

Zentralprojektion (Perspektive)

1. Koordinatensysteme

Gegenstände oder Landschaften perspektivisch richtig zu zeichnen muss man üben. Aber auch Geübte müssen sich beim Zeichnen oder Malen konzentrieren, wenn sie eine Ansicht so darstellen wollen, wie das Auge oder eine Kamera sie sieht. Das liegt vermutlich daran, dass unser Auge das 2-dimensionale Bild auf der Netzhaut ohne bewussten Zwischenschritt sofort in die 3-dimensionale Welt zurücktransformiert. Der unbewusste Zwischenschritt ist die perspektivische Abbildung. Um die geht es hier. Statt 2- bzw. 3-dimensional schreiben wir im Folgenden kurz *2D* bzw. *3D*.

Wir beschreiben die darzustellenden Objekte in einem *3D*-Koordinatensystem¹, dessen Achsen ein Rechtssystem bilden. Die Koordinaten eines Punktes in diesem System kennzeichnen wir mit X , Y und Z , also Großbuchstaben. Im Ursprung des Koordinatensystems denken wir uns die Linse des Auges bzw. das Objektiv der Kamera – im einfachsten Fall die Blende einer Lochkamera. Ohne Beschränkung der Allgemeinheit nehmen wir an, dass Auge bzw. Kamera entlang der positiven Z -Achse blicken. Die Netzhaut des Auges bzw. Filmebene der Kamera ist somit eine Ebene senkrecht zur Z -Achse, die sich im Abstand der *Brennweite* f hinter Linse bzw. Objektiv befindet. Diese Ebene, wie üblich *Bildebene* genannt, spiegeln wir der Einfachheit halber am Ursprung, so dass sie sich in unserem Modell vor der Linse bzw. vor dem Objektiv befindet (und f damit eine positive Zahl wird). Auch das schränkt die Allgemeinheit unserer Rechnungen nicht ein. Die Koordinaten in der *2D*-Bildebene bezeichnen wir mit kleinen Buchstaben x und y . Sie werden vom Durchstoßpunkt der Z -Achse durch die Bildebene aus gezählt. Abbildung 1 zeigt die beiden so festgelegten Koordinatensysteme.

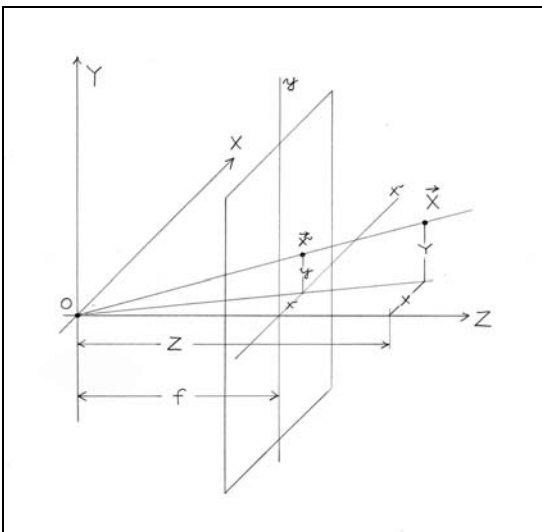


Abbildung 1 Koordinatensystem. Das Auge bzw. Objektiv der Kamera befindet sich im Ursprung O , die Bildebene hat den Abstand f (Brennweite) von O . Sie ist der Einfachheit halber am Ursprung gespiegelt. Der Punkt $(X, Y, Z)^T$ wird in die Bildebene als $(x, y)^T$ abgebildet.

