Dies ist eine Ausarbeitung bereits bekannter Prüfungsfragen, sowie möglicher Prüfungsfragen die uns nach dem Durchsehen der Folien eingefallen sind. Es sollte (bis auf Kleinigkeiten) der ganze Stoff abgedeckt sein. Dieses Dokument ist als google doc von einigen Studenten die sich im Informatikforum zusammengefunden haben vor der Prüfung am 29.01.2013 gemeinsam ausgearbeitet worden.
Weitere Verbesserungen sind ausdrücklich erwünscht!

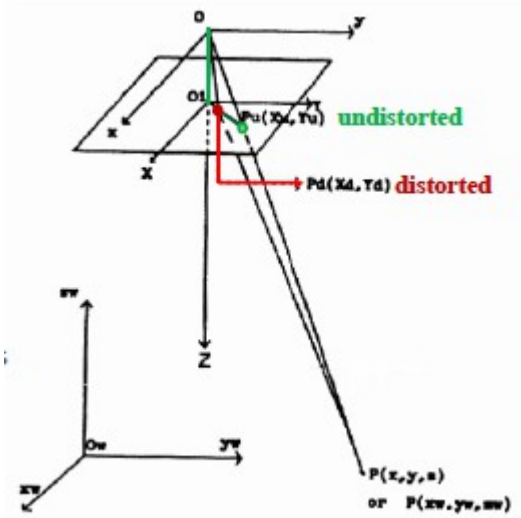
lg
R2D2

Inhaltsverzeichnis

Theoriefragen

Innere Kameraparameter

Aufzählung der inneren Kameraparameter mit Skizze. Was bewirken sie?



Definieren Pixelkoordinaten der Bildpunkte relativ zu den Koordinaten im camera reference frame. (Kamerakoordinaten)
5 oder 6 innere KameraParameter:
Fokale Länge **f** = Abstand Bildebene – Projektionszentrum
Verzerrungskoeffizient **k** (k1, k2) = Linsenverzerrung
Skalierungsfaktor **s** = Abtastfaktor in x- Richtung (Kamera AD Wandler)
Bildhauptpunkt (**Cx**, **Cy**) = Schnittpunkt optische Achse mit Bildebene

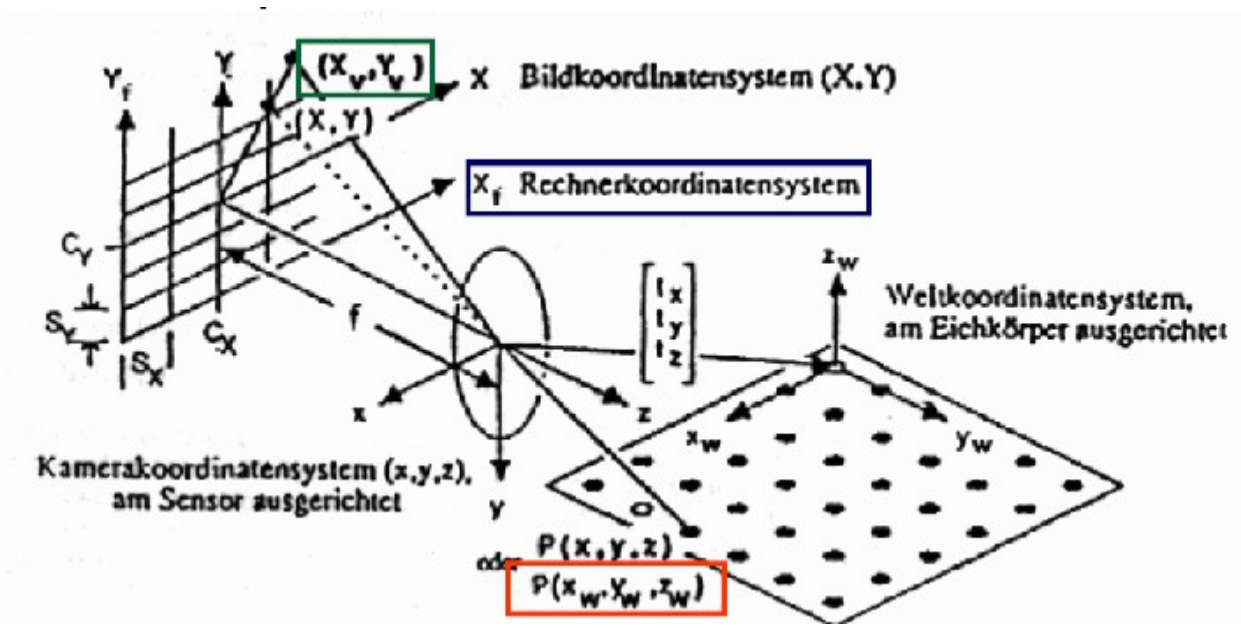
äußere Kameraparameter

Aufzählung der äußeren Kameraparameter mit Skizze

Extrinistische Parameter beschreiben Ort und Orientierung des camera reference frame (Kamerakoordinaten) zum world frame (Weltkoordinaten).

- 3 Eulersche Winkel: yaw, pitch, tilt
- 3 Translationsvektorkomponenten (Verschiebungsvektor Objektkoordinatensystem durch Projektionszentrum)

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \quad R \equiv \begin{bmatrix} r_1 & r_2 & r_3 \\ r_2 & r_5 & r_6 \\ r_3 & r_8 & r_9 \end{bmatrix}, \qquad T \equiv [T_x \ T_y \ T_z]^T$$



Dünnes Linsenprinzip

mit Skizze und Formel für Linsengleichung

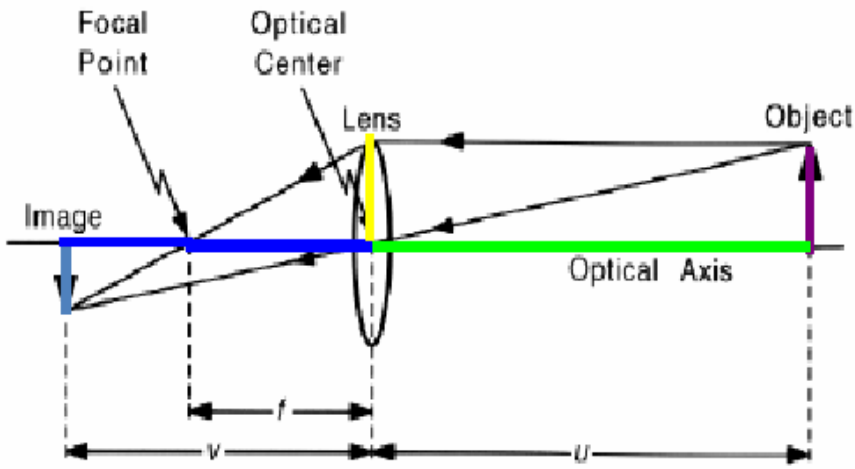
wenn u = ∞ ist v = f

Strahlen durchs Optische Zentrum werden nicht gebrochen.

