Virtual Reality 1

Dongho Kang, Jaeyoung Lim, Soomin Lee and Jaeryeong Choi

[1]{kangd, jalim}@ethz.ch

1. Chapter 1: Introduction into Virtual Reality

1.2. The development of the computer

- **the constant increase of computational power available in modern computer** → today's abundance of virtual reality.

Numbers

- 6000 years ago: the earliest use of numbers.
  - 3300 B.C. Egypt: battle report in hieroglyphs.
- number system
  - The first number systems exist in hieroglyphs e.g. $1 =$ *line*, $10 =$ *horseshoe* ...
  - Roman:

I $= 1$	V $= 5$
X $= 10$	L $= 50$
C $= 100$	D $= 500$
M $= 1000$	

$$2388 = \text{MMCCCLXXXVIII}$$

  - Addition-system (Roman, Greeks, Syrians, Slavic tribes ...): large numbers are made by adding up the symbols
  - Indo-Arabian: the system used until today (1, 2, 3, 4, 5, 6, 7, 8, 9, 0)
    * more complicated calculation can be performed very easily.
    * very forgery-proof
    * 6th century: Indian to Mesopotamia (Severus Sebakt)
    * 10th century: came to Europe (Leonardo Fibonacci)
    * 16th century: printing numbers were introduced (Adam Ries and Albrecht Dürer)

Calculation and calculation machine

- First means to calculate
  - calculating with the fingers - Rome: 40 finger positions, displays up to 200000
  - tally stick (Europe), knots (native Americans)
- The first scientific calculation: **abacus**
- **Written calculation became popular (since 15c)**
  - eased calculations and reduced the required time.

- lowered the failure rate.

- The first calculation machine (W. Schickardt, 17c) **Figure 1.16, page 1-12**
  - add numbers up to 6 digits
  - multiplication were possible.
  - toothed gears and slide rules.
- B. Pascal(1623 - 1662) **Figure 1.17, page 1-12**
  - made money with calculation machine (the first one)
  - add and subtract numbers with 8 digits.
  - more than 50 pieces were produced.
- G. W. Leibniz (1646 - 1716) **Figure 1.18, page 1-12**
  - introduced the machine suitable for all four basic calculation.
- The calculating machines of 18c: more reliable, but became a piece of jewelry than a device for daily use **Figure 1.19, page 1-13**
- C. X. Thomas (1785 - 1870) **Figure 1.20, page 1-13**
  - precise, efficient and ergonomic.
  - "arithmometer": multiplied two numbers with 8 digits in 18 sec.
  - more improved later
- **Programmable calculation machines** (C. Babbage, 1792 - 1871)
  - punched cards for programming.
  - could run the program autonomously.
  - used steam machine and weights, but failed to realize his idea.
- **Calculation machine with electrical devices** (H. Hollerith, 1860 - 1929)
  - **Figure 1.22, page 1-15**
  - control mechanism activated by electrical devices (relays, motors ...) with punched cards
  - American Census in 1890: shorten the time for analysis 7 yrs $\rightarrow$ 1 yrs
  - founded IBM

Computer era

- Theoretical background of the first programmable electrical calculation machine (The first computer): **L. Couffignal and A. M. Turing**
- The first computer: **ZUSE Z3** by K. Zuse (1941)
  - Z1 (didn't work) $\rightarrow$ Z2 (simple tests) $\rightarrow$ Z3
  - could handle 64 numbers with 22 digits.
  - 2600 relays and 8 line punched tape
  - Z4 (1950s) worked at ETH
- **The first encryptor & decryptor**: Enigma vs Colossus (1941)
  - MARK 1 (1943) in US
  - MARK 2 (1944) in US
- **The first electronical computer, ENIAC (1946) Figure 1.25, page 1-16**
  - J. P. Eckert, J. Mauchley and H. H. Goldstine
  - 18000 tubes, 1500 relays, 1000 capacitors and 6000 switch.
  - very fast... 0.0002 sec for addition, 0.0028 sec for multiplication
- **Transistor** invented (1948) **Figure 1.26, page 1-17**
  - **smaller, faster, less heat, longer life cycle, manufactured automatically**
  - **switch: relays $\rightarrow$ tubes $\rightarrow$ transistor**

- **single chip computer(CPU)** (Integrated Circuits)
  - 4-bit processor INTEL and Texas Instrument in 1970
  - 8-bit (1972), 16-bit (1978) ... now 32 bit / 64 bit
- **Personal Computer (PC)**
  - because of IC, the size of computers could be drastically reduced
  - Kenbak1 (1971): the first single chip computer
  - ALTAIR 880 (1974): mass market
    - **Basic**: a new programming language developed for ALTAIR
  - Apple II & Commodore PET: the first commercial success
  - IBM PC became very popular
- SGI were used for high-performance computer graphics: Virtual Reality...

Moore's law and the development of computer

- **Moore's law** (1965): **the amount of transistors** in a computer is doubled all 18 months. **Figure 1.34, page 1-22**
- but also the required space for the transistors becomes smaller and smaller. i.e. **the available memory per chip** is increases: x4 every 3 yrs
- **chip size** also increases: x2 every 10 yrs
- **CPU speed** increases: x2 every year

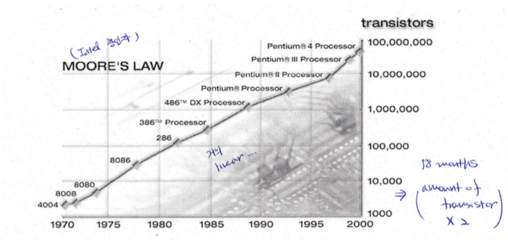


Figure 1.34 Moore's law (https://www4.in.tum.de/misc/perlen/perlen-folien/schaetz-Produktivitaet.pdf)