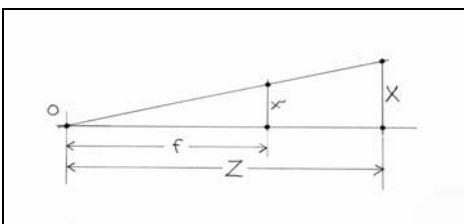


Abbildung 2 Zur Herleitung von Gl. (1). Der Strahlensatz der Geometrie liefert als Abbildungsvorschrift $(X, Y, Z)^T \rightarrow (x, y)^T = (f X/Z, f Y/Z)^T$

2. Perspektivische Abbildung

Das perspektivische Bild eines Punktes $\mathbf{X} = (X, Y, Z)^T$ ist der Punkt $\mathbf{x} = (x, y)^T$, an dem die Ursprungsgerade, die zu \mathbf{X} hinführt, die Bildebene durchstößt². Wie Abbildung 2 zeigt, liefert der Strahlensatz der Geometrie für die Koordinaten des Bildpunktes

$$(1) \quad x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z}.$$

Die perspektivische Abbildung ist damit gegeben durch

$$(2) \quad \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \rightarrow \begin{pmatrix} f \frac{X}{Z} \\ f \frac{Y}{Z} \\ Z \end{pmatrix}.$$

In dieser Form ist sie nicht linear, kann also nicht als Matrix-Vektor-Produkt dargestellt werden. Diesen Nachteil behebt man durch einen Trick: man erweitert die 2D- und 3D-Vektoren \mathbf{x} bzw. \mathbf{X} um je eine weitere Komponente und unterwirft die veränderten Vektoren einer Reihe von Bedingungen. Man geht über zur so genannten *homogenen Notation* (der Vektoren). In dieser Schreibweise heißen die Vektorkomponenten auch *homogene Koordinaten*. Der Vorteil homogener Koordinaten ist, dass sich nicht nur die perspektivische Abbildung $\mathbf{X} \rightarrow \mathbf{x}$ mit den Mitteln der Linearen Algebra behandeln lässt. Auch die Verschiebung (Translation) eines 3D-Punktes kann jetzt als Matrix-Vektor-Produkt geschrieben werden. Näheres dazu weiter unten.

3. Homogene Koordinaten

Zunächst zur Definition homogener Koordinaten: Ein Punkt $\mathbf{X} = (X, Y, Z)^T$ oder, explizit als Spaltenvektor geschrieben

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix},$$

wird durch einen homogenen 4-Vektor $(X_1, X_2, X_3, X_4)^T$ oder (Spaltenschreibweise)

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix}$$

dargestellt, wenn gilt

$$X = \frac{X_1}{X_4} \quad Y = \frac{X_2}{X_4} \quad Z = \frac{X_3}{X_4}.$$

Im einfachsten Fall ist $X_4 = 1$, so dass eine praktische Regel für den Übergang von „normalen“ Koordinaten (*inhomogenen* Koordinaten) zu *homogenen* Koordinaten lautet:

Konvertiere den Vektor $\mathbf{X} = (X, Y, Z)^T$, indem du als vierte Komponente eine Eins hinzufügst. Also

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \rightarrow \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}.$$

Es ist aber auch jeder andere Wert $X_4 = t$ mit $t \in R$ erlaubt. Das heißt,

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad \text{und} \quad t \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad \text{mit } t \in R$$

repräsentieren denselben Punkt $X = (X, Y, Z)^T$. Zum Beispiel sind $(4, 5, 3, 1)^T$ und $(8, 10, 6, 2)^T$ gleichwertige Darstellungen des Punktes $(4, 5, 3)^T$. Umgekehrt *de-homogenisiert* man einen Vektor $(X_1, X_2, X_3, X_4)^T$, indem man die ersten 3 Komponenten durch die vierte dividiert:

$$(3.1) \quad \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} \rightarrow \begin{pmatrix} X_1/X_4 \\ X_2/X_4 \\ X_3/X_4 \end{pmatrix}.$$

In gleicher Weise verfährt man mit den 2D-Punkten $\mathbf{x} = (x, y)^T$ der Bildebene, also

$$(3.2) \quad \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \rightarrow \begin{pmatrix} x_1/x_3 \\ x_2/x_3 \end{pmatrix}.$$

In homogener Notation lässt sich nun die perspektivische Abbildung Gl. (2) durch das Produkt der 3×4 -Matrix (*Projektionsmatrix*)

$$(4) \quad P = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

mit dem (homogenen) 4-Vektor $\mathbf{X} = (X, Y, Z, 1)^T$ ausdrücken. Also $\mathbf{x} = P \cdot \mathbf{X}$ oder

$$(5) \quad \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}.$$

Der homogene 3-dimensionale Vektor \mathbf{x} wird ebenfalls de-homogenisiert. Nach Gl. (3.2) erhält man

$$(6) \quad \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} f X \\ f Y \\ Z \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_1/x_3 \\ x_2/x_3 \end{pmatrix} = \begin{pmatrix} f \frac{X}{Z} \\ f \frac{Y}{Z} \end{pmatrix},$$

und damit unseren aus Gl. (2) bekannten Vektor $\mathbf{x} = (x, y)^T$.

