichkeit der Herausbildung von neuen Eigenschaften oder Strukturen eines Systems infolge des Zusammenspiels seiner Elemente.

Evolutionäre Algorithmen

Evolutionäre Algorithmen (EA) sind eine Klasse von stochastischen, metaheuristischen Optimierungsverfahren, deren Funktionsweise von der Evolution natürlicher Lebewesen inspiriert ist.

GIS

Geoinformationssysteme, Geographische Informationssysteme (GIS) oder Räumliche Informationssysteme (RIS) sind Informationssysteme zur Erfassung, Bearbeitung, Organisation, Analyse und Präsentation räumlicher Daten. Geoinformationssysteme umfassen die dazu benötigte Hardware, Software, Daten und Anwendungen.

Globale Beleuchtung

Globale Beleuchtung (englisch Global Illumination, abgekürzt GI) bezeichnet in der 3D-Computergrafik allgemein die Simulation der Licht-Wechselwirkungen von Objekten

Graph

Ein Graph ist in der Graphentheorie eine abstrakte Struktur, die eine Menge von Objekten zusammen mit den zwischen diesen Objekten bestehenden Verbindungen repräsentiert. Die mathematischen Abstraktionen der Objekte werden dabei Knoten des Graphen genannt. Die paarweisen Verbindungen zwischen Knoten heißen Kanten. Die Kanten können gerichtet oder ungerichtet sein. Häufig werden Graphen anschaulich gezeichnet, indem die Knoten durch Punkte und die Kanten durch Linien dargestellt werden.[2]

HCI

Mensch-Computer-Interaktion (häufig als HCI abgekürzt, englisch Human-computer interaction) erforscht das Design und die Verwendung von Computer-Technologie an der Schnittstelle zwischen Menschen (Anwendern) und Computern

Heuristik

Heuristik (von [altgriechisch](#) εὕρισκω heurísko (ich finde); von εὕρίσκειν heurískein (auffinden entdecken)) bezeichnet die Kunst, mit begrenztem Wissen ([unvollständigen Informationen](#)) und wenig Zeit dennoch zu wahrscheinlichen Aussagen oder praktikablen Lösungen zu kommen.

Implizite Flächen

Implizite Flächen sind Flächen, die im Raum benachbarte Punkte gleicher Merkmale oder Werte einer bestimmten Größe wie zum Beispiel Temperatur oder Dichte miteinander verbinden. Sie sind das dreidimensionale Gegenstück zu Isolinien, die Punkte auf einer Fläche verbinden.

Industry Foundation Classes

Die Industry Foundation Classes sind ein offener Standard im Bauwesen zur digitalen Beschreibung von Gebäudemodellen.

Internet of Things

Das Internet der Dinge (IdD) (auch: „Allesnetz“; englisch Internet of Things, Kurzform: IoT) ist ein Sammelbegriff für Technologien einer globalen Infrastruktur der Informationsgesellschaften, die es ermöglicht, physische und virtuelle Objekte miteinander zu vernetzen und sie durch Informations- und Kommunikationstechniken zusammenarbeiten zu lassen.

Mesh

Untereinander mit Kanten verbundene Punkte bilden in der Computergrafik ein Polygonnetz, das die Gestalt eines Polyeders definiert. Dreiecksnetze und Vierecksnetze sind hier am geläufigsten.

In der 3D-Computergrafik und der Volumenmodellierung ist ein Polygonnetz eine Sammlung von Scheitelpunkten, Kanten und Flächen, die die Form eines polyedrischen Objekts definieren. Die Flächen bestehen normalerweise aus Dreiecken (Triangle Mesh), Vierecken (Quads) oder anderen einfachen konvexen Polygonen (n-Gons), da dies das Rendern vereinfacht, können aber auch allgemeiner aus konkaven Polygonen oder sogar Polygonen mit Löchern zusammengesetzt sein.

Metaballs

Die Form von Metabällen, auch bekannt als „Blobs“ oder „weiche Objekte“, wird durch Eingabeformen gesteuert, bei denen es sich in der Regel um einfache Objekte wie Punkte, Linien und einfache Oberflächen handelt. Unter Verwendung des Abstands zu den Eingabeformen wird jede Eingabeform einer lokalen Feldfunktion zugeordnet. Diese lokalen Feldfunktionen werden summiert, was eine Funktion $f(x,y,z)$ ergibt.