- Linsengesetz:

$$\frac{y_0}{y_i} = \frac{u}{v}$$

$$\frac{y_0}{y_i} = \frac{f}{v-f}$$



Thin-Lens Equation: $\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$

Gesichtsfeld: ist jener Bereich, der von einer Kamera aufgenommen werden kann.

Je größer f desto kleiner der Ausschnitt der abgebildet wird. (Weitwinkel: kurze f, Zoom: lange f)

Focal Point = Punkte an dem alle Punkte eine Objektes in unendlicher Entfernung zusammenfallen. (= alle Strahlen parallel zur opt. Achse)

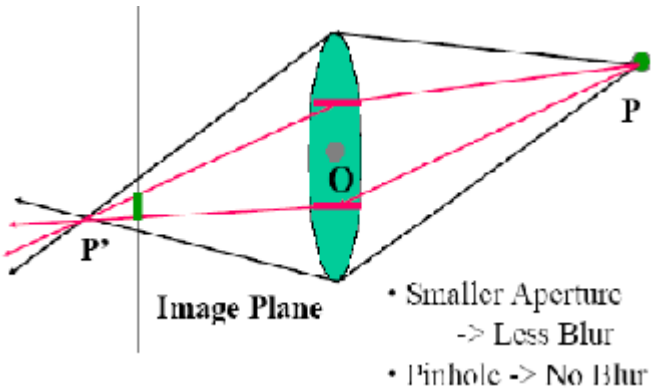
fokale Länge: Abstand von Focal Point zur Achse(Grafik: in gelb) durchs Optisches Zentrum (Linsenebene)

front/back focal length ab Linsenkrümmung

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$$

Depth of Field / Tiefenschärfebereich

Nur Objekte eines gewissen Abstandes sind scharf auf der Bildebene abgebildet, alle anderen unscharf als Kreise.



Apertur

Öffnung der Linse (Blende, Dicke und Krümmung der Linse)
Je größer die Apertur desto größer die unscharfen Kreise außerhalb des Tiefenschärfebereichs. (→ geringer Tiefenschärfebereich)
Je kleiner die Apertur, desto schärfer das Gesamtbild (aber weniger Licht kommt durch). => Je mehr Tiefenschärfe desto weniger Lichteinfall.
Apertur gemessen in f-stop (Änderung in f-stops = entweder Verdopplung oder Halbierung des einfallenden Lichtes.)

- Lower f-stop, more light
- Higher f-stop, less light

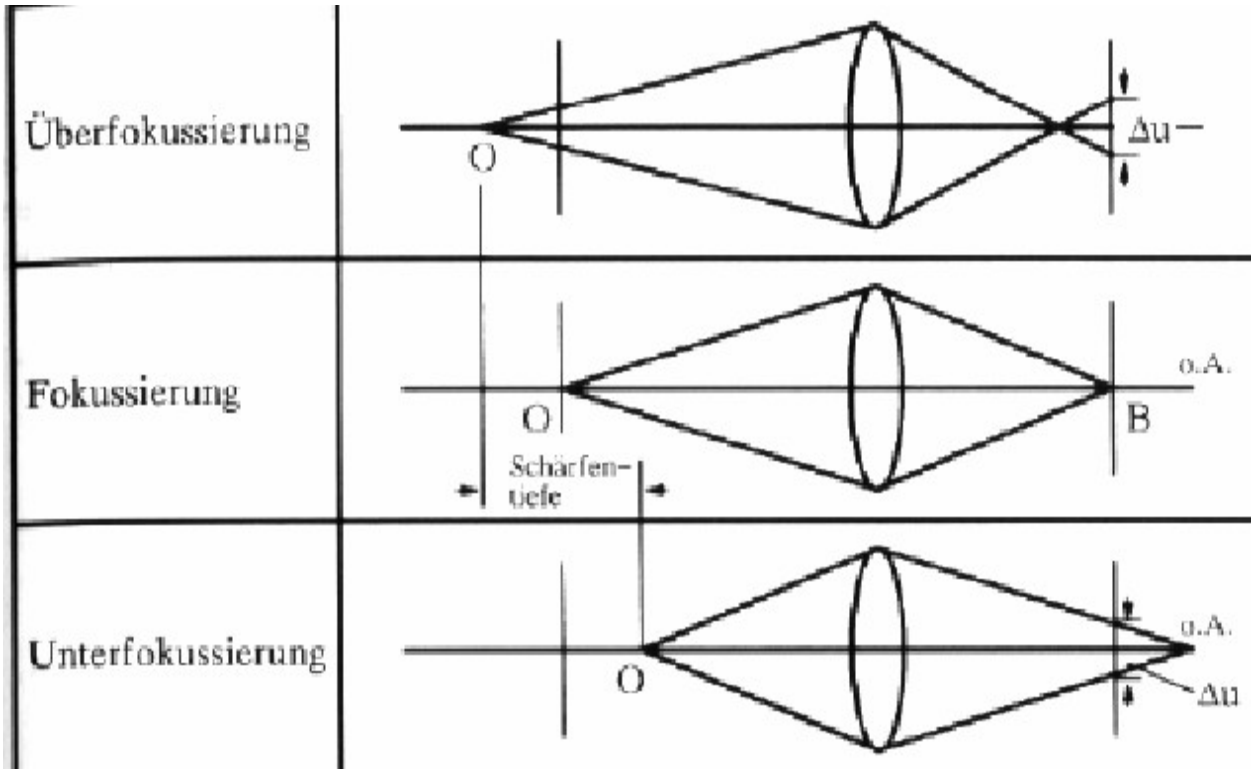
Blendenzahl k = f/D (D = Linsenhöhe)

Zusammenhang fokale Länge

raus zoomen (=fokale Länge verkürzen) erhöht Tiefenschärfebereich.
Rein zoomen (=fokale Länge vergrößern) verringert Tiefenschärfebereich. (nur mehr ein kleiner Teil im Bild ist scharf.)