4. Fluchtpunkte

Eine Besonderheit der perspektivischen Projektion sind die so genannten *Fluchtpunkte*. Das sind die Punkte, an denen eine Gerade in der Projektion „verschwindet“, wenn man sie im Raum ins Unendliche verfolgt³. Ist die Gerade in Parameterform gegeben, lässt sich der „Ausflug“ ins Unendliche mathematisch verfolgen: Eine Gerade $G: \mathbf{X} = \mathbf{A} + t \mathbf{R}$ durch den Punkt \mathbf{A} mit dem Richtungsvektor \mathbf{R} , in Komponentenschreibweise also

$$(7) \quad G: \mathbf{X} = \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix} + t \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix},$$

wird nach Gl. (2) in die Bildebene transformiert zu

$$(8) \quad g: x = \begin{pmatrix} f \frac{A_x + t R_x}{A_z + t R_z} \\ f \frac{A_y + t R_y}{A_z + t R_z} \end{pmatrix}.$$

Die Koordinaten des Fluchtpunkts v ergeben sich als Grenzwerte von (x, y) für $t \rightarrow \pm \infty$, also

$$(9) \quad v: x = \lim_{t \rightarrow \pm \infty} \begin{pmatrix} f \frac{A_x + t R_x}{A_z + t R_z} \\ f \frac{A_y + t R_y}{A_z + t R_z} \end{pmatrix}.$$

Da wir die Koordinaten von \mathbf{A} für große t gegenüber denen von $t \mathbf{R}$ vernachlässigen können, ergibt sich als Fluchtpunkt

$$(10) \quad v: x = \begin{pmatrix} f \frac{R_x}{R_z} \\ f \frac{R_y}{R_z} \end{pmatrix}.$$

Seine Lage hängt daher nicht von \mathbf{A} ab, sondern ist alleine durch den Richtungsvektor \mathbf{R} gegeben. Parallele Geraden haben den gleichen Fluchtpunkt. Ein Blick entlang geradlinig verlaufender Eisenbahngleise bestätigt das.

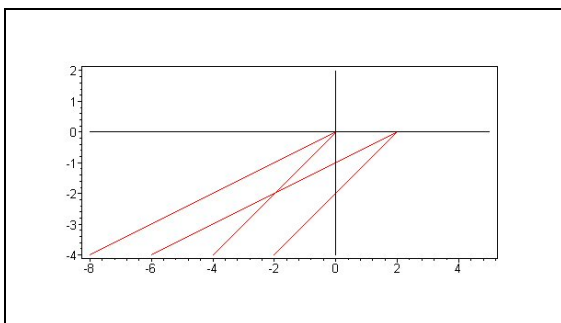


Abbildung 3 Projektion zweier Paare paralleler Geraden mit den Richtungsvektoren $(0,0,1)^T$ bzw. $(1,0,1)^T$. Die Punkte der Geraden wurden mit Hilfe der Parameterform Gl. (8) von Maple für $f = 2$ berechnet und dargestellt. Sie enden in den Fluchtpunkten $(0,0)^T$ bzw. $(2,0)^T$ der Bildebene, wie in Gl. (10) angegeben.

