

# Chapter 3

## 3D Modelling

**Abstract** Space modelling uses geometric topological elements. The chapter presents all the basic topological elements, as well as the matrix calculus for translation, rotation, scaling, mirroring and perspective. These are the key operations for space modelling.

In order to represent objects or products in a virtual environment, many types of equipment have been used since early times. Besides analogue records in 3D space (such as a sketch, an illustration or a photograph), solutions for the best possible form of digital recording were investigated. When the computer processor and the corresponding mathematical relations appeared they offered, for the first time, an opportunity for a high-quality digital record of 3D space. Of course, this had to be accompanied by appropriate 3D modelling methods, and several procedures and approaches appeared during their development. The approaches were, however, initially very different. After some development, some methods established themselves and proved to be suitable for computer processing. Seeking the right solutions was necessary and important, particularly from the viewpoint of getting a high-quality basis for digital recording. However, the issue of standardizing communications between the user and the computer has remained open.

Historically, development progressed gradually, from simply describing objects with wireframe models and a surface description of 3D models to the solid model, as the most reliable way of describing real models in space (Fig. 3.1).

Increasing computer capacities gradually provided users with new solutions based on different modellers. However, all modeller developers need to follow the global user. Standard imaging has been established in global engineering practice, i.e., it is understood by engineers and technicians in different parts of the world. This was the reason why the large number of original software solutions was gradually reduced to a smaller number of providers for general 3D-modelling equipment. However, there are also developers for specific needs and forms on the market.

To understand the structure of a 3D model it is vital to also understand the fundamentals of the topological definition of a geometric model. Below, the basic topological elements will be presented: point, edge, loop, surface, and volume. This is



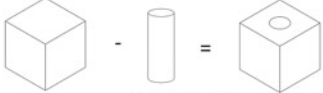
	 Wireframe	 Surface Model	 Solid Model
1960	The development of wireframe and surface models		
1970	The first use of wireframe and surface models	Development of modeling with solid models	
1980	Specific use of wireframe and surface models	The first use of solid models	
1990	General use of wireframe and surface models	Specific use of solid models	
2000	General use of wireframe, surface and solid models		
2005	The introduction of special techniques of wireframe, surface and solid models in products modeling		

Fig. 3.1 Development of 3D modelling

followed by presenting the different types of geometric models, i.e., from a wireframe and surface model to a solid model. The result is geometrical transformations that allow the representation of an object on the screen and its manipulation.

### 3.1 Topological Elements in a 3D Modeller

The topology of an object in space consists of five topological elements, with each of them having its own characteristics. The geometric modeller’s database considers the method of describing the basic topological elements and their relations.

The basic topological elements include: point, edge, loop, surface and volume, all of which can be used to represent a 3D model of an object (Fig. 3.2). The parameters of each of the described topological elements and their characteristics are presented.

Any object, however simple, represents a volume in nature. This volume is described by the surface, dividing the object’s interior from its environment. Each surface is surrounded by a loop, consisting of a finite set of edges. Each edge is defined by at least two points in space, representing its beginning and end.

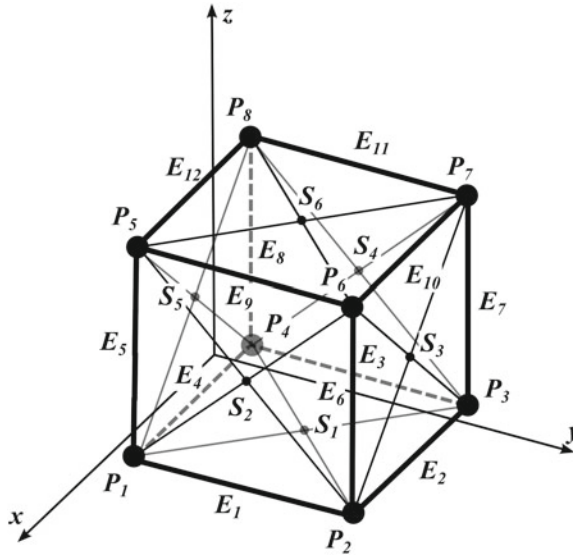


Fig. 3.2 Topological elements on a model

### 3.1.1 Point

The point is the basic topological element of 3D space; its position in space is described by coordinates that depend on the chosen coordinate system.

#### 3.1.1.1 Cartesian Coordinate System

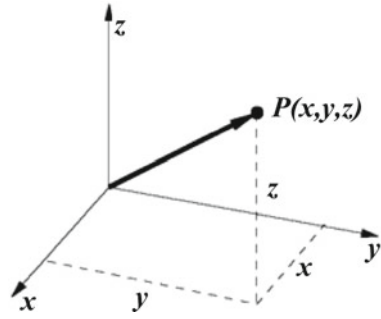
$$P = P(x, y, z)$$

A point in a Cartesian coordinate system is specified by its distance from the main three planes: the  $x$  coordinate—distance from the front view  $y-z$  plane; the  $y$  coordinate—distance from the side view  $x-z$  plane, the coordinate  $z$ —distance from the top view  $x-y$  plane (Fig. 3.3).

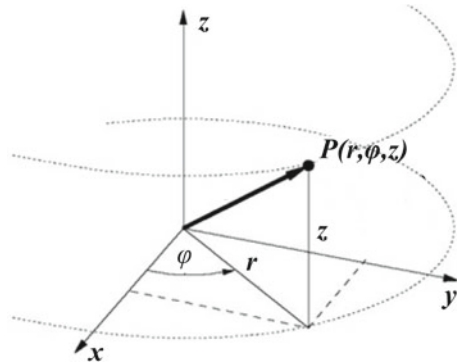
#### 3.1.1.2 Cylindrical Coordinate System

$$P = P(r, \varphi, z)$$

**Fig. 3.3** A point in a cartesian coordinate system



**Fig. 3.4** A point in a cylindrical coordinate system



This is an upgrade of the so-called polar coordinate system, where point T is determined by a distance from the origin, i.e., the local vector ( $r$ ), the rotation of the point around the  $z$  axis and the axis itself by the revolution angle ( $\varphi$ ) relative to the  $x$ - $z$  plane. The ( $z$ ) coordinate is added, representing movement in the  $z$  direction (Fig. 3.4).