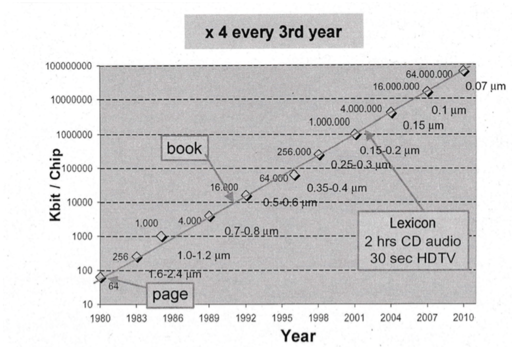
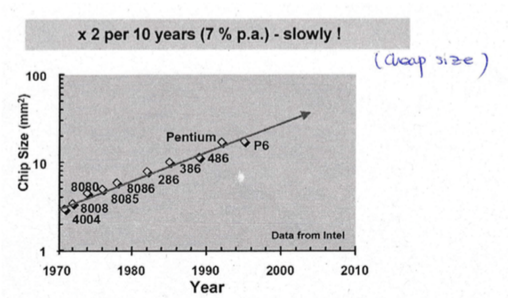
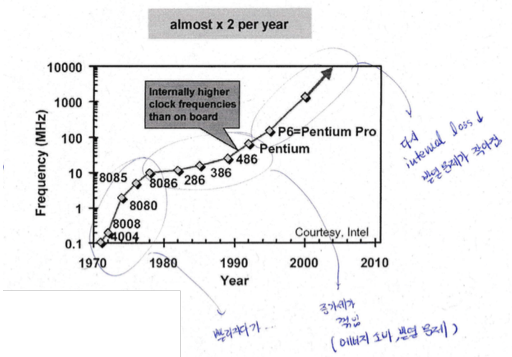


Figure 1.35 Available memory per chip





1.3. Definitions

- Reality (exist): real object $\gg$ virtual object
- Mixed Reality
  - Extended Reality (Augmented Reality): real obj $>$ virtual obj
  - Extended Virtuality (Augmented Virtuality): real obj $<$ virtual obj
- Virtual Reality (non exist): real obj $\ll$ virtual obj
- **Note: Figure 1.39, page 1-25**

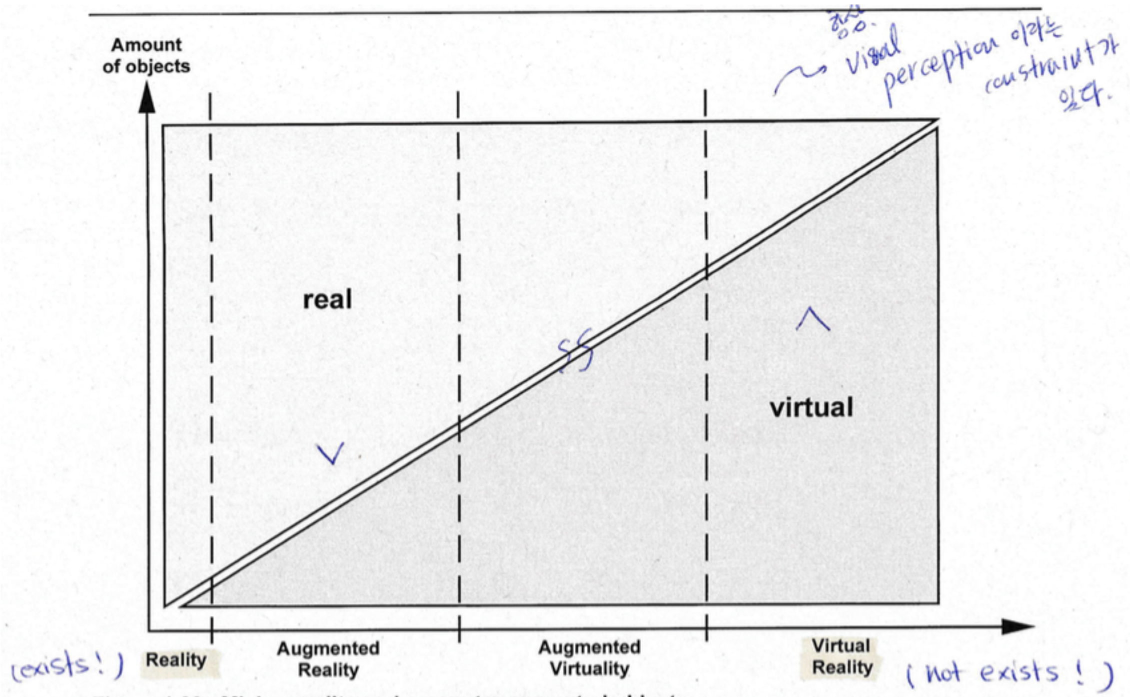


Figure 1.39 Mixing reality and computer generated objects

1.3.1. Extended Reality (augmented reality)

- **real object > virtual object**
- the computer generated objects are characterized by the fact but usually differ from reality in scale, shape or color.
- e.g. AR on copy machine: indicate parts to be fixed

1.3.2. Extended Virtuality (autmented virtuality)

- **real object < virtual object**
- comparing to AR, virtual objects are much more detailed and fit much better to reality.
- e.g. Reconstruction of ruins **Figure 1.40**

1.3.3. Virtual Reality

Definition

- Former Definition: a possible or considered reality, which is not available yet.
- Complete Definition: **"Virtual Reality is a computer generated, interactive and three-dimensional environment, in which the user is completely immersed"**
  1. a fictive and non-real world (e.g. TV, movie, dream are fictive and non-real world)
  2. "cybernetic room" or "cyberspace": controllable and manageable room, in which information is processed. (e.g. chess, board games)

3. generated by computer, using specialized program: imaginary reality can only be reached by processing numerous infos.

The VR reference model

- **Figure 1.41, page 1-27**
- Three component (three axis)
  - interaction (none → interactive → immersive)
  - semantics of data (none → static → dynamic)
  - presentation (single event → sequence of events → real time)
- optimal VR application = <u>immersive interaction</u> + <u>dynamic semantics</u> + <u>real time presentation</u>
- Note, but not each VR application requires the maximum amount of all three criteria.