Benutzt man homogene Koordinaten, wird die Rechnung einfacher (aber nicht unbedingt anschaulicher). Gleichung (7) wird dann zu

$$(11) \quad G: X = \begin{pmatrix} A_x \\ A_y \\ A_z \\ 1 \end{pmatrix} + t \begin{pmatrix} R_x \\ R_y \\ R_z \\ 0 \end{pmatrix},$$

wobei die rechte Seite mit einem beliebigen Faktor ungleich Null multipliziert werden darf, ohne die dargestellte Gerade zu verändern (siehe oben). Von dieser Möglichkeit machen wir Gebrauch: Multiplizieren wir mit $1/t$, folgt

$$(12) \quad G: X = \frac{1}{t} \begin{pmatrix} A_x \\ A_y \\ A_z \\ 1 \end{pmatrix} + \begin{pmatrix} R_x \\ R_y \\ R_z \\ 0 \end{pmatrix}$$

und damit für $t \rightarrow \pm \infty$ als Punkt auf der Geraden

$$(13) \quad G(t \rightarrow \pm \infty): X = \begin{pmatrix} R_x \\ R_y \\ R_z \\ 0 \end{pmatrix}.$$

Die Projektion ergibt dann

$$(14) \quad x = P \cdot X = \begin{pmatrix} f R_x \\ f R_y \\ R_z \end{pmatrix}$$

und, de-homogenisiert, den Punkt

$$(15) \quad x = \begin{pmatrix} f \frac{R_x}{R_z} \\ f \frac{R_y}{R_z} \\ R_z \end{pmatrix}$$

in der Bildebene – dasselbe Ergebnis wie in Gl. (10).

5. Translation und Rotation

Homogene Koordinaten sind nicht nur von Vorteil, wenn es um die perspektivische Abbildung geht. Wie schon oben angedeutet, lässt sich mit ihrer Hilfe auch die Verschiebung oder *Translation* eines Punktes im 3D-Raum als das Produkt einer Matrix mit einem Vektor (dem Ortsvektor des betreffenden Punktes) darstellen. In „normalen“ Koordinaten entspricht die Verschiebung eines Punktes $X = (X, Y, Z)^T$ um den Vektor $T = (T_x, T_y, T_z)^T$ der (Vektor-)Addition $X' = X + T$. In Komponentenschreibweise also

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} = \begin{pmatrix} X + T_x \\ Y + T_y \\ Z + T_z \end{pmatrix}.$$

Benutzen wir homogene Koordinaten, lässt sich schreiben

$$(16) \quad \begin{pmatrix} X'_1 \\ X'_2 \\ X'_3 \\ X'_4 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} = \begin{pmatrix} X_1 + T_x X_4 \\ X_2 + T_y X_4 \\ X_3 + T_z X_4 \\ X_4 \end{pmatrix}.$$

Das heißt, es existiert eine Matrix

$$(17) \quad T_R = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(*Translationsmatrix*), so dass gilt $X' = T_R X$. Das ist eine lineare Abbildung in einem 4-dimensionalen „Raum“, dessen Vektoren wir nach Gl. (3.1) in unseren 3D-Raum zurück transformieren können – und die dort eine Verschiebung des Vektors $X = (X, Y, Z)^T$ um den Vektor $T = (T_x, T_y, T_z)^T$ bewirkt.

Um einen Körper (z. B. unseren Quader) unter verschiedenen Winkeln in Übereckansicht, Frosch- und Vogelperspektive darzustellen, muss er um die drei Raumachsen unserer 3D-Welt drehbar sein. Dazu benutzt man normalerweise die üblichen 3×3-Drehmatrizen. Diese Matrizen können so erweitert werden, dass sie die gewünschte Drehung auch im Koordinatensystem unserer 4-dimensionalen Vektoren ausführen. Die 4×4-Matrix (*Rotationsmatrix*) für die Drehung eines Punktes X um die X-Achse mit dem Winkel ϕ lautet

$$(18) \quad M_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Für die Drehungen um die Y- und Z-Achse erhält man

$$(19) \quad M_y = \begin{bmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

bzw.