Mixed reality

Unter Mixed reality, Vermischte Realität oder Gemischte Realität werden Umgebungen oder Systeme zusammengefasst, die die natürliche Wahrnehmung eines Nutzers mit einer künstlichen (computererzeugten) Wahrnehmung vermischen. Neben der hauptsächlich computererzeugten virtuellen Realität sind dies insbesondere Systeme der erweiterten Realität und der erweiterten Virtualität.

Model

Ein Modell ist ein vereinfachtes Abbild der Wirklichkeit. Das Abbild kann die Form konkreter Gegenstände haben (Modell-Eisenbahn, Computersimulation u. ä.) oder rein abstrakt dargestellt sein (Theorien; Gleichungen).

NURBS

Non-uniform rational **B-Splines** (deutsch: nicht-uniforme rationale B-Splines, kurz NURBS) sind **mathematisch** definierte **Kurven** oder **Flächen**, die im **Computergrafik**-Bereich, beispielsweise im **CGI** oder **CAD**, zur Modellierung beliebiger Formen verwendet werden. **NURBS** sind mathematische Darstellungen, die beliebige Formen von einfachen 2D-Linien, -Kreise, -Bogen oder -Kurven bis hin zu hoch komplexen organischen 3D-Freifformflächen und -Volumenkörpern darstellen können. Aufgrund ihrer Flexibilität und Genauigkeit können NURBS-Modelle in allen Prozessen von Illustration und Animation bis hin zur Fertigung verwendet werden.

Ontologie

Die Ontologie (im 16. Jahrhundert als griechisch ὄντολογία ontología gebildet aus **altgriechisch** ὄν ὄν ‚seiend‘ bzw. **altgriechisch** τὸ ὄν ‚das Sein‘ und λόγος lógos ‚Lehre‘, also ‚Lehre vom Seienden‘ bzw. ‚Lehre des Seins‘) ist eine **Disziplin** der (theoretischen) **Philosophie**, die sich mit der Einteilung des Seienden und den Grundstrukturen der **Wirklichkeit** befasst, z. B. mit Begriffen wie **Existenz**, **Sein**, **Werden** und **Realität**.

Parameter

Ein Parameter oder auch eine Formvariable ist eine an sich unabhängige Variable, die aber zumindest in einer gegebenen Situation eher als eine festgehaltene Größe aufgefasst wird.

Partikel

Ein Partikelsystem ist eine Funktion im Bereich der **Computeranimation**, mit der sich eine große Anzahl von Objekten animieren lässt. Partikelsysteme werden beispielsweise eingesetzt, um Feuer-, Rauch- oder Explosionseffekte zu simulieren.

Physics Engine

Eine Physik-Engine ist eine **Funktionseinheit der Informationstechnologie**, die zur **Simulation** physikalischer Prozesse sowie der Berechnung objektimmanenter Eigenschaften (z. B. **Impuls**) dient. Ziele sind eine Vereinfachung der Programmierung und die Vermittlung von realistischen Bewegungsabläufen in einer 3D-Ansicht, beispielsweise soll eine realistische „Spielphysik“ erzielt werden. Hauptanwendungsgebiete sind moderne **Computerspiele** und Simulationssoftware, sowie in Filmstudios zur Erzeugung von **spezial Effekten**.

Programmiersprache

Eine Programmiersprache ist eine formale Sprache zur Formulierung von Datenstrukturen und Algorithmen, d. h. von Rechenvorschriften, die von einem Computer ausgeführt werden können.

Punktwolken

Eine Punktwolke oder ein Punkthaufen (englisch point cloud) ist eine Menge von Punkten eines Vektorraums, die eine unorganisierte räumliche Struktur („Wolke“) aufweist.[1] Eine Punktwolke ist durch die enthaltenen Punkte beschrieben, die jeweils durch ihre Raumkoordinaten erfasst sind. Punktwolken mit Georeferenzierung enthalten Punkte in einem erdbezogenen Koordinatensystem. Zu den Punkten können zusätzlich Attribute, wie z. B. geometrische Normale, Farbwerte, Aufnahmezeitpunkt oder Messgenauigkeit, erfasst sein.