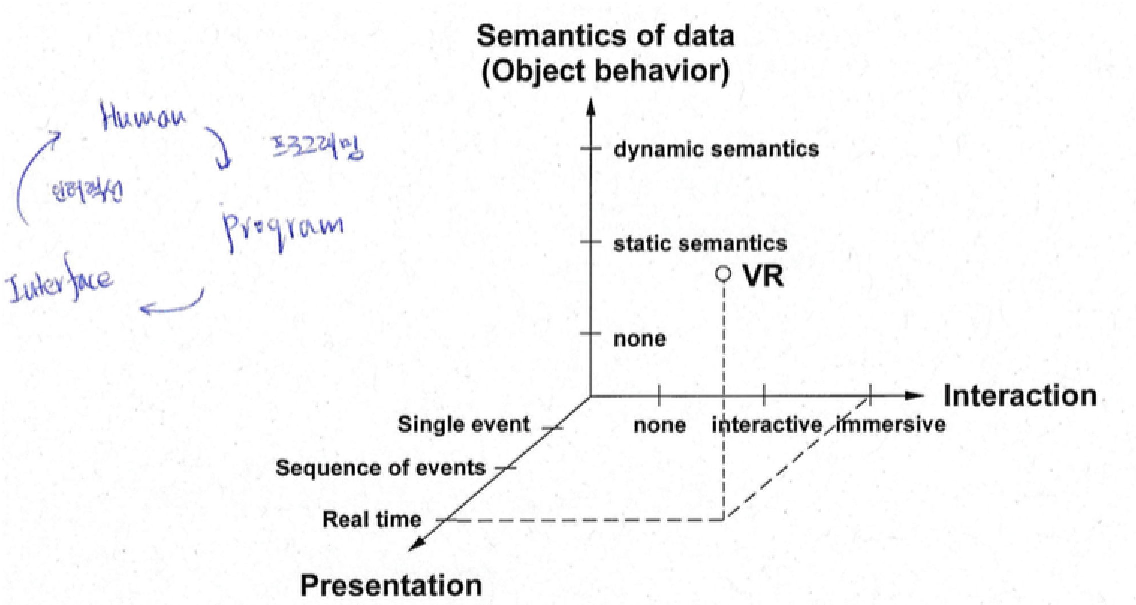


Figure 1.41 The VR reference model

I^3 - Immersion - Interaction - Imagination

- technology provided by VR ...
  - is used to immerse into the virtual env. (immersion)
  - is used to interact with its objects. (interaction)

    → VR addresses all perception channels of the user and thus he is convinced of the reality of VR env. (imagination)
- i.e. Two prerequisites for VR: immersion & interaction (**immersion & interaction** → imagination)
- Interaction:
  - **interplay of information, action and reaction.**

  - describes the effect of computer generated models onto the **human perception channels.**
  - continuous: continuous interaction with the virtual environment **(low delay)**
  - reaction: every movement of the user causes a reaction in VR (note: difference with movies)
  - generation / modification of geometry displayed, visual, acoustic, haptic, olfactory, and physical properties (e.g. collision, material prop., torque, DOF ...)of virtual object (copy or supplement real object)
  - interactivity: ability of the system to allow a user's interaction, while the application is running.
- Immersion:
  -

2. Chapter 2

3. Chapter 3: Introduction into Computer Graphics

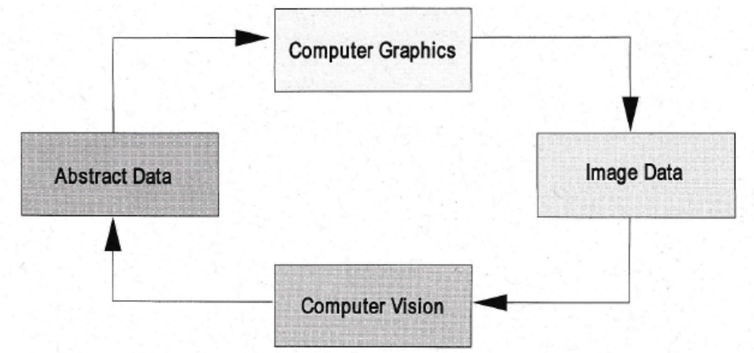
3.1. Introduction

- Quick development in computer graphics: from an expensive toy to an attractive research field
  - main reason: human receives most of the information through the visual perception channel
- Rapid development concerning hardware, software, and applications
  - graphics can be easily accessed and understood by different cultures/people/languages (high synergy among people)

3.2. Why do we need Computer Graphics?

- The representation of texts is also a special form of graphics
- Human being seems to have a better access to images and pictures than letters and numbers
  - e.g. complex simulation results by visualization, logo of a company

Computer Graphics and Picture Recognition



- When a human perceives the surrounding world, the brain generates an abstract ”data model”
- Computer Vision: extract relevant data out of existing graphics or images
- Computer Graphics: display the data

3.3. Applications of Computer Graphics

CAD

- Construction (Functional Design)
  - the requirements to an assembly part are well defined by function and dimension
  - all drawings on a CAD system are a complex relationship between dimensions, constraints, and material properties, which describe a part very well
- Design (Aesthetic Design)
  - do not focus on the dimension of a part, but factors like ergonomics or aesthetics
  - many suppliers of design software try to combine the relatively free aesthetic design field with the very compulsory functional design, but hasn’t been successful due to difficult exchange of data between both worlds

Gaming

- Caused a significance increase of graphic performance in the private computer sector
- Cheap components and fast, high- quality software were developed due to the growing demands of players and the competition of the industry

Visualization

- Visualization is the ancestor of 3D computer graphics
- **Definition: Visualization addresses special properties of human perception to visualize and to represent information, which is extracted from large amount of data**
- Visualization intends to:
  - display structures, models, trends, anomalies, and relationships
  - give an overview of large amounts of data
  - give support by means of an easy modification of parameters when searching for interesting regions in a large data field

  These points can be summarized in the mantra of Ben-Shneiderman:

  overview, zoom-in, filtering, details on request

  Thus, a good visualization has the following properties:
  - it prevents misinterpretation and ambiguity
  - it optimizes the perception of subtle properties
  - it allows displaying more data at a time
- Typical data sets which need to be visualized very often:
  - MRI (the density can be visualized by different colors and transparencies)
  - CFD (the location and the direction of movement of points with 6 DOF, color and particles, size or shape can display the information)
  - Financial Data (display in a diagram, so that a user can see correlations between variables)
  - CAD (3D data with additional information for edges, corners, surfaces, and surface properties. Complex data structures are used because the data is used not only for visualization but also for other processes. Complexity of the models strongly depends on the display quality)
  - Statistical data sets (the correlation between different parameters only can be seen through visualization)
- Different kinds of visualization in the following table. The different categories do overlap in many cases, "Illustration" has an exceptional position since it can be used everywhere