Zusammenhang mit Auflösung bei Digitalkameras

Der Tiefenschärfeberich ist durch die Pixelgröße gegeben → Solange Delta U kleiner ist als ein Pixel ist der Strahl scharf → Tiefenschärfebereich wird kleiner bei höherer Auflösung.



Stereo Vision

Erklärung der Begriffe "epipolare Linien", "Epipole" & "Disparität"

Baseline: line joining the camera centers

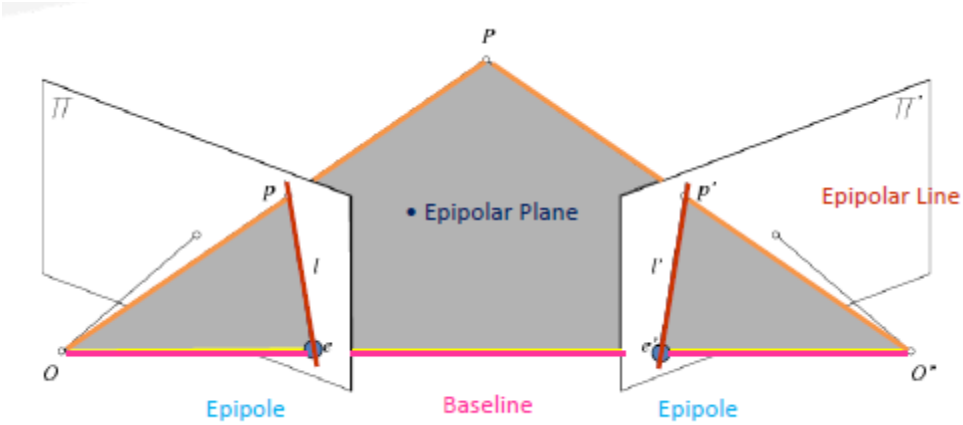
Epipole: point of intersection of baseline with the image plane

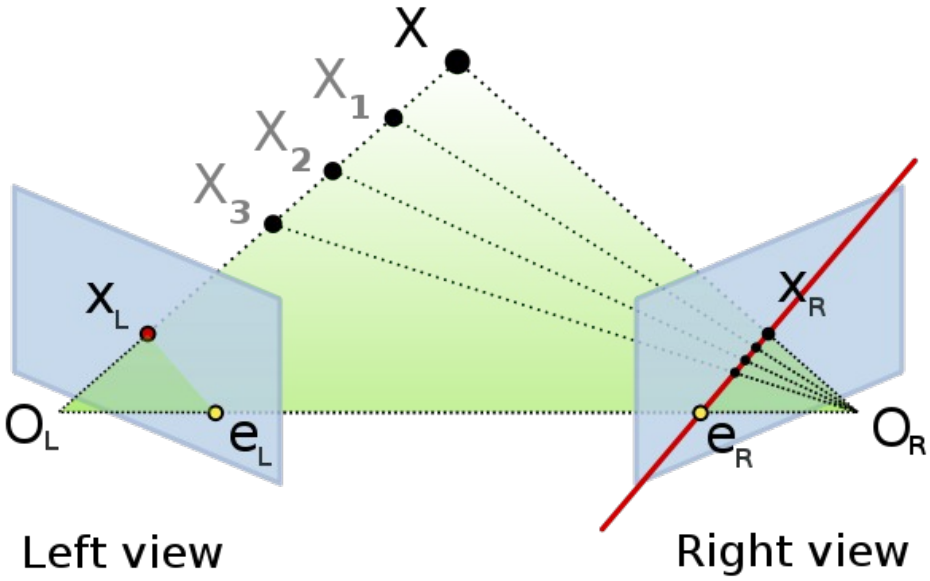
Epipolar plane: plane containing baseline and world point

Epipolar line: intersection of epipolar plane with the image plane
All epipolar lines intersect at the epipole.
An epipolar plane intersects the left and right image planes in epipolar lines.

Disparität: der Versatz (engl. Offset) in der Position, den das gleiche Objekt in der Abbildung auf zwei unterschiedlichen Bildebenen einnimmt. Die zu den Bildebenen gehörenden Brennpunkte sind dabei durch die Basis *b* räumlich voneinander getrennt. Haben beide Linsen die Brennweite *f* gilt für den Abstand *r*: $r = b \cdot f \div d$, wobei *d* für die Disparität steht. Man kann also den Abstand *r* zu einem Objekt durch eine Messung der Disparitäten im Stereobild ermitteln. Eine Disparitätenkarte eines Stereobildes ist somit gleichbedeutend zu einem Tiefenbild.

Je näher das Objekt zur Bildebene liegt desto größer ist die Disparität -> d indirekt proportional zur Distanz





Zusatzinfo:

Epipol:
(Epipol = Das Projektionszentrum der einen Kamera projiziert in die Bildebene der andere Kamera. s.WIKI)
Epipole = Schnittpunkte der Baseline (Verbindung der beiden Projektionszentren) mit den Bildebenen.

epipoleare Linen:

- If the projection point **x**L is known, then the epipolar line **e**R–**x**R is known and the point **X** projects into the right image, on a point **x**R which must lie on this particular epipolar line. This means that for each point observed in one image the same point must be observed in the other image on a known epipolar line. This provides an *epipolar constraint* which corresponding image points must satisfy and it means that it is possible to test if two points really correspond to the same 3D point. Epipolar constraints can also be described by the essential matrix or the fundamental matrix between the two cameras.
- If the points **x**L and **x**R are known, their projection lines are also known. If the two image points correspond to the same 3D point **X** the projection lines must intersect precisely at **X**. This means that **X** can be calculated from the coordinates of the two image points, a process called *triangulation*.