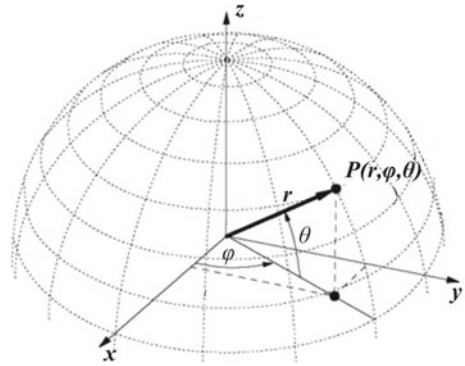
A cylindrical coordinate system is used for different types of rotated or turned parts. It is useful for a computer-controlled lathe. Figure 3.4 presents the ( $r$ ) radius, acquired by moving the knife, mounted on the lathe support, the revolution angle ( $\varphi$ ), the ( $z$ ) coordinate represents movement along the lathe's bed.

### 3.1.1.3 Spherical Coordinate System

$$P = P (r, \varphi, \theta)$$

In contrast to a cylindrical coordinate system, where the height is specified by the  $z$  coordinate, the height in a spherical coordinate system is specified by the revolution angle  $\theta$  between the point's local vector and the  $x$ - $y$  plane (top view). The distance from the centre is specified by the radius of a sphere (Fig. 3.5).

**Fig. 3.5** A point in a spherical coordinate system



A spherical coordinate system is used for all products that are either built or operate in the shape of a sphere. The most well-known application of a spherical coordinate system is Gauss-Krueger's coordinates for determining the positions on a globe. These coordinates are used by surveyors, who determine two types of coordinate: the latitude ( $\varphi$ ) and the longitude ( $\theta$ ). A more accurate determination would generally also require a distance from the centre of the Earth ( $r$ ); however, it varies from place to place in different parts of the world, and instead we use the height above sea level, according to conventional bases in different parts of the world. It is to be expected that some definitions of the radius itself and sea level will change again.

### 3.1.2 Edge

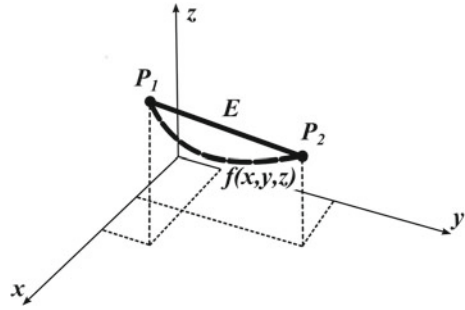
$$E \Rightarrow f(x, y, z)$$

The edge is a topological element—the connection between at least two points (Fig. 3.6). In a linear connection, the edge is represented by the line, defined by its beginning ( $P_1$ ) and end ( $P_2$ ) points.

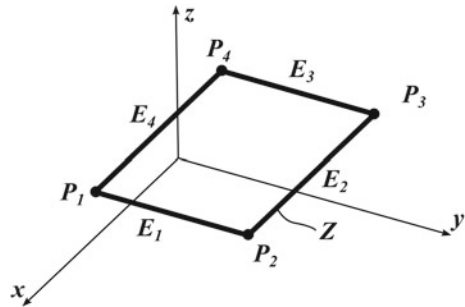
$$E \Rightarrow (P_1, P_2)$$

The edge can be generally presented as a function (curve) between two points that are running in space through a defined point. The function can be of any order. As a rule, it should be as close as possible to the function of a natural phenomenon, taking place on its surface or in its vicinity and defined by the said function. For example, hydraulic phenomena (vessels at sea, the blade of a Kaplan turbine, etc.) or aero phenomena (windmills, cars, car spoilers etc.) are described by the Bernoulli equation, which makes it possible for a surface to be designed by a fourth-order curve, and not at all by a third-order curve.

**Fig. 3.6** Possible connections between two points with different functions in space



**Fig. 3.7** A closed loop in space



### 3.1.3 Loop

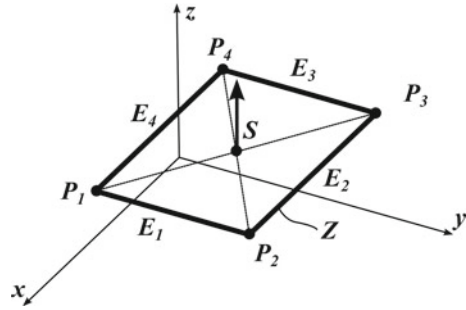
$$L \Rightarrow (E_1, E_2, \dots, E_n)$$

The loop consists of a set of edges, connected one to another in one or another way (Fig. 3.7).

A closed loop represents sequentially connected and closed edges. The said conditions provide logically executed and repeatable computer operations. An open loop is referred to as an edge assembly or polylines. The polylines start by defining the first point, which is then in the following step defined as the last point with  $n$  data polylines. The number  $n$  specifies the number of connecting lines or curves of particular polylines. Such designed polylines can be quickly transformed into different types of curves. The method is very useful for specifying the regression lines or curves with an order  $m$ .

An open loop represents a set of edges where the last edge with its second point does not connect with the first point of the first edge. An open loop can appear with wireframe models. Surface or volume models cannot use an open loop, and surfaces in open loops cannot be defined.

**Fig. 3.8** A topological element—surface



More frequent and useful for modelling purposes is the so-called closed loop with a number of edges finishing in the initial point. Only a closed loop, lying on a plane, can represent a surface.

### 3.1.4 Surface

$$S \Rightarrow L \Rightarrow (E_1, E_2, \dots, E_n)$$

The surface is represented by a closed loop. A surface is defined by a loop and a normal vector to a surface. A surface is required to manage surface and solid models (Fig. 3.8). Wireframe models can be presented without defined surfaces. For simple surface-model presentations, there are simple laws used to create the surface filling. Surfaces sometimes come with surface colour information or even a “pattern” to improve the image of the object itself.