Rapid Prototyping

Rapid Prototyping (übersetzt schneller Modellbau) ist der Überbegriff über verschiedene Verfahren zur schnellen Herstellung von Musterbauteilen ausgehend von den Konstruktionsdaten.[2]

Rastergrafik

Eine Rastergrafik, auch Pixelgrafik (englisch raster graphics image, digital image, bitmap oder pixmap), ist eine Form der Beschreibung eines Bildes in Form von computerlesbaren Daten. Rastergrafiken bestehen aus einer rasterförmigen Anordnung sogenannter Pixel, denen jeweils eine Farbe zugeordnet ist

Semantik

Semantik (von **altgriechisch** σημαίνειν sēmaínein, deutsch ‚bezeichnen, ein Zeichen geben‘), auch Bedeutungslehre, nennt man die Theorie oder Wissenschaft von der Bedeutung der Zeichen. Zeichen können hierbei beliebige Symbole sein, insbesondere aber auch Sätze, Satzteile,

Wörter oder Wortteile. Soweit sich die Semantik mit Zeichen aller Art befasst, ist sie ein Teilbereich der Semiotik. Sofern sie sich allein mit sprachlichen Zeichen befasst, ist sie eine Teildisziplin der Linguistik. Allgemeinsprachlich wird unter Semantik auch einfach die Bedeutung eines bestimmten Wortes, Satzes oder Textes verstanden.

Signed Distance Functions

Signed Distance Function oder kurz SDFs geben, wenn sie die Koordinaten eines Punktes im Raum übergeben, den kürzesten Abstand zwischen diesem Punkt und einer Oberfläche zurück. Das Vorzeichen des Rückgabewerts gibt an, ob sich der Punkt innerhalb oder außerhalb dieser Oberfläche befindet (daher vorzeichenbehaftete Abstandsfunktion).

Simulation

Die Simulation oder Simulierung bezeichnet die Nachbildung von realen Szenarien oder der Analyse von [Systemen](#), deren Verhalten für die theoretische, formelmäßige Behandlung zu [komplex](#) sind. Bei der Simulation werden Experimente oder Trainings an einem [Modell](#) durchgeführt, um Erkenntnisse über das reale System zu gewinnen. Dabei wird das Modell unterschiedlichen Einflussgrößen ausgesetzt. Letzteres stellt eine [Abstraktion](#) des zu simulierenden Systems dar (Struktur, Funktion, Verhalten).

Topologie

Die Topologie ([griechisch](#) τόπος *tópos*, deutsch ‚Ort, Platz, Stelle‘ und [-logie](#)) ist die Lehre von der Lage und Anordnung geometrischer Gebilde im Raum und ein fundamentales [Teilgebiet der Mathematik](#). Sie beschäftigt sich mit den Eigenschaften mathematischer Strukturen, die unter stetigen Verformungen erhalten bleiben, wobei der Begriff der [Stetigkeit](#) durch die Topologie in sehr allgemeiner Form definiert wird.

Variable

In der Programmierung ist eine Variable ein abstrakter Behälter für eine Größe, welche im Verlauf eines Rechenprozesses auftritt. Im Normalfall wird eine Variable im Quelltext durch einen Namen bezeichnet und hat eine Adresse im Speicher einer Maschine.

Vektorgrafik

Eine Vektorgrafik ist eine Computergrafik, die aus grafischen Primitiven wie Linien, Kreisen, Polygonen oder allgemeinen Kurven (Splines) zusammengesetzt ist.

Voxel

Bei einem räumlichen Datensatz, der in diskretisierter Form in kartesischen Koordinaten vorliegt, bezeichnet Voxel den diskreten Wert an einer XYZ-Koordinate des Datensatzes. Voxel (zusammengesetzt aus dem englischen *volume* *vox* und *el* von *elements*[1]) bezeichnet in der Computergrafik einen Gitterpunkt („Bild“punkt, Datenelement) in einem dreidimensionalen Gitter. Dies entspricht einem Pixel in einem 2D-Bild, einer Rastergrafik.