This is useful because it reduces the correspondence problem to a 1D search along an epipolar line.

oder anders ausgedrückt:

Liegen Punkte X1 bis Xn auf einer Linie zwischen X und dem Projektionszentrum einer Kamera (=> werden auf einen Punkt X_L in der Bildebene abgebildet), so liegen diese Punkte auch auf einer Linie zwischen dem projizierten Punkt X_R und dem Epipol in der Bildebene der anderen Kamera.

Anzahl der Zeilen im Bild = Anzahl der Epipolarlininen.(Nur wenn die optischen Achsen parallel sind)

Was versteht man unter rektifizierten Bildern?

Wenn die optischen Achsen der beiden Kameras nicht parallel sind kann man die Bilder auf eine gemeinsame Hilfsebene projezieren. Mit diesen rektifizierten Bildern kann man rechnen als wären sie mit Kameras mit parallelen optischen Achsen aufgenommen worden. Laut meiner Mitschrift wurde in der VO gesagt dass das rechnen mit rektifizierten Bildern (trotz Interpolation beim transformieren) genauer ist als das Rechnen mit der Essentiellen Matrix.

Essentielle Matrix = Projektion von Punkten im linken Bild ins rechte Bild. X' = RX +T (Rotation und Translation um ein Koordinatensystem ins andere überzuführen.)
Stereo-Rekonstruktion funktioniert nur wenn genügend Merkmale im Bild. Ein 3D-Scanner ist normalerweise besser.

Lösungen für das Korrespondenzproblem?

Ansatz: Ähnlichkeitsfunktion die Helligkeitswerte vergleichen.
Entlang Epipolarlinie Helligkeitswerte allein vergleichen würde nicht funktionieren, weil zu viele Kandidaten. Darum Vergleich zweier kleiner Fenster. Wenn Werte in beiden Fenstern sehr ähnlich -> passt! Wähle jene Fensterfunktion mit minimalem quadratischen Fehler.
Problem: welche Fenstergröße?
Wenn zu kleine (3x3): Rauschen zerstört Ergebnis.
Wenn zu groß: Details werden weg gemittelt => keine große Genauigkeit
Lösung um das zu kombinieren: Bildpyramiden!
Die maximale Anzahl an Tiefenwerten ist abhängig vom Basisabstand (je größer umso mehr Tiefenwerte; wenn Basis = 5cm und das 100 Pixel entspricht => 100 Tiefenwerte).
Der Abstand der beiden Kameras (und damit der Basisabstand) sollte auch nicht zu groß sein, weil es dann zu mehr Abschattungen kommt.

<u>Ordering Constraint</u>
Reihenfolge der Punkte nicht bekannt.
Nur mit 2 Bildern nicht lösbar.
mit SIFT-Features bestimmen (begrenzt möglich)
Lösen durch Kamerabewegung -> um zu sehen was vorne und was hinten.
Problem tritt auch bei sich wiederholenden Mustern auf

SIFT

wurde eigentlich für Stereo entwickelt, um korrespondierende Punkte zu finden bei Bildern von unkalibrierten Kameras.

Wie funktioniert SIFT?

Stage 1: Scale-space extrema Detection

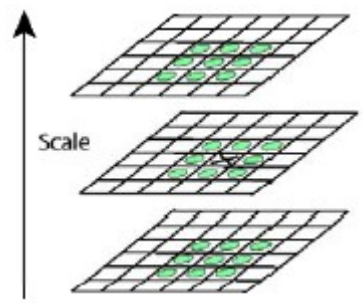
For scale invariance, search for stable features across all possible scales using a continuous g function of scale, scale space.
SIFT uses Difference of Gaussian (DoG) filter for scale space because it is efficient and as stable as scale-normalized Laplacian of Gaussian.

In computer vision, **Difference of Gaussians** is a grayscale image enhancement algorithm that involves the subtraction of one blurred version of an original grayscale image from another, less blurred version of the original. The blurred images are obtained by convolving the original grayscale image with Gaussian kernels having differing standard deviations. Blurring an image using a Gaussian kernel suppresses only high-frequency spatial information. Subtracting one image from the other preserves spatial information that lies between the range of frequencies that are preserved in the two blurred images. Thus, the difference of Gaussians is a band-pass filter that discards all but a handful of spatial frequencies that are present in the original grayscale image.

-> find extrema in 3D DoG space

Stage 2: Keypoint Localization

find keypoints in 3D SIFT scale space!
X is selected if it is larger or smaller than all 26 neighbors.
Reject points with low contrast and poorly localized along an edge data.
Using the available pixel data, subpixel values are generated by the Taylor expansion of the image around the approximate key point.



Eliminating Edge Responses

For poorly defined peaks in the DoG function, the principal curvature across the edge would be much larger than the principal curvature along it. Finding these principal curvatures amounts to solving for the eigenvalues of the second-order Hessian matrix, **H**:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

The eigenvalues of **H** are proportional to the principal curvatures of D. It turns out that the ratio of the two eigenvalues, say α is the larger one, and β the smaller one, with ratio $r = \alpha / \beta$, is sufficient for SIFT's purposes. The trace of **H**, i.e., $D_{xx} + D_{yy}$, gives us the sum of the two eigenvalues, while its determinant, i.e., $D_{xx} D_{yy} - D_{xy}^2$, yields the product. The ratio $\mathrm{R} = \mathrm{Tr}\,(\mathbf{H})^2 \,/\, \mathrm{Det}\,(\mathbf{H})$ can be shown to be equal to $(r+1)^2 / r$, which depends only on the ratio of the eigenvalues rather than their individual values. R is minimum when the eigenvalues are equal to each other. Therefore the higher the absolute difference between the two eigenvalues, which is equivalent to a higher absolute difference between the two principal curvatures of D, the higher the value of R. It follows that, for some threshold eigenvalue ratio r_{th}, if R for a candidate keypoint is larger than $(r_{\mathrm{th}} + 1)^2 / r_{\mathrm{th}}$, that keypoint is poorly localized and hence rejected.