	Static	Dynamic
Interactive	Multi-dimensional vis.	Simulation
Static/Batch	Data visualization	Simulation

- Data Visualization
  - The most common way would be Excel-like presentation (e.g. pancake diagram, 3D graphs)

- Cybernetic Visualization (Simulation)
  - The main interest in the visualization of a cybernetic application: speed and possible detection of errors and irregular behavior of the simulated system
  - The classical way: showing data in a 2D graph as a time-dependent value
- Multi-dimensional Visualization
  - when much data has to be displayed simultaneously within one image
  - every point in the space can represent 3 independent values + color, line thickness, transparency, different shapes, etc.
  - other ways exist but a compromise has to be made concerning the simplicity (e.g. switch between different data visualization planes on any given point of a curve or shape)
- Statistics
  - typically statistics contain many different values for one single measurement.
  - the visualization's task is showing possible correlations among the different values
  - only valid for the given constraints

Illustration

- Illustrations are 2D add-ons to a 3D object. They contain much additional information about the 3D representation.
- GUIs
  - the most important means of communication between the user and the computer
  - the most important input device is the mouse
  - basic idea of GUI: pointing on an object is one of the easiest gestures of a human being
  - within the GUI an icon is a small image, which is a representation for a program sequence or for the content of a defined type (they made working with a computer much faster and more comfortable because people can just 'click' on them instead of entering complex instructions)
- Fonts
  - within the visualization, text is used to assign abstract contents like titles, classifications, or dimensions to a graphical object
  - Typesets and fonts are the medium of text (based on typography and applications)
- Layout
  - By placing graphical elements on a surface, the underlying information is structured more clearly and more detailed
  - typical applications: windows-based desktops such as MacOS, Windows (e.g. word wrap)
  - Another kind of layout is characterized by the given application (e.g. simulation of operation elements in a car- the ergonomic placement of the elements is essential for the overall handling of the vehicle)

- Drawings
  - very often used to visualize abstract situations
  - e.g. organization charts or flowcharts, manuals
- Instructions
  - instructions consist of a mixture of text, graphical visualization, photos, etc.
  - very often they visualize products in an abstract and simplified way
  - e.g. service manuals, assembly manuals

3.4. Definition of 3D Graphics

- Definition of 3D graphics: **The field 3D graphics deals with the generation of 3D objects and their representation on a two-dimensional surface(e.g. a screen)**
- the main focus is on projection and visualization of a 3D space on a 2D surface of any kind
- example of 3D graphic devices: video camera, photo camera
- main characteristics of 3D graphics:
  - data acquisition / data transfer / storage
  - transformation / processing
  - data display on 2D
  - input and output
- simple pipeline for data processing:
  $\rightarrow$ Definition of geometry as digital data
  $\rightarrow$ Processing and transformation of data
  $\rightarrow$ Transformation of data to 2D
  $\rightarrow$ Display of the results on a screen

3.5. Rendering Pipeline

- The rendering pipeline is the basis for almost every application in computer graphics (can be slightly modified for special applications such as augmented reality)
- The simplest form of the rendering pipeline:
  $\rightarrow$ Processing and transformation of data
  $\rightarrow$ Transformation of data to 2D
  $\rightarrow$ Display of the results on a screen
- transformation / processing: modify the original data in such a way that it can be displayed well on a 2D surface. contains data modified by the user e.g. modified viewing angle onto the geometry
- transformation of data to 2D: all modified data has to be transformed onto the 2D surface. e.g. removal of hidden objects or surfaces, transferring a curved surface into triangles, or the integration of illumination models
- input/output: reduce the amount of data so that the image can be optimally displayed on the output device (due to limitation in resolution)

3.6. Definition of Geometry in the Computer

- Algorithms for a geometric representation of curves and bodies were developed in order to model hulls, car bodies, and fuselages

- Bézier developed a CAD-system (further developed into the 'UNISURF' system later on), close approximation to geometry
- In order to define geometry, the information should be sufficient but not redundant
- It can be distinguished between volumetric and surface models

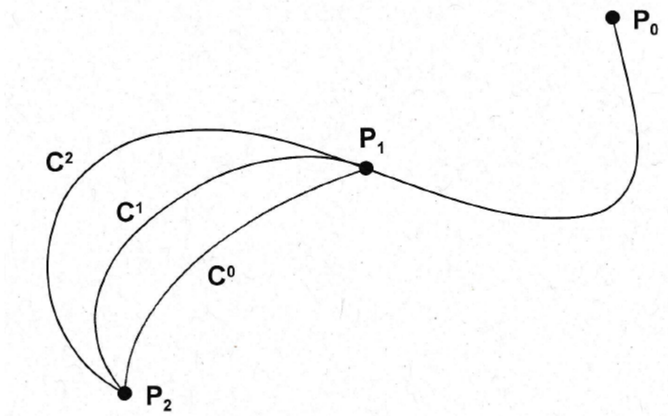
Discrete Definition of Surfaces

- Cloud of Points:
  - the simplest model, every point of the cloud is defined by its coordinates
  - The point cannot be rendered(no information on color or texture) and there is no connectivity among the points(separate parts cannot be selected in the model)
  - 3D-scanner typically provides a cloud of points
  - difficult to create a good surface out of the clouds of points because the acquired data is noisy and erroneous
- Polygons, Tessellation:
  - Tessellation is a method for defining or generating a mesh of geometric basic elements(polygons), which approximate a complex surface
  - Polygon, especially triangle, is one of the most important basic elements of computer graphics
  - "triangulation": the process in which a free-form surface is transformed into triangles (The mesh approximates the surface by triangles and reduces the complexity)
  - additional data has to be considered when rendering (e.g. perpendicular of a surface)
- Polystrips
  - when a point is added to a triangle, a second triangle will be generated
  - advantage: a large amount of surfaces can be generated by a very low amount of data
  - stored in two tables: the point(vertex) table with all coordinates of the triangle's corners and the strip table with all coherent triangles
  - details could be lost since any curve is approximated by straight lines of different lengths
  - many times the triangulation is done at a very late stage in rendering because often working with mathematically exact representations of the geometry is required