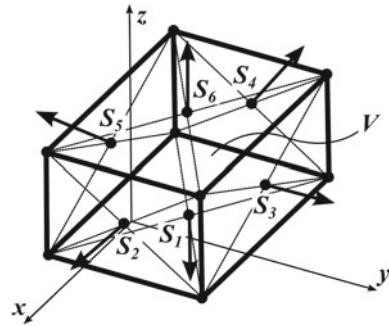
Models that are presented with surfaces only, are very useful for simulating phenomena in the environment. However, when reshaping the model, their major drawback is errors in the surface connection, resulting in inaccurate details.

### 3.1.5 Volume

$$V_1 \Rightarrow (S_1, S_2, \dots, S_n)$$

The volume is the final element in a geometric modeller. The volume is described as a set of surfaces, dividing the object’s interior from the exterior—the environment. The volume is characterized by the requirement that all surfaces need to be connected and all normals should point outwards (Fig. 3.9).

**Fig. 3.9** Volume as a complex topological element



Similar to surfaces, where a loop may not have any undefined edges, the volume also allows no undefined surfaces. Volume elements are required to model solid bodies.

Modellers were also developed in line with the topological structure. Looking at the required parameters for a single topological element, one can see that at the very beginning of modeller development, memory was the main obstacle. With the arrival of sufficiently capable RAM units (exceeding 4 Gb), modeller development accelerated in the direction of digitising objects.

This all suggests that the development of useful methods and modelling largely depends on the development of other systems, which is computer power in this case.

## 3.2 Presenting 3D Models

### 3.2.1 Wireframe Model

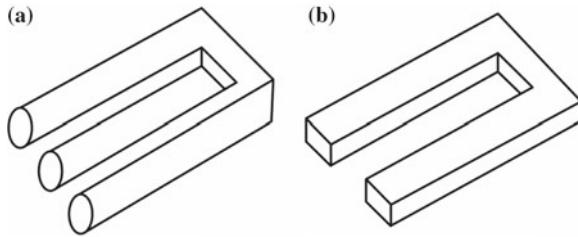
The first 2D and 3D modelling computer programmes were based on two topological elements: the point and the edge. With the development of 3D modelling, other topological elements came into use.

The wireframe model is the simplest way to describe an object in 3D space. Representing a model, its edges are shown as lines, connecting the points in space with a wire.

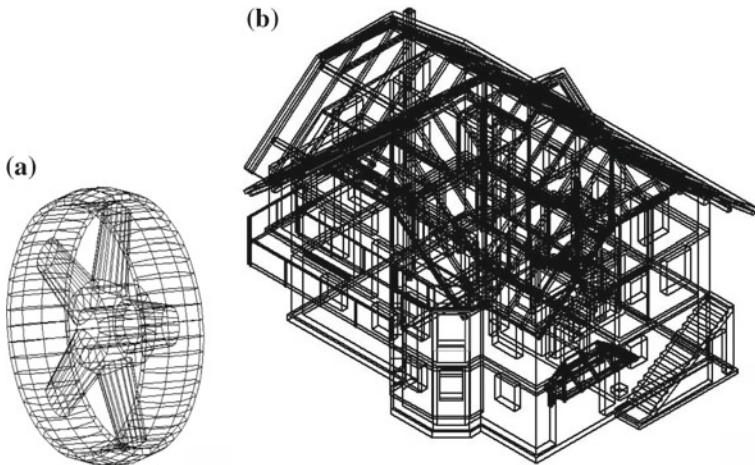
Containing only vertices and edges and their relations, the databases for wireframe modellers are simple. Due to simple relations and a small volume of data, the computation is fast and does not require a lot of computer memory.

A drawback of wireframe modellers is that an image with a larger number of edges becomes unclear. The clear orientation of an object in space can be lost. The incorrect connections of edges between individual points can lead to an anomaly, as shown in Fig. 3.10.





**Fig. 3.10** Anomaly of a wireframe model, caused by incorrect connections between points: incorrect (a) and correct (b) connections



**Fig. 3.11** Examples of a wireframe models: a pulley (a) and a house (b)

Wireframe models are not suitable for computer-aided analyses (CAA) as they do not allow the calculation of even basic quantities, such as the model’s surface or volume.

### 3.2.2 Surface Model

The surface model is an upgrade of the wireframe model. For the wireframe model, the loop is the highest topological model. The surface model requires all loops with defined edges, plus normal vectors for each closed loop. In terms of the surface, the models are divided according to the method of describing the surface. The following description methods are the most commonly used:

- surfaces as parts of the shape of geometric bodies (Fig. 3.12),
- interpolate surfaces (Fig. 3.13),

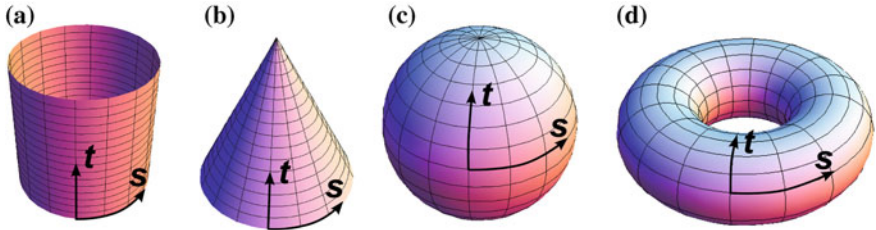
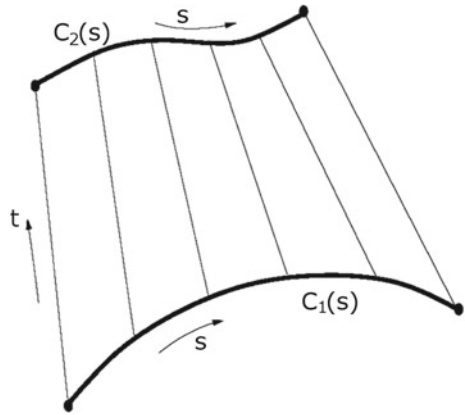


Fig. 3.12 Surface as part of the shape of a cylinder (a), a cone (b), a sphere (c) and a torus (d)

Fig. 3.13 An interpolate surface, defined by two edge curves



- parametric surfaces (Fig. 3.14),
- polygon meshes (Fig. 3.15).