$$(20) \quad M_z = \begin{bmatrix} \cos \phi & -\sin \phi & 0 & 0 \\ \sin \phi & \cos \phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Im Allgemeinen befindet sich der Körper, den wir darstellen wollen (im einfachsten Fall wieder unser Quader), mit seinem Mittelpunkt nicht im Koordinatenursprung. Ist seine Entfernung in Z-Richtung vom Ursprung beispielsweise A_z , muss er um Achsen gedreht werden, die zwar parallel zu den Koordinatenachsen (X-, Y- bzw. Z-Achse) verlaufen, aber durch den Punkt $A = (0, 0, A_z)$ gehen. In diesem Fall verschieben wir den Körper (Quader) in Z-Richtung zunächst so, dass sein Mittelpunkt mit dem Ursprung zusammenfällt, führen mit Hilfe von M_x , M_y bzw. M_z die Drehung

aus, und schieben ihn schließlich wieder an seine ursprüngliche Position zurück. Eine Drehung um die X-Achse mit dem Winkel ϕ beispielsweise lässt sich dann als Produkt der 4×4 -Matrizen $T_R(A_Z)$, $M_X(\phi)$ und $T_R(-A_Z)$ schreiben. Das heißt, jeder Punkt X des Körpers (in homogener Notation) wird gemäß

$$(21) \quad X' = T_R(A_Z)M_X(\phi)T_R(-A_Z) \cdot X$$

zum Punkt X' gedreht. Multiplizieren wir anschließend mit der 3×4 -Projektionsmatrix P , erhalten wir den zugehörigen Bildpunkt. Entsprechendes gilt für die Drehungen um die Y- und Z-Achse.

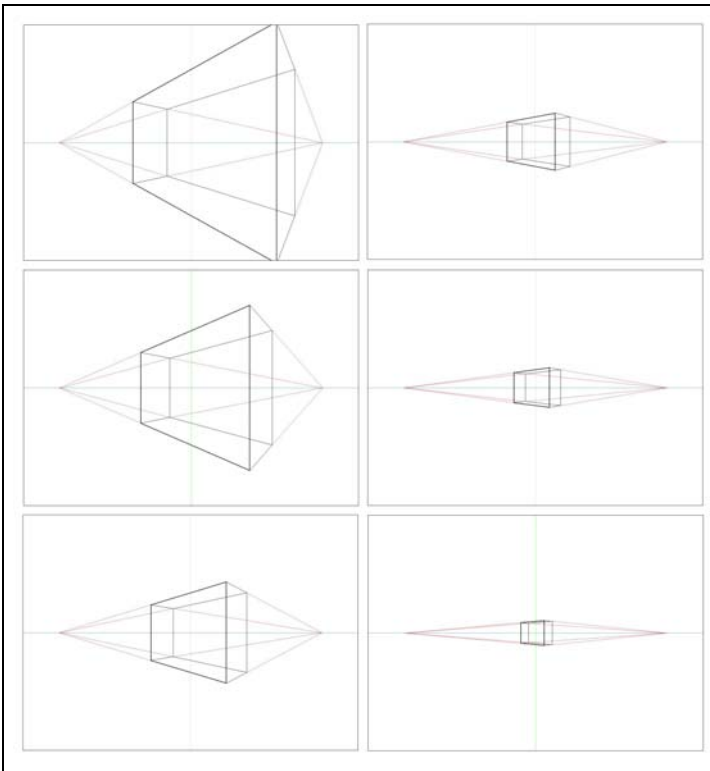


Abbildung 4 Zentralperspektivische Abbildung (Übereckansicht) eines Quaders, betrachtet aus verschiedenen Entfernungen A_Z . Vier Kanten verlaufen parallel zur Y-Achse. Die übrigen Kanten streben auf 2 Fluchtpunkte zu, die auf dem Horizont liegen.

6. Einfaches Java-Programm

Die Matrixoperationen für Projektion, Translation und Rotation wurden als Routinen (Methoden) in einer Java-Klasse zusammengefasst und getestet. Als Testobjekt diente, wie üblich, ein Quader. Je nach der gewünschten Ansicht wurden dessen Eckpunkte mit Hilfe der Matrixoperationen für die Rotation in die gewünschte Lage gedreht – Gl. (21) zeigt das Beispiel einer Drehung um die X-Achse – und dann in die Bildebene projiziert. Abbildung 4 zeigt eine Reihe dieser Projektionen, die das Programm auf dem Bildschirm darstellt: Übereckansichten des Quaders mit zur XZ-Ebene paralleler Grundfläche, der um eine dazu senkrechte Achse (eine Parallele zur Y-Achse) gedreht wurde. Dabei wurde die Entfernung (in Gl. (21) A_Z genannt) variiert, nicht aber die Brennweite f .