Mathematical Description of Geometries

- Typically, the surfaces of objects are represented by approximating functions
- A parametric representation is used for the approximation or interpolation of these surfaces (e.g. parametric polynomials - Bézier, B-Spline)
- Surface models are mainly used- drawback: there are no relationships among the individual parts of the surface (The individual surfaces are not correlated with an object and thus the shape of an object is a question of interpretation)

- possibilities for the mathematical description of curves and surfaces: explicit, implicit, parametric
  - explicit: only one value for y is associated with the variable x
    thus problems arise for closed curves or objects
    it is not invariant to rotations
    only a few geometries can be described by such an explicit notation, rarely used
  - implicit: typically implicit equations have more solutions than required and thus additional constraints have to be considered
    also rarely used in virtual reality
  - parametric: the most common descriptive form
    no equivocations could occur, the definition is invariant to rotations
    infinite slopes can be described by tangential vectors which are finite
    typically consist of rational polynomials of the nth degree, most frequently cubic polynomials are used
    any point of the geometry can be exactly determined by mathematical functions (particular interest for CAD)
- Bézier splines:
  - first used in shipyards
  - goal: interconnect given points by a smooth line
  - points that approximate the curve are called ”control points”
  - mathematical function that delivers the result is named ”base function”, guarantees that the requirement for continuity in the control points is kept by interpolation or approximation
- approximation: a curve approximates given control points / interpolation: the calculated curve has to meet the control points exactly; important in both cases that continuity is fulfilled
- ”parametric continuity”:
  - represented by the letter C and its exponent i which defines the degree of the i^{th} derivation



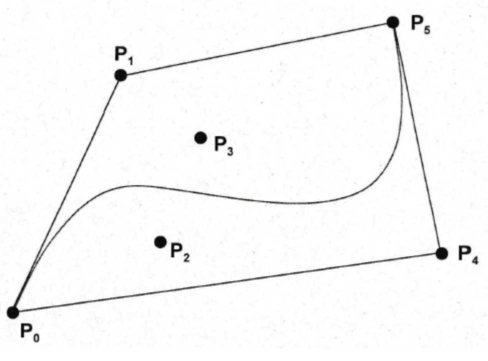
  - C^0 continuity: guarantees that the curve is not interrupted, but it could happen that a curve is not smooth in the point P1 but has an edge instead
  - C^1 continuity: the first derivation is continuous, all tangent vectors in the point P1 have the same slope, the curve is smooth at the point P1 (continuity of tangents)

- C^2 continuity: the second derivation is continuous, the curve looks even smoother, also named as continuity of curvature

	Hermite	Bézier	Uniform B-Spline	Uniformly shaped β-Spline	Nonuniform B-Spline	Catmull-Rom	Kochanek-Bartels
Convex hull defined by control points	N/A	Yes	Yes	Yes	Yes	No	No
Interpolates some control points	Yes	Yes	No	No	No	Yes	Yes
Interpolates all control points	Yes	No	No	No	No	Yes	Yes
Ease of subdivision	Good	Best	Average	Average	High	Average	Average
Continuities inherent in representation	C^0	C^0	C^2	C^0	C^2	C^1	C^1
Number of parameters controlling a curve segment	4	4	4	6	5	4	7

Approximating Splines

- If an approximating curve is described by control points, there is an additional requirement that the resulting curve has to be within the polygon shaped by the control points
- The shape is a complex polygon that encloses all control points like a rubber band
- If the control points are extreme points, they are part of the polygon; otherwise, they are inside of the polygon
- The complex shape allows a good control of the curve (guarantees the curve is always within the visible area)



- Bézier Curve
  - the base function is defined in a parametric form Q(t), t is a variable within [0, 1)
  - the base function depends on the amount of control points; the whole curve is changed by just changing one point
  - in addition to Bézier curve, also the first and last control point belong to the curve Q(t)
  - always inside a complex polygon, because at least three control points are extreme points

- The Algorithm of de Casteljau:
  The base function of a Bézier curve can be calculated by the algorithm of de Casteljau
  idea: choose t and subdivide the first line b_0b_1 at the distance t

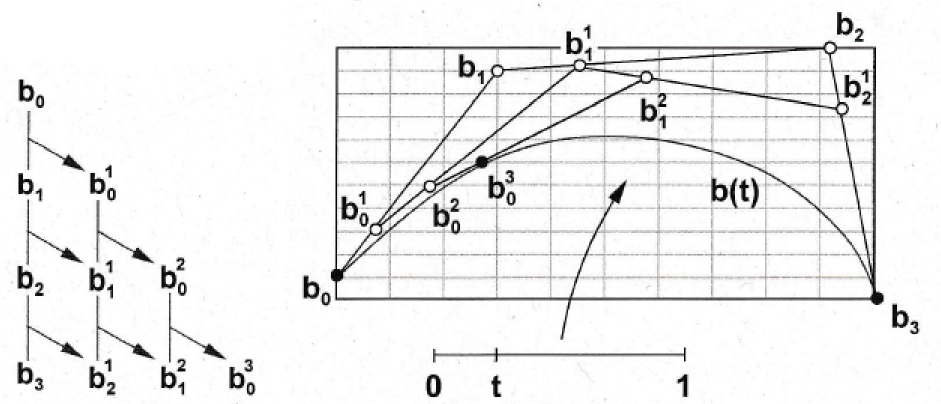
$$b_0{}^1(t) = (1-t)b_0 + tb_1 \tag{1}$$