Keypoint Detector: apply peak value threshold and test ratio of principle curvatures

Stage 3: Orientation assignment

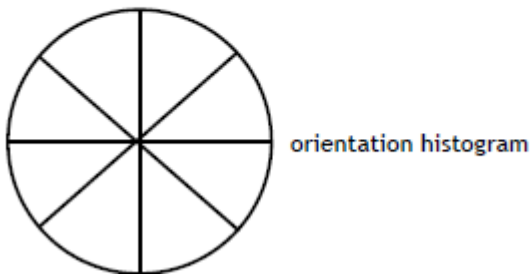
By assigning a consistent orientation, the keypoint descriptor can be orientation invariant.

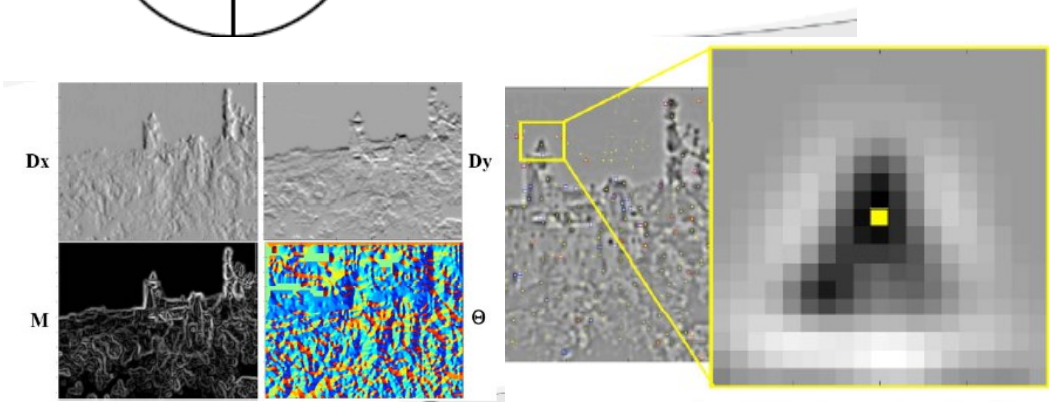
Assign dominant orientation as the orientation of the keypoint.

First, the Gaussian-smoothed image $L(x, y, \sigma)$ at the keypoint's scale σ is taken so that all computations are performed in a scale-invariant manner. For an image sample $L(x, y)$ at scale σ, the gradient magnitude, $m(x, y)$, and orientation, $\theta(x, y)$, are precomputed using pixel differences:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$



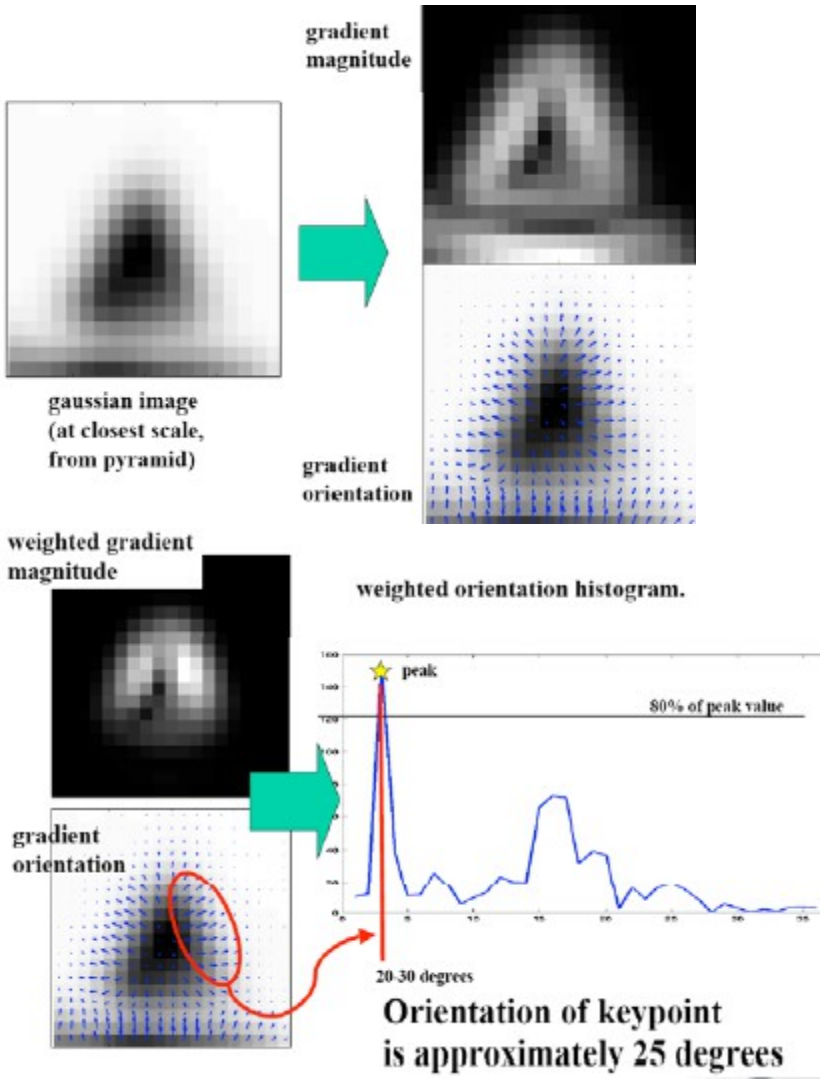


keypoint location = extrema location

keypoint scale is scale of DoG image

gaussian image -> compute gradient magnitude and gradient orientation

gradient magnitude weighted by 2D gaussian kernel



combine weighted gradient magnitude and orientation and apply a threshold for the peak value of the weighted orientation histogram.