Da die Grundfläche des Quaders parallel zur XZ-Ebene ist, haben die vier dazu *senkrechten* Kanten den 3D-Richtungsvektor $(0, 1, 0)^T$. Diese Kanten werden daher durch die Parameterform

$$K: X = \begin{pmatrix} Q_x \\ Q_y \\ Q_z \end{pmatrix} + t \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

beschrieben. Nach Gl. (9) haben sie denselben Fluchtpunkt

$$v: x = \lim_{t \rightarrow \pm\infty} \begin{pmatrix} f \frac{Q_x + 0}{Q_z + 0} \\ f \frac{Q_y + t}{Q_z + 0} \end{pmatrix} = \begin{pmatrix} f \frac{Q_x}{Q_z} \\ \pm\infty \end{pmatrix}.$$

Das heißt, sie verlaufen auch in der Bildebene senkrecht zur x -Achse. Die übrigen Kanten sind *parallel* zur XZ -Ebene und stehen senkrecht aufeinander, haben also die $3D$ -Richtungsvektoren $(m, 0, 1)^T$ bzw. $(-1/m, 0, 1)^T$ mit $m \in \mathbf{R}$. Daher gibt es in der Bildebene zwei weitere Fluchtpunkte, deren Koordinaten nach Gl. (10) $(fm, 0)$ bzw. $(-f/m, 0)$ sind. Das bedeutet, sie liegen auf dem Horizont $y = 0$ und ihre Lage hängt nicht ab von der Entfernung zwischen Auge (Kamera) und Bildebene. Abbildung 4 bestätigt das.

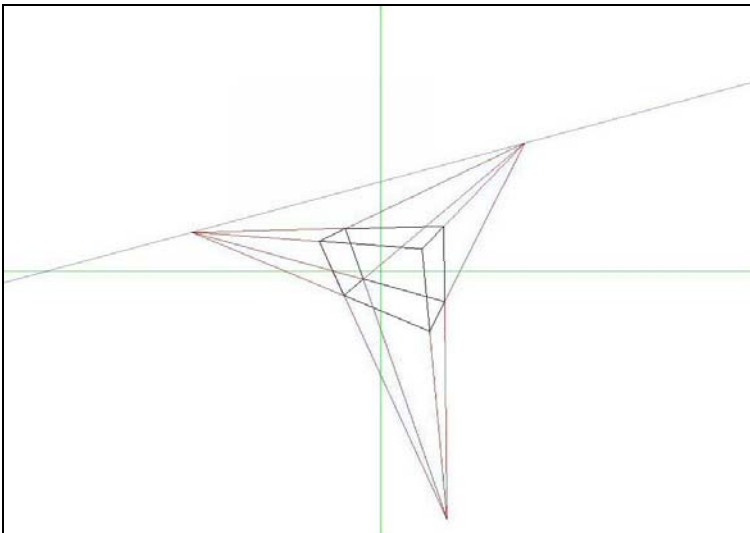


Abbildung 5 Ansicht eines Quaders mit drei Fluchtpunkten. Nimmt man an, dass die Grundfläche des Quaders auf dem waagerechten Boden aufliegt, ist die Gerade durch die beiden Fluchtpunkte der horizontalen Quaderkanten der Horizont (blaue Gerade).

Dreht man den Quader um alle drei Raumachsen, erhält man eine Projektion mit 3 Fluchtpunkten. Ein Beispiel zeigt Abbildung 5. Statt den Quader zu drehen, können wir annehmen, Auge bzw. Kamera zu schwenken – jeweils um den entgegengesetzten (negativen) Winkel. Dann definiert die Grundfläche des Quaders den waagrecht gedachten Boden und der Horizont ist gegenüber der dem xy -Koordinatensystem geneigt. Er verläuft dann durch die beiden Fluchtpunkte der horizontalen Quaderkanten. Das ist in Abbildung 5 angedeutet.