  This process is repeated for the lines b_1b_2 and b_2b_3, until in total three new points $b_0{}^1$, $b_1{}^1$, $b_2{}^1$ will result
  the calculation is repeated with the three new points and so on
  In general:

$$b_{\mathrm{i}}{}^{\mathrm{r}}(t) = (1-t)b_{\mathrm{i}}{}^{\mathrm{r-1}} + tb_{\mathrm{i+1}}{}^{\mathrm{r-1}} \tag{2}$$

$$b_{\mathrm{i}}{}^{\mathrm{r}}(t) = b_{\mathrm{i}} \tag{3}$$

  i = number of control points; i=0,...(n-r), r = number of lines; r=1,...,n



- For a better approximation of the curve, more control points have to be used which will increase the computing time
- The algorithm of de Castelijau can also be used, if more than four control points exist
- The number of control points(L+1) defines the kind of curve
- For L=3 (cubic Bézier curve), it is further elaborated and simplified as:

$$Q(t) = b_0(1-t)^3 + b_1 3t(1-t)^2 + b_2 3t^2(1-t) + b_3 t^3 \tag{4}$$

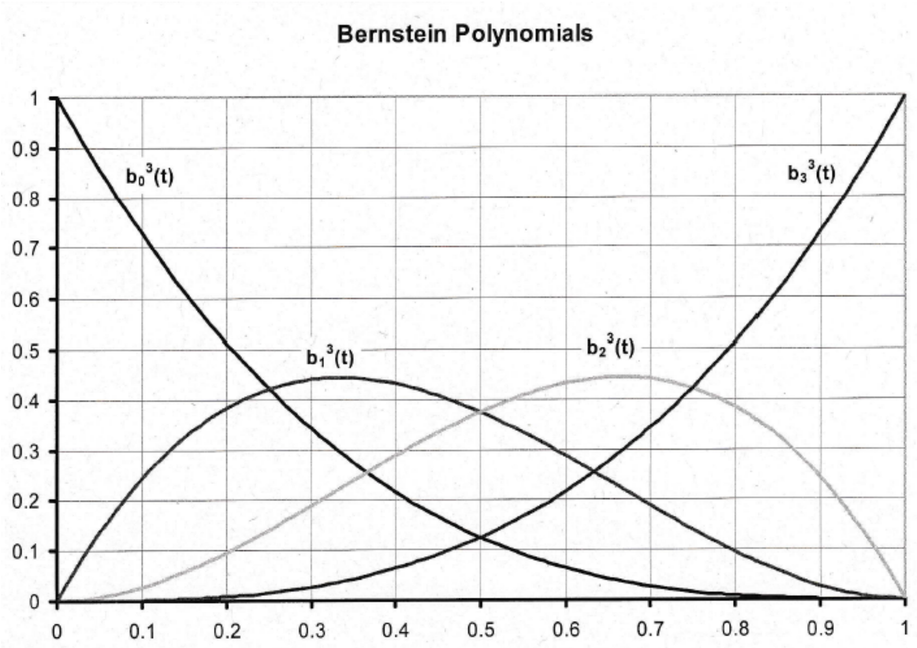
- Bernstein Polynomials:
  Bézier curves can be more easily calculated by using it; do not work recursively but directly deliver the result at the position t by using polynomial coefficients
  When (L+1) control points are given, the function is defined by:

$$Q(t) = \sum_{i=0}^{L} b_{\mathrm{i}} B_{\mathrm{i}}{}^{\mathrm{L}}(t) \tag{5}$$

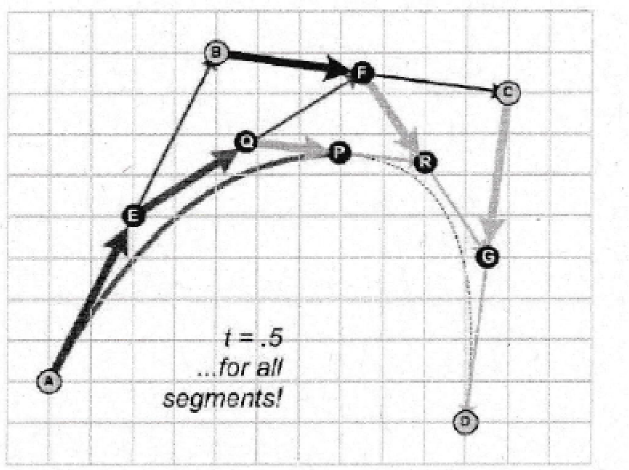
$$B_{\mathrm{i}}{}^{\mathrm{L}}(t)$$

  is the Bernstein polynomial, which is defined as:

$$B_{\mathrm{i}}{}^{\mathrm{L}}(t) = {}_{\mathrm{L}}C_{\mathrm{i}}(1-t)^{\mathrm{L\text{-}i}}t^{\mathrm{i}}, \quad L >= i \tag{6}$$



- If Bézier curves of a high degree are employed, it is difficult to keep the curve smooth
- Bernstein polynomials can also be calculated by a successive superposition of linear Bézier splines



Interpolating Splines

- Boundary Representation - B-Rep
  - Edge representations(or surface models or boundary representation) use 3D polygons to define the limiting surfaces of an object
    The limiting surfaces can be plain surfaces, but also surfaces of higher order
    It allows an exact mathematical description for many geometries, good approximation for the others
  - B-Rep models consist of three object types: surfaces, edges, and corners

- To create on object out of the individual surfaces, the data structure must also contain a topological part(which defines the neighborhood of the surfaces) beside the normal geometric definition
  very common topological list contains 4 parts: a list of points/edges/surfaces/volumetric objects
  This list corresponds to a point-edge-surface model
  advantage: compact storage of all defining elements, since every point has to be stored only once
- The advantage of surface models consists in the complete availability of all topological information
  Surfaces, edges, and points can be stored as individual objects, which significantly increases the flexibility of the object
  one drawback: high calculation effort and complex network structure

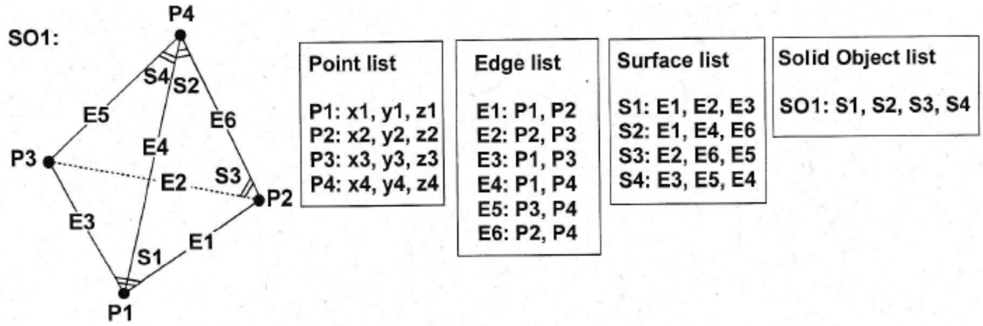


Figure 3.56 Topological list of a tetrahedron

The linkages between the individual tables are shown in Figure 3.57.