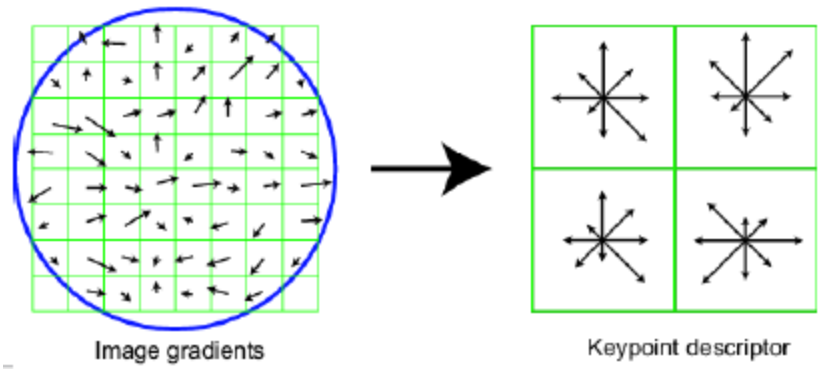
An orientation histogram with 36 bins is formed, with each bin covering 10 degrees. Each sample in the neighboring window added to a histogram bin is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the keypoint. The peaks in this histogram correspond to dominant orientations. Once the histogram is filled, the orientations corresponding to the highest peak and local peaks that are within 80% of the highest peaks are assigned to the keypoint. In the case of multiple orientations being assigned, an additional keypoint is created having the same location and scale as the original keypoint for each additional orientation.

About 15% has multiple orientations

Stage 4: Keypoint Descriptor

This step is performed on the image closest in scale to the keypoint's scale.

- Image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms on 4x4 pixel neighborhoods with 8 bins each
  (These histograms are computed from magnitude and orientation values of samples in a 16 x 16 region around the keypoint such that each histogram contains samples from a 4 x 4 subregion of the original neighborhood region. The magnitudes are further weighted by a Gaussian function with σ equal to one half the width of the descriptor window)
- 4 x 4 = 16 histograms each with 8 bins = 128 dimensions descriptor vector.
- Normalized, clip the components larger than 0.2
  (Normalizing enhances invariance to affine changes in illumination. To reduce the effects of non-linear illumination a threshold of 0.2 is applied and the vector is again normalized.)



finally Match features:
Nearest neighbor, Hough voting, Least-square affine parameter fit

Eigenschaften von Sift

rotations-, skalierungs- und helligkeits-unabhängig

Wie werden die Features Orientierungsinvariant gemacht?

siehe Orientation assignment: each keypoint is assigned one or more orientations based on local image gradient directions. This is the key step in achieving invariance to rotation as the keypoint descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation.

Single Choice Fragen zu SIFT

<u>6 Statements zu SIFT (correct/incorrect ankreuzen, falsche Antwort -1 Punkt!)</u>

- Der Bereich 0 bis 360 Grad wird in 128 Bins geteilt.
  Bei der *Orientierung* wird der Bereich von 0 bis 360 Grad in 36 Bins zu je 10 Grad geteilt.
  (*Deskriptor*: pro 4x4-Sample-Kästchen in 8 Bins, also Schritte zu 45 Grad. 8 Bins mal 16 Kästchen gibt dann einen Deskriptor der Länge 128.)
- unterschiedliche Belichtung --> Skalierung auf Einheitslänge
  Ja, SIFT ist beleuchtungsinvariant. Die Gradienten (Helligkeitsänderungen) werden auf 1 normalisiert, dann auf [0, 0.2] geclampt und nochmal normalisiert. Damit haben hohe Gradienten nicht so viel Einfluss.
- rotationsinvariant
  ja
- scale-invariant
  ja
- Difference-of-Gaussian-Filter wird verwendet.
  ja
- Beim Matching werden nur Features mit gleichem Scale verglichen.
  nein

<u>SIFT berechnet (ja/nein)</u>

- skalierung der Keypoints JA
- histogram des DoG scale space NEIN -> berrechnet weighted orientation histogram (weighted gradient magnitude and orientation)

hauptorientierung der keypoints JA (peak des weighted orientation histogram)

8-bin histogramm der gradientenorientierung in 16 zellen um die keypoints JA
(*Deskriptor*: pro 4x4-Sample-Kästchen in 8 Bins, also Schritte zu 45 Grad. 8 Bins mal 16 Kästchen gibt dann einen Deskriptor der Länge 128.)

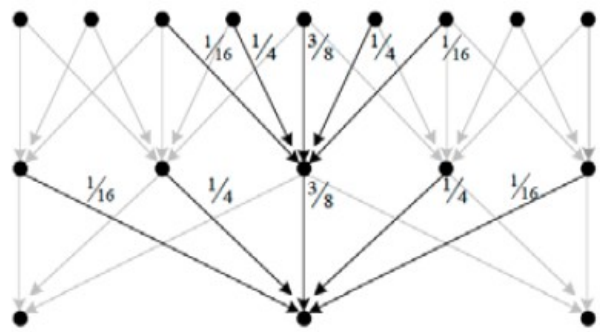
grauwerthistogram der lokalen nachbarschaft der Keypoints NEIN

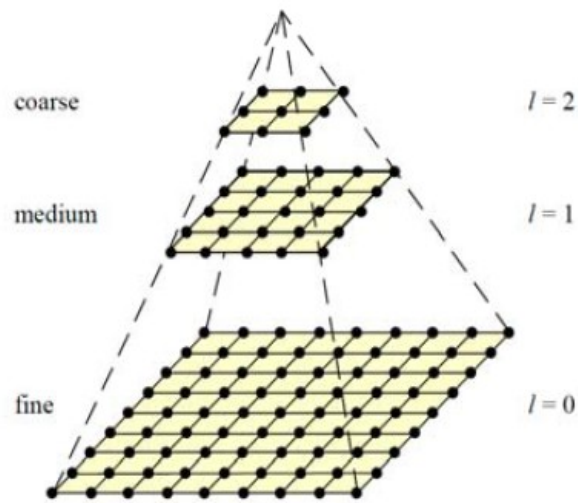
- (X,Y) -Koordinaten der Keypoints JA

farbhistogramm der lokalen nachbarschaft der keypoints NEIN

Gaussian Pyramide

Smooth with gaussians, because a Gaussian*Gaussian = another Gaussian
Gaussians are low pass filters, so representation is redundant.
Jedes Level ist ½ Länge und ½ Breite des vorhergehenden.
A **Gaussian pyramid** is a technique used in image processing, especially in texture synthesis. The technique involves creating a series of images which are weighted down using a Gaussian average (Gaussian blur) and scaled down. When this technique is used multiple times, it creates a stack of successively smaller images, with each pixel containing a local average that corresponds to a pixel neighborhood on a lower level of the pyramid.