### 3.2.3 Volume (Solid) Model

Solid models can be defined by a Euclidean space, defined by two regions, i.e., the interior one and the exterior one, divided by the boundary of the object. Its boundary is defined by at least one closed surface and/or a set of connected open surfaces. A common feature of all the representations of solid models is that the interior of the object consists of a number of points, geometrically closed by the boundary of the object.

#### 3.2.3.1 Instances and Parameterized Shapes

This method describes simple and similar shapes of objects with the use of basic, i.e., parameterized, shapes. Figure 3.16 shows a couple of examples of new shapes,

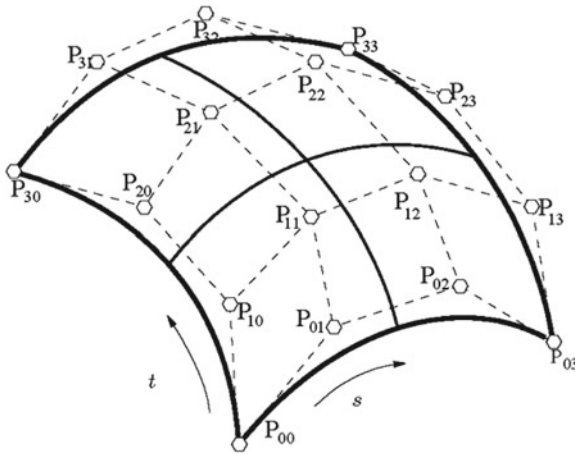


Fig. 3.14 A free parametric surface, described with 16 control vertices

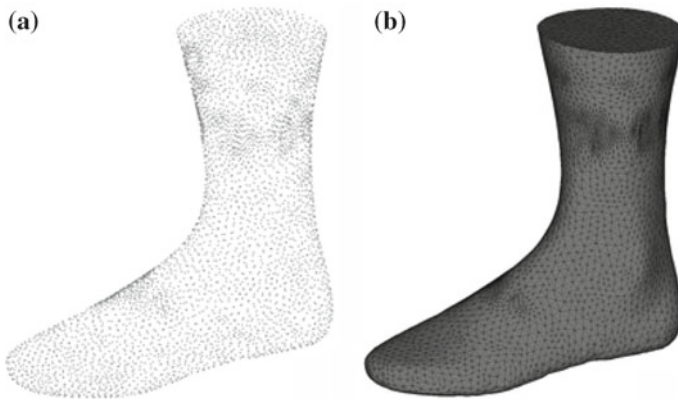
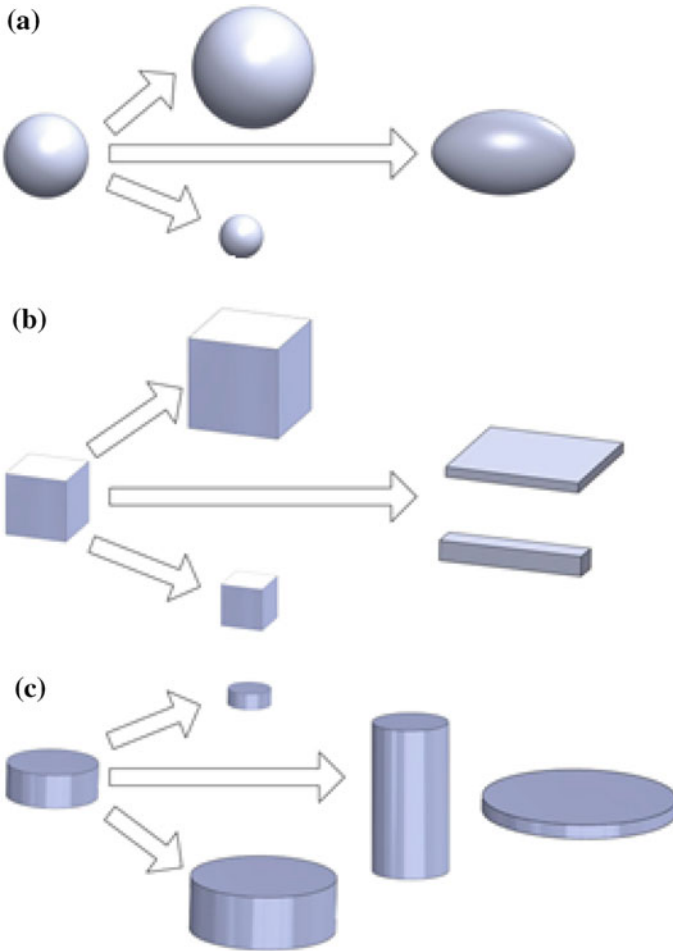


Fig. 3.15 A polygon mesh of a human foot: a scanned surface (a) and a triangle mesh (b)

designed by simple linear transformations of existing models, such as a unit sphere, a cube and a cylinder. A family of similar shapes can be created by parameterizing instances. All the resulting variants can be created by changing the parameters. Figure 3.17 shows an example of an instance and the parameterization of a Z-profile.

### 3.2.3.2 Boundary Representation

A boundary representation (B-Rep) is based on an argument that a physical object is closed from all sides with boundary faces (surfaces), dividing the model from the rest of the space. Each face is limited by edges and they are limited by vertices (points). Figure 3.18 shows a boundary-represented object.



**Fig. 3.16** Instances and linear transformations or better parameters extrude: a sphere (a), a cube (b) and a cylinder (c)

### 3.2.3.3 Constructive Solid Geometry

The constructive solid geometry (CSG) method represents one of the most popular techniques for generating solid models. The method is simple—for both understanding and communicating with the user. A 3D model validation according to this method is simple.