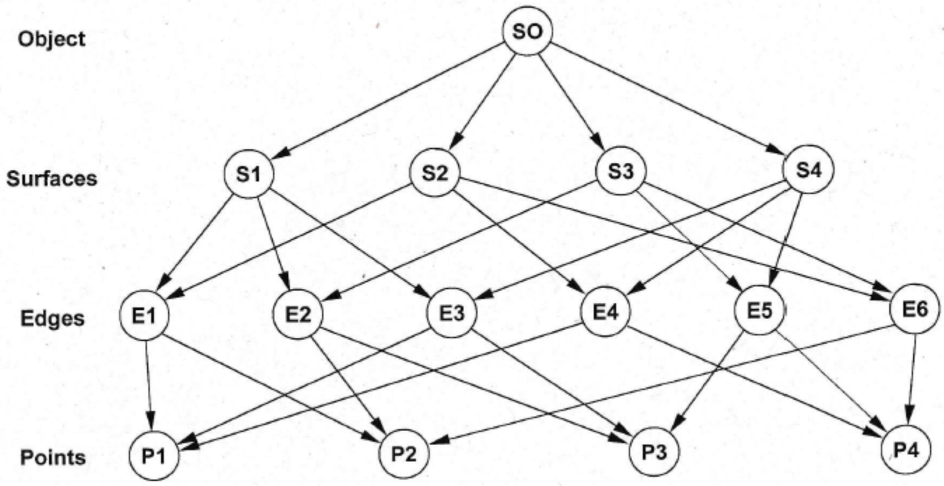


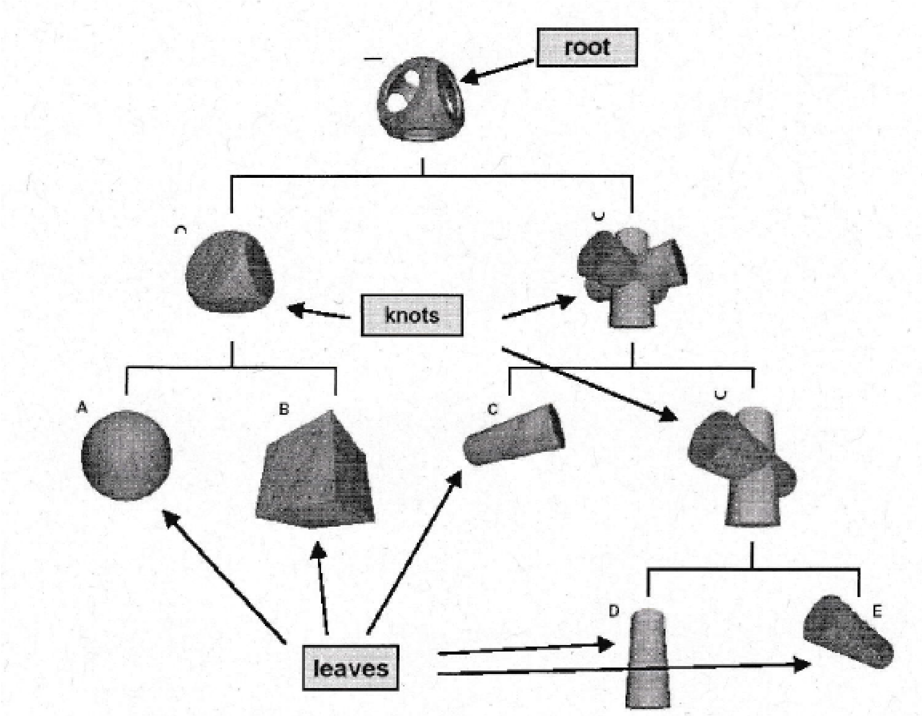
Figure 3.57 Net of linkages between objects, surfaces, edges, and points

Volumetric Models

- The goal is to create objects that can be generally used (not only suitable for special applications)
- Only complete representations of physical objects are accepted
- Volumetric models can be distinguished into:
  cell models: these models discretize the required space of the object

accumulative models: the object consists of a sum of basic elements
generative objects: predefined volumetric elements (primitives) and Boolean operations on the elements
other models: parametric models(e.g. CAD), sweep models, breakdown into elements(FEM)

- CSG(Constructive/Computational Solid Geometry)
  - 3 primitive methods to shape geometry out of simple objects: merging parts(set union), cutting parts out(difference), calculate common parts of objects(intersection) $\rightarrow$ this approach is used by CSG
  - CSG uses simple basic geometries like cones, spheres, cubes which are completely described by mathematical functions
  - Complex objects can be described by a tree of Boolean operations



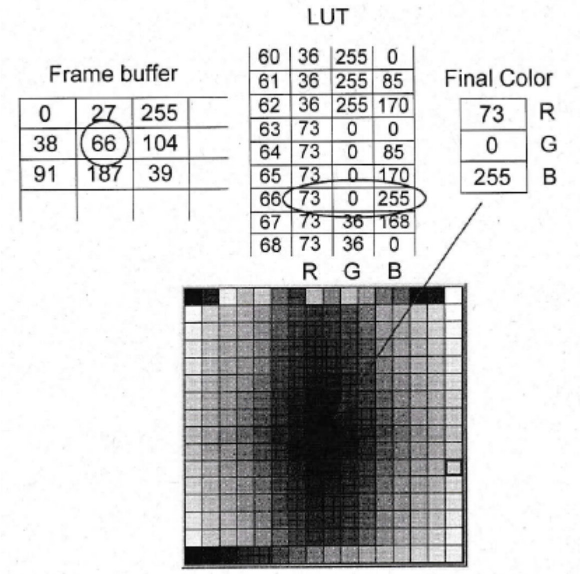
- Cell Models
  - In analogy to pixel models, cell models use voxels(volume elements) of uniform size to represent volumetric models
  - the voxels are arranged in a regular 3D grid and are represented by the coordinates of the cell's center
  - the maximum resolution is defined by the cell size
  - the representation of 3D objects using cell models is suitable for the computation of volumes and other Boolean operations; however the geometrical elements(corner, edge, surface) can only be represented imprecisely or with a large effort