Why Pyramid?

Keep filter same size
Change image size
scale factor of 2
Compression: Capture important structures with fewer bytes
Denoising: Model statistics of pyramid sub-bands
Image blending

Some desirable Properties for a Blur Kernel

Positivity: h(m) >= 0
Symmetry: h(m) = h(-m)
Unimodality: h(m) >= h(m+1) for m >= 0
Normalized: h(m) = 1
Equal contribution: h(2m) = h(2m+1)

used for

up- or down- sampling images.
Multi-resolution image analysis

- Look for an object over various spatial scales
- Coarse-to-fine image processing: form blur estimate or the motion analysis on very low-resolution image, upsample and repeat.
- Often a successful strategy for avoiding local minima in complicated estimation tasks.

Laplacian of Gaussian

LoG ist Fxx + Fyy, also die Summe von der 2ten Ableitung in x-richtung und der 2ten Ableitung in y-richtung von der Gauss-Funktion. schaut dann eben aus wie in umgekehrter sombrero, darum auch 'mexican hat'.

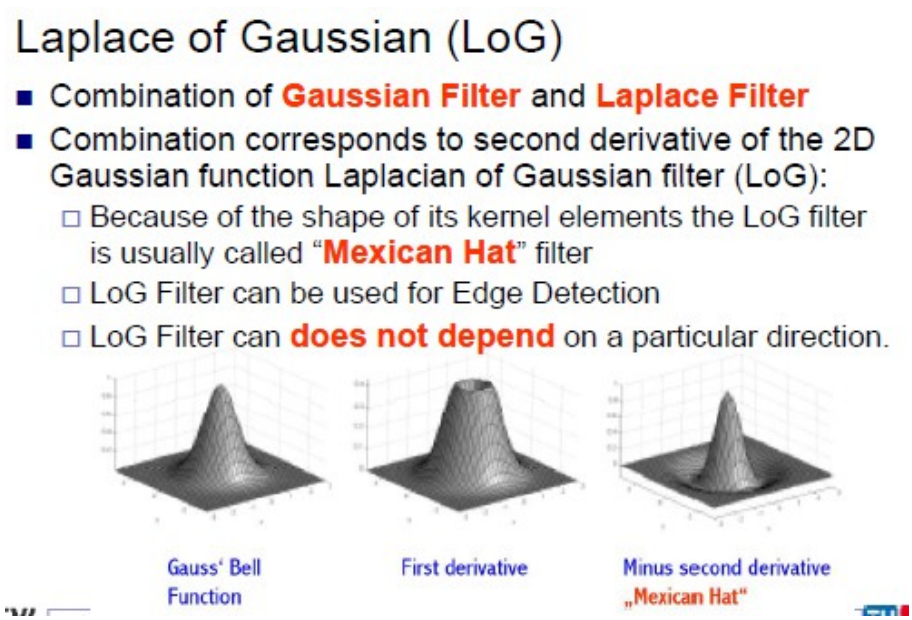
1) Input image is convolved by a Gaussian kernel at a certain scale t to give a scale space representation.

2) Then the Laplacian operator $\nabla^2 L = L_{xx} + L_{yy}$ is computed, which usually results in strong positive responses for dark blobs of extent $\sqrt{2t}$ and strong negative responses for bright blobs of similar size.

A main problem when applying this operator at a single scale, however, is that the operator response is strongly dependent on the relationship between the size of the blob structures in the image domain and the size of the Gaussian kernel used for pre-smoothing. In order to automatically capture blobs of different (unknown) size in the image domain, a multi-scale approach is therefore necessary. -> *scale-normalized Laplacian operator*

$$\nabla^2_{norm} L(x, y; t) = t(L_{xx} + L_{yy})$$

3) for Blob detection: search in scale space for maxima/minima (like Harris)

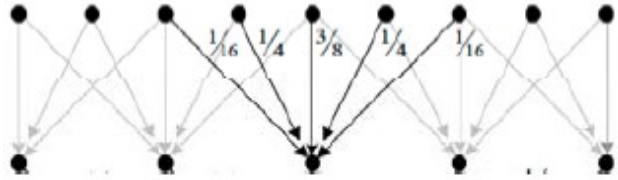


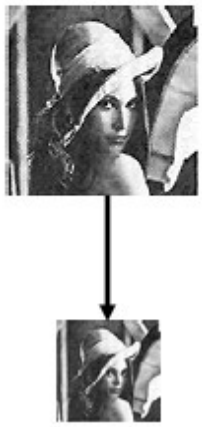
Laplace Pyramide

verwendet DoG (Difference of Gaussian) filter = Difference between Upsampled Gaussian pyramid level and Gaussian pyramid level.

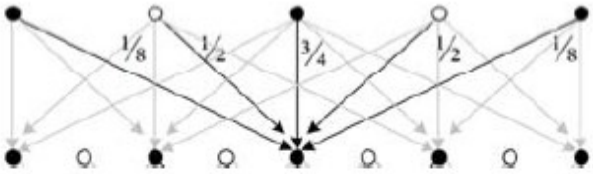
-> band pass filter - each level represents spatial frequencies (largely) unrepresented at other level.