The method is based on the principle that any physical body can be generated as a combination of elementary shapes, i.e., primitives. A large set of primitives can be used. In practice, the most frequently used ones include a rectangular solid, a cylinder, a cone and a sphere, as shown in Fig. 3.19.

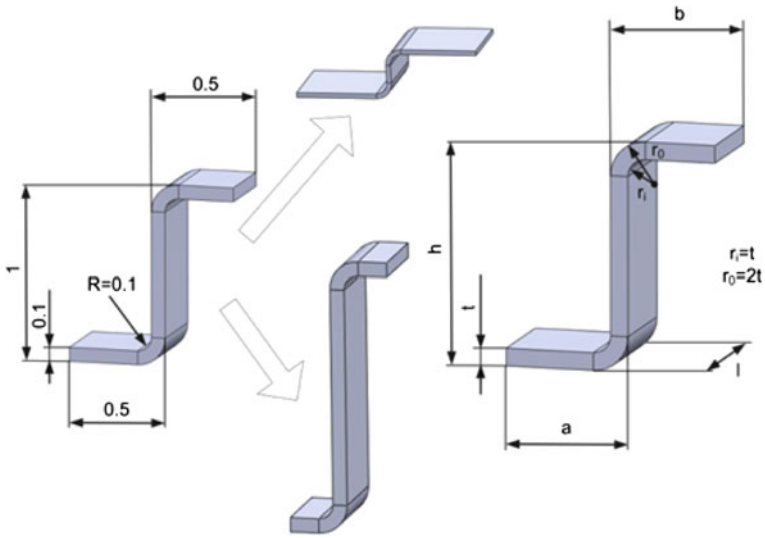


Fig. 3.17 Parameterization of a Z-profile

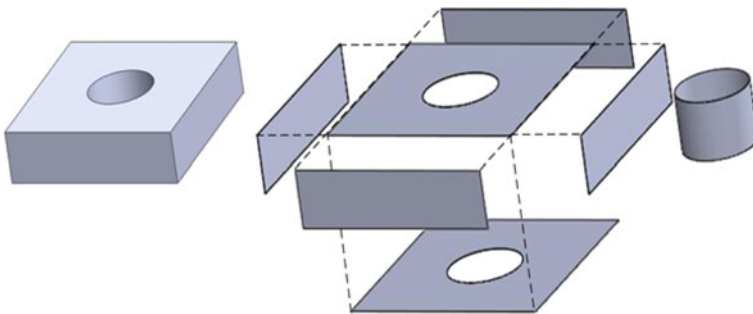
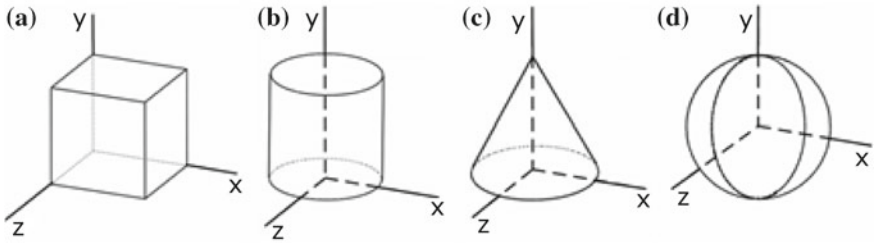


Fig. 3.18 An object with boundary surfaces

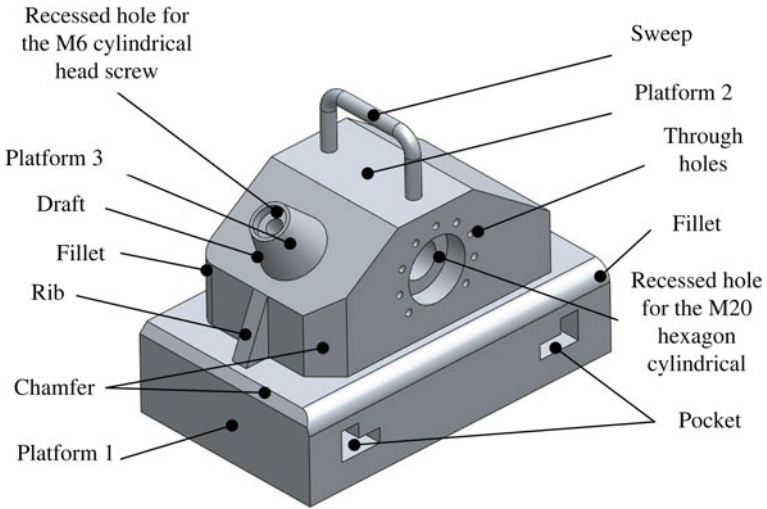
### 3.2.3.4 Feature-Based Modelling

Feature-based modelling is a 3D modelling technique that makes it possible to build a model at a higher level, such as manipulation with the basic geometric entities (point, line, etc.) or primitives. A model is represented as a combination of the CSG and B-rep algorithms. Besides the basic geometric and the topological modelling structure, the method also takes account of higher information levels, such as the geometric characteristics of holes, slots, fillets and other shapes.