3.7. Color Definitions in the Computer

Color Depth and Color Palettes

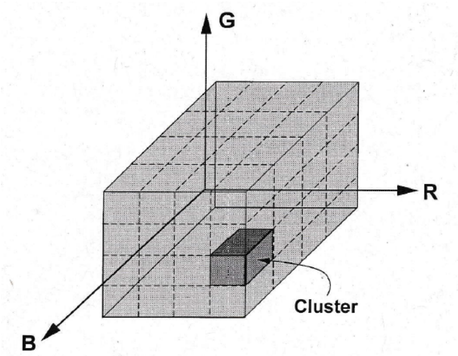
- The maximum number of colors to be used by the computer is defined by the color depth, i.e. the amount of bits that describe a color

- (number of colors) = $2^{colordepth}$ (e.g. color depth 1 means 2 colors can be used)
- If the color depth is smaller than 8 bits, color palettes(color look-up tables, LUT) are used
    - color palettes: an array in which each entry defines exactly one color
    - thus, this array contains color information for each pixel
- In the picture memory, references to the LUT exist
    - if colors change, the entry in the LUT is modified while the picture memory stays unchanged
- Each entry in the look-up table consists of a tuple of numbers for the colors RGB
- If the color depth is larger then 8 bits, the color look-up table would become too large and thus color information is directly stored in the picture memory



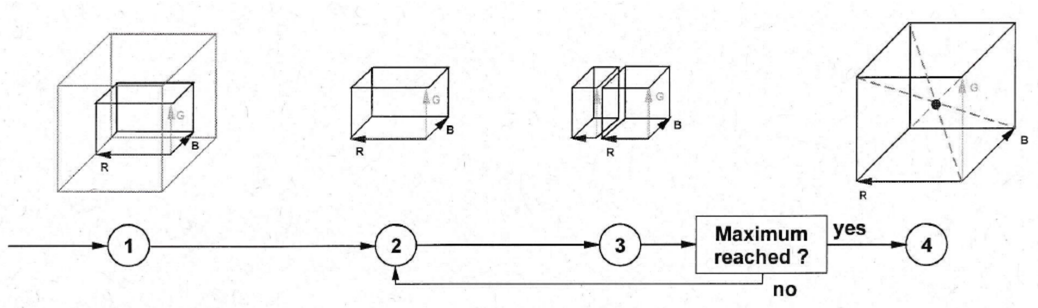
Color Quantization

- Color Quantization: to reduce the required memory for displaying images
    - Due to the immense costs of memory chips, the available graphic memory was restricted for a long time
    - Thus, old graphic cards could only display images with a color depth of 8 Bit with a poor resolution
    - Images with a higher color depth had to be reduced to 8 Bit color depth
    - Even today it is used e.g. graphics which are transferred over the Internet, are compressed in color to diminish the time for transmission
    - Furthermore, human's eye can resolve a limited amount of different colors(color resolution from 5-8 Bits per color) so images can be reduced in color depth without being noticed by the user
- Need for high color depth: typical images with a color gradient or antialiasing are only possible with a higher color depth
- The colors in the original image, which will be assigned to be the same color in the color palette, form a cluster(= color domain)

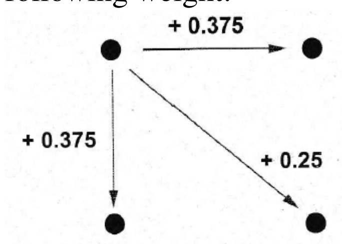


- Pre-clustering procedure:
  - divides the color space into a certain amount of clusters
  - the color of a cluster is one color of the color palette
  - RGB color space is uniformly divided into 256 colors without considering whether the colors exist in the real image or not
  - this method delivers unsatisfying outcomes
- Popularity algorithm:
  - one of the algorithms to create an adapted color palette
  - a histogram of colors is created: it is examined how often every color exists in the original image
    $\rightarrow$ A new color palette is generated out of m(=number of the desired color depth, e.g. 256) mostly used colors
    $\rightarrow$ the colors of the new palette are associated with the original colors of the image i.e. the colors of the palette which fit best the original color
    $\rightarrow$ the rest of the image colors is replaced by the one from the palette, which is very close to it (to determine the most similar color, the Euclidian distance is calculated)
  - drawback:
    very time consuming; no structure is created so the complete color palette has to be searched for every color assignment
    any color details in small segments of the image could be completely wrong
- Median-cut algorithm:
  - In order to find the colors for the adapted color palette, the best approximation of the available colors is created instead of choosing colors from the original image
  - all colors from the original image are marked in the RGB color cube
    $\rightarrow$ the cube is reduced in size until the cloud of marked colors fits the cube best
    $\rightarrow$ the cube is cut parallel to its longest edge(median-cut) in such a way that every sub-cube contains the same number of pixels
    $\rightarrow$ both sub-cubes are contracted until they fit exactly the number of pixels
    $\rightarrow$ again, the sub-cubes are cut along the median line and procedure recurs
    The procedure stops when a previously defined number of cub-cubes is reached
  - in the best case, the resulting segments of the RGB color cube contain only one color that can be assigned to the pixels

- if more than one color is in the final cube, a mean value is calculated



- after the determination of the color palette, the difference(failure) between the needed(real) color and the associated color of the palette is not taken into account
- Floyd-Steinberg algorithm:
  - in order to remove the remaining color steps, smoothens the color transitions
  - the algorithm spreads this failure to the point of the neighborhood with the following weight:



  - the failures are gradually spread over the complete image without increasing the possible amount of palette colors

$$\alpha\delta\Delta\delta \int_a^b f(x) \sum_{i=1}^n \exp(-t)$$

$$f(x) = \exp(-t) \tag{7}$$

$$A = B \tag{8}$$
$$= C \tag{9}$$

4. Chapter 5