Das Programm sollte nicht nur Quader darstellen, sondern auch ungewöhnliche Perspektiven korrekt wiedergeben. Zum Beispiel Straßen mit starkem Gefälle und/oder großer Steigung. Das lässt sich sehr schön an Straßenansichten von *San Francisco* testen. Abbildung 6 zeigt ein Beispiel: Den Blick entlang der *Greenwich St.*, vom *Pioneer Park* auf dem *Telegraf Hill* (*Coit-Tower*) hinüber zum *Russian Hill*.



Abbildung 6 Die Greenwich Street in San Francisco⁴ als Testobjekt für das Computerprogramm (siehe Text)

Der Pioneer Park liegt auf einer Höhe von etwa 60 m. Von hier aus geht es, mit Unterbrechungen durch die Querstraßen *Grant* und *Stockton*, bergab zur *Powell St.* Wir befinden uns jetzt auf einer Höhe von ca. 20 m. Auf diesem Niveau bleibt die *Greenwich St.* bis zur *Taylor St.* (und kreuzt dabei die *Columbus Avenue*). Danach steigt sie wieder an bis zur *Hyde St.*, die hier am *Lombard St. Reservoir* vorbeiführt. Der Wasserspeicher liegt an der *Leavenworth St.* auf einer Höhe von etwa 100 m – und damit ungefähr 40 m oberhalb des Kamerastandortes.



Abbildung 7 Vom Programm berechneter und auf dem Bildschirm ausgegebener Verlauf der Greenwich Street. Eingegeben wurde das Höhenprofil der Straße (siehe Text). Die Dachhöhen- und Bodenlinien der Häuser im Vordergrund wurden bis zu ihrem Fluchtpunkt verlängert (rote Linien). Der Horizont ist die grüne waagerechte Gerade.

Dieses Höhenprofil erhält das Programm als Eingabedaten. Ausgeben sollte es die Ränder der *Greenwich St.* und, bis zur Kreuzung mit der *Grant St.*, auch eine Andeutung der Dachhöhen der Häuser. Abbildung 7 zeigt den vom Computer berechneten Straßenverlauf. Die Dachhöhen- und Bodenlinien der Häuser im Vordergrund wurden bis zu ihrem Fluchtpunkt verlängert (rote Linien). Der liegt erwartungsgemäß sehr weit unten im Bild (Andere Fluchtpunkte sollte das Programm der

Übersichtlichkeit halber nicht einzeichnen). Der Horizont ist die grüne waagerechte Linie im oberen Viertel des Bildes. Vergrößert man das Foto, sieht man, dass er mit dem Niveau der *Leavenworth St.* zusammenfällt. Die verläuft dort in etwa 60 m Höhe - Horizont und Kamerahöhe stimmen also überein, wie es die Perspektive verlangt. Das heißt, auch diesen Test besteht der Computer.

Literatur und Anmerkungen

¹ Das hier beschriebene Koordinatensystem wird z. B. von Dr. *ID Reid* in seiner Vorlesung *Computational Geometry* verwendet. Auch die Bezeichnungen der Vektoren bzw. Punkte wurden zum Teil von dort übernommen. Quelle: www.robots.ox.ac.uk/~ian/Teaching/CompGeom .

² Das hochgestellte **T** bedeutet, wie in der Linearen Algebra üblich, im Fall einer Matrix die *transponierte* Matrix (Zeilen und Spalten vertauscht). Bei Vektoren ist das der Übergang vom *Spalten-* zum *Zeilenvektor* oder umgekehrt. Die Ausdrücke $(X, Y, Z)^T$ bzw. $(x, y)^T$ sind daher transponierte Zeilenvektoren, mithin Spaltenvektoren. In dieser Schreibweise lassen sich Spaltenvektoren in einer Textzeile unterbringen.

³ In der angelsächsischen Literatur heißt der Fluchtpunkt folgerichtig auch *vanishing point*. Ein anschauliches Beispiel für (2D-) Grenzwerte.

⁴ Das Foto zeigt, was man nur selten erlebt: Einen nebefreien und sonnigen Morgen in San Francisco (aufgenommen im Sommer 2014).