A feature represents a set of geometric entities that appear and are recorded in a certain order. It allows—by means of a couple of simple operations—the creation of a large number of geometric primitives that make up individual model parts (Fig. 3.20). The features are very useful for the user, as they allow him or her to upgrade a feature



**Fig. 3.19** Examples of the basic elementary shapes, i.e., primitives: a cube (a), a cylinder (b), a cone (c) and a sphere (d)

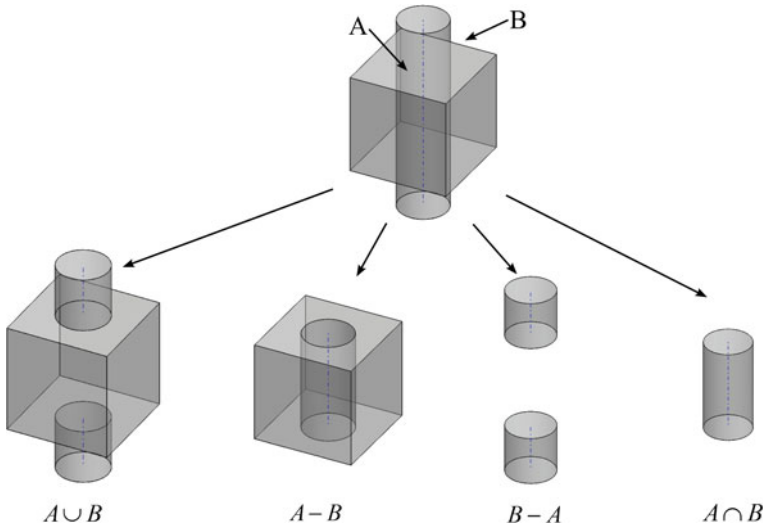


**Fig. 3.20** Using features for modelling a product of moderate complexity

that represents a certain manufacturing technology or its form. The features bring the user close to technological manufacturing operations, which significantly improves the practical value of a modeller.

The basis for feature manipulation (Fig. 3.21) are Boolean operations:

- union,
- difference,
- intersection.



**Fig. 3.21** Boolean operations between the objects (from left to right): union ( $A \cup B$ ), difference ( $A - B$  and  $B - A$ ) and intersection ( $A \cap B$ )

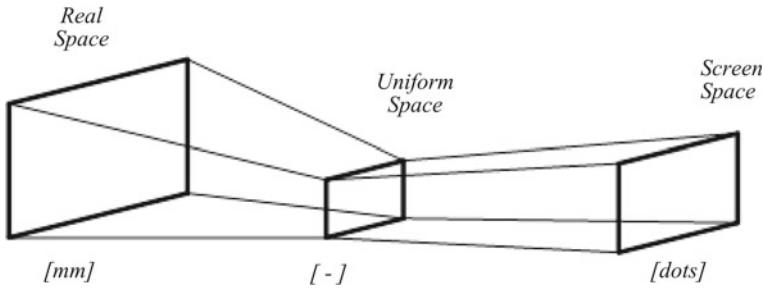
### 3.3 Geometric Transformations

A 3D graphics user would like to observe the scene from different points of view and move some objects in space relative to other objects. These operations are made possible by *geometric transformations*. They are used for the purpose of positioning, orienting and scaling the objects, as well as for mirroring, perspective view, etc. Before describing the transformations, let us take a look at some mathematical operations that make the passage into matrix formulations possible, as these are the easiest way to perform transformations.

When mirroring objects from the real world into the virtual one, it is vital to ensure independent imaging, independent of the size and type of display. In order to achieve that, a new environment, a new space—called a uniform space—has to be created. A uniform space should provide complete neutrality for different types of mirroring from the real into the virtual display world (Fig. 3.22).

Homogenous coordinates are used in a uniform space. A 3D description of a point is translated into homogenous coordinates by adding to Cartesian coordinates  $\{x, y, z\}$  a fourth component  $w$ , also called a homogenous coordinate.

$$\mathbf{p} = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}, \quad \mathbf{p}^H = \begin{Bmatrix} x \\ y \\ z \\ w \end{Bmatrix}$$



**Fig. 3.22** Mirroring from the real into the virtual display world by means of a uniform space

An inverse transformation that projects homogenous coordinates back into a 3D space is called a *projection*. The neutrality to newly introduced coordinates  $w$  is established by its value 1. All the values from the real world are therefore projected onto a display of a certain resolution by practically retaining—in a homogenous space—identical values, identical dimensions.

### 3.3.1 Translation

With the geometric transformation that is to be used to move point  $\mathbf{p}$  to  $\mathbf{p}'$ , the translation is performed by specifying the size of the movement by the vector  $\mathbf{t} = \{T_x, T_y, T_z\}$ . In a matrix equation and using homogenous coordinates, the transformation can be formulated as follows (Fig. 3.23):

$$\mathbf{p}' = \begin{bmatrix} x + T_x \\ y + T_y \\ z + T_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \cdot x + 0 \cdot y + 0 \cdot z + T_x \cdot 1 \\ 0 \cdot x + 1 \cdot y + 0 \cdot z + T_y \cdot 1 \\ 0 \cdot x + 0 \cdot y + 1 \cdot z + T_z \cdot 1 \\ 0 \cdot x + 0 \cdot y + 0 \cdot z + 1 \cdot 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

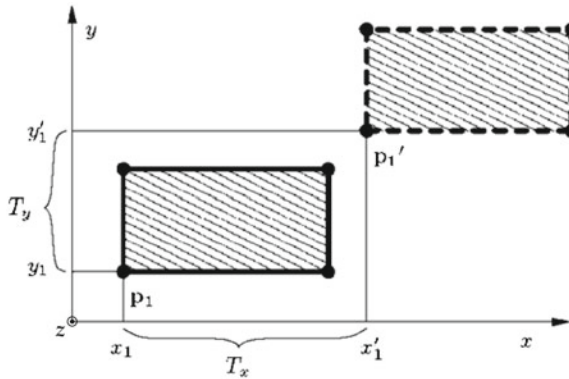
This equation can also be written in a more concise way:

$$\mathbf{p}' = \mathbf{T} \cdot \mathbf{p}$$

where  $\mathbf{T}$  is the transformation matrix:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$





**Fig. 3.23** An example of translating an object on a plane

The matrix yields more accurate values if only translations are performed. But when translations are performed together with rotation, especially with large translation values, it can result in large discrepancies due to the increased influence of the asymmetric values of individual matrix elements. For this reason, translations are generally performed independently and not together with other transformations.

### 3.3.2 Rotation

A rotation transformation rotates a selected point  $\mathbf{p}$  into a point  $\mathbf{p}'$  about one of the coordinate axes by the revolution angle  $\phi$ . Figure 3.24 shows an example of rotating a rectangle. When rotating about the coordinate axis  $z$ , the coordinates of the rotated point are calculated as follows.

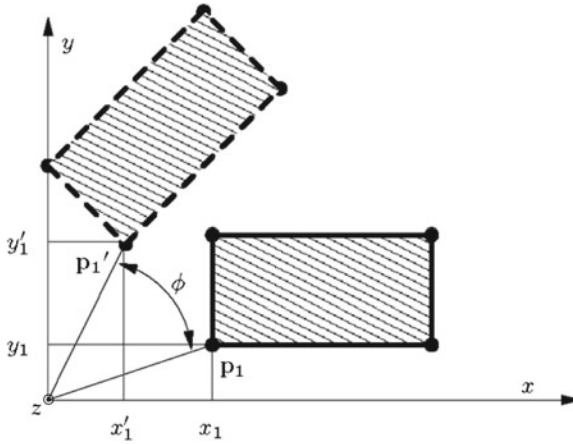
With matrix calculus and using homogenous coordinates, this transformation can be presented as:

$$\mathbf{p}' = \mathbf{R}_z \cdot \mathbf{p}$$

where  $\mathbf{R}_z$  is a transformation matrix, rotating a chosen point about the  $z$  axis:

$$\mathbf{R}_z = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 & 0 \\ \sin(\phi) & \cos(\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Analogous to this is the remaining two rotations in space, i.e., rotations about the  $y$  and  $x$  axes:



**Fig. 3.24** Rotating an object about the z-axis

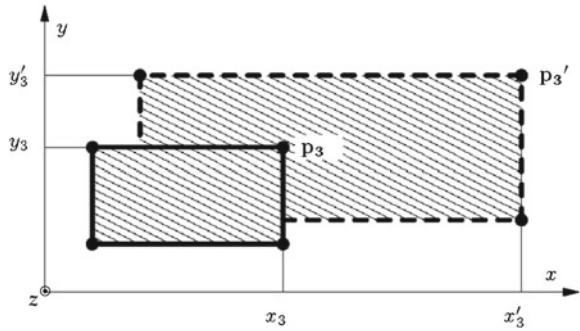
$$\mathbf{R}_y = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\phi) & 0 & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As explained later on, each of these three rotations can be combined in any order into a general 3D rotation. It should be noted that great differences can appear in the ranges between  $0^\circ$  and  $5^\circ$ , and  $85^\circ$  and  $90^\circ$ . The resulting differences in the numerical part can be corrected with two operations: rotation is performed by first executing it in a negative direction and an angle of  $45 - \alpha/2$  is deduced, followed by rotation in a positive direction by an angle of  $45 + \alpha/2$ , where the angle  $\alpha$  represents the required rotation, set by the user. The decision for such an operation should be made when the  $\alpha$  angle is smaller than a specified value. More advanced modellers have this method built into their programme code, while the more basic ones do not include it. As a result, rotations of less than  $5^\circ$ , for example, are becoming unreliable after several repetitions, and distorted objects and ratios between the surfaces begin to appear.

### 3.3.3 Scaling

A general form of a scaling matrix, defining the scaling matrix elements, should first specify the size of the scaling. The size is specified according to the origin of the coordinate system and provides different enlarging or shrinking factors in a chosen direction. To scale the point  $\mathbf{p} = \{x, y, z\}^T$  by the chosen scaling factors  $S_x, S_y, S_z$  the following form of matrix calculus can be used:

Fig. 3.25 Scaling an object



$$\mathbf{p}' = \mathbf{S} \cdot \mathbf{p}$$

whereby  $\mathbf{S}$  is a scaling matrix:

$$\mathbf{S} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

When all three factors are identical this is called uniform scaling. When the scale factor  $S_i$  is larger than 1, this is enlarging in a given direction, and when  $S_i$  is smaller than 1 and larger than 0, this is an object contraction. Figure 3.25 shows an example of non-uniform rectangle scaling, i.e., when the  $S_x, S_y, S_z$  scalars are different for each coordinate.

A scaling matrix has only diagonal elements, which makes it robust and reliable also for larger scaling values, using the **ZOOM** function, for example. It is a known fact that the accuracy of the details can be maintained without problems for enlargements by a factor up to 400.

### 3.3.4 Mirroring

Mirroring is a special form of scaling, where some of the  $S_i$  factors are identical to  $-1$ . When  $S_x = -1$ , this is mirroring across the  $yz$  plane. When  $S_y = -1$ , this is mirroring across the  $xz$  plane, and finally, when  $S_z = -1$ , this is mirroring across the  $xy$  plane.

$$\mathbf{p}' = \mathbf{Z} \cdot \mathbf{p}$$

whereby:

- mirroring across the  $y$ - $z$  plane (Fig. 3.26)

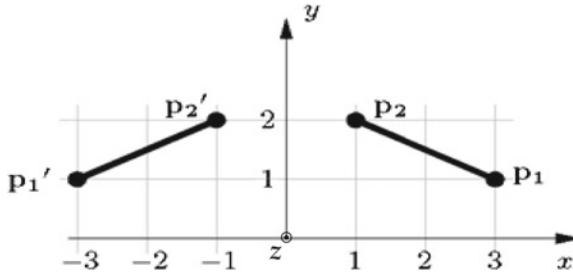


Fig. 3.26 An example of mirroring a line across the  $y$ - $z$  plane

$$\mathbf{Z}_{yz} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- mirroring across the  $x$ - $z$  plane

$$\mathbf{Z}_{xz} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- mirroring across the  $x$ - $y$  plane

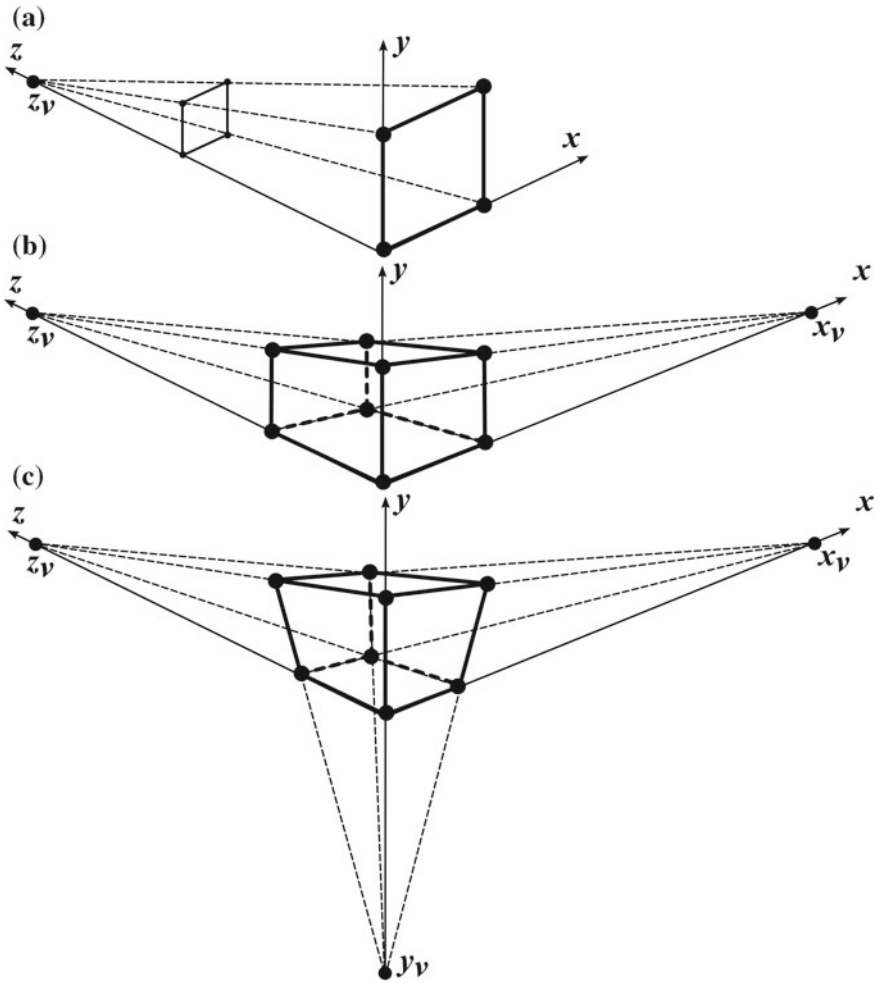
$$\mathbf{Z}_{xy} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 3.3.5 Perspective Projection

A perspective projection is required to show the depth of an object. The objects are deformed by showing the closer objects larger than the more distant ones. The perspective transforms the parallel lines into lines that converge to the vanishing point. In terms of the number of vanishing points, there are (a) one-point, (b) two-point and (c) three-point perspective projections (Fig. 3.27).

General presentation of the perspective projection in a matrix form:

$$\mathbf{p}' = \mathbf{W} \cdot \mathbf{p}$$



**Fig. 3.27** Perspective projection. One-vanishing point (a), two-vanishing point (b), three-vanishing point (c)

where  $\mathbf{W}$  is the transformation matrix for a perspective projection:

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p_x & p_y & p_z & 1 \end{bmatrix}, \text{ where: } p_x = -\frac{1}{x_v}, p_y = -\frac{1}{y_v}, p_z = -\frac{1}{z_v}$$

Let us take a look at projecting the point  $\mathbf{P}$  onto the  $x$ - $y$  ( $z = 0$ ) plane, where the point of the observation is on the  $z$  axis (Fig. 3.28). Considering the similarity of triangles, we can write:

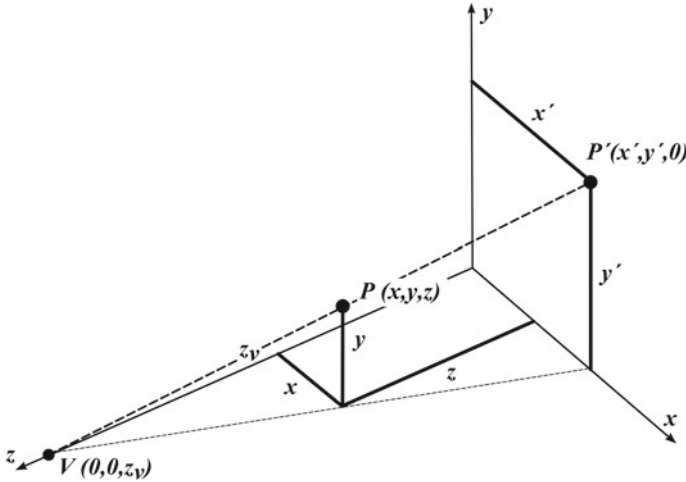


Fig. 3.28 Perspective projection of a point onto a plane ( $z = 0$ )

$$\frac{x'}{x} = \frac{x}{z_v - z} \Rightarrow x' = \frac{x}{1 - \frac{z}{z_v}} \quad \text{and} \quad \frac{y'}{y} = \frac{y}{z_v - z} \Rightarrow y' = \frac{y}{1 - \frac{z}{z_v}}$$

As this is a projection on the  $x$ - $y$  plane, we know that  $z = 0$ . In this case, the perspective projection can be written as:

$$W_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & p_z & 1 \end{bmatrix}, \text{ or } \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & p_z & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 0 \\ p_z \cdot z + 1 \end{bmatrix}$$

Projecting the point  $\mathbf{p}'$  onto a plane ( $w = 1$ ) results in the point  $p'$  in three-dimensional space:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \frac{x}{1 + p_z \cdot z} \\ \frac{y}{1 + p_z \cdot z} \\ 0 \end{bmatrix}$$

Assuming that  $p_z = -\frac{1}{z_v}$ , this results in:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \frac{x}{1 - \frac{1}{z_v} \cdot z} \\ \frac{y}{1 - \frac{1}{z_v} \cdot z} \\ 0 \end{bmatrix}$$