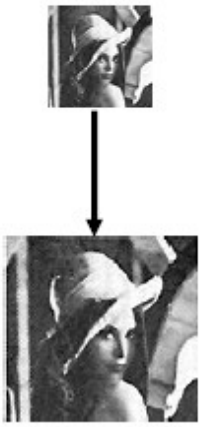
Reduce

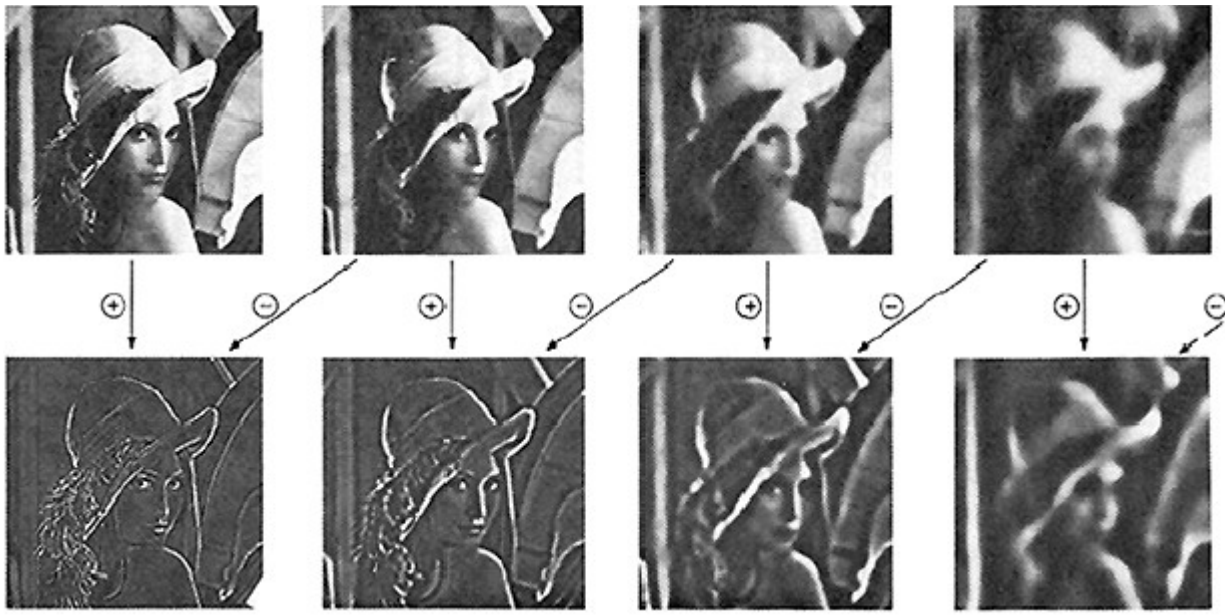




Expand

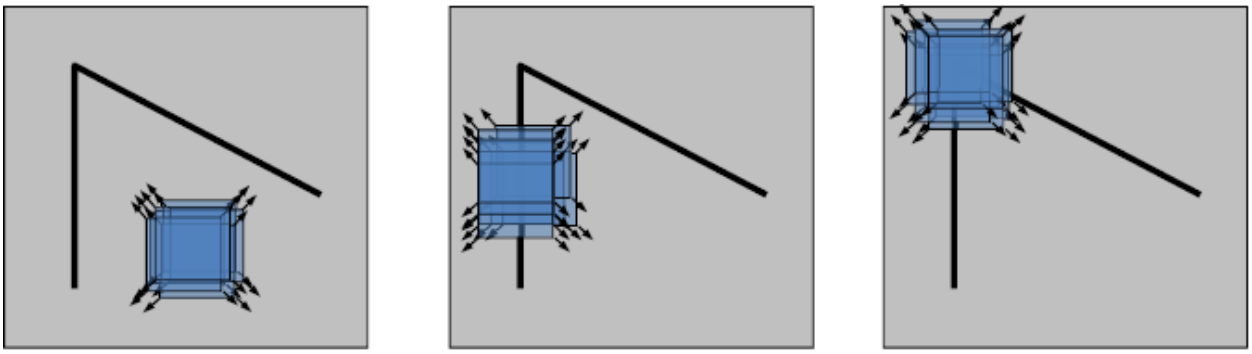






Harris Corner Detection

Basic idea:

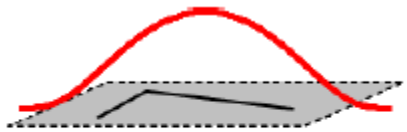


"flat" region:
no change as shift window in all directions

"edge":
no change as shift window along the edge direction

"corner":
significant change as shift window in all directions

noisy response due to a binary window function => use a Gaussian function



Gaussian

- ist rotations invariant (weil eigenwerte gleich bleiben)
- Partial invariance to affine intensity change (invariant zu intensity shift weil nur ableitungen verwendet werden)
- invariant zu intensity scale? ich würde sagen das ist abhängig vom threshold.

Man berechnet die Ableitungen des Bildes in beide Richtungen I_x und I_y um dann für jedes Pixel die Hesse Matrix (I_x^2, I_{xy}; I_{xy}, I_y^2) auszuwerten.

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

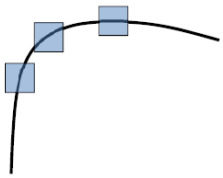
I_x^2 und I_y^2 stellen die Kanten in x- und y Richtung dar, I_{xy} zeigt wo Kanten aufeinander treffen (also wo Ecken sind). Die Resultierenden Matrizen / Bilder werden mit einer Gauss-Funktion gefiltert um durch diese Fensterfunktion Rauschen zu unterdrücken.
Aus diesen Matrizen berechnet man dann für jedes Pixel ein 'cornerness' Maß mit der Formel

R = det(A) - alpha * trace(A)^2, wobei A jeweils die Hesse-Matrix mit den gefilterten Ableitungen ist.
Auf R wendet man non-maximum suppression mit 8 Nachbarn und einem passenden minimum-threshold an und das Ergebnis sind die gefundenen Kanten.

3 Bilder (homogener Bereich, schräge Kante, Ecke) sind gegeben. Nach der Summe der Eigenwerte reihen!

Homogener Bereich: beide Eigenwerte sehr nahe bei 0
Kante: Ein Eigenwert sehr nah bei 0 der andere nimmt einen deutlich höheren Wert an
Ecke: Beide Eigenwerte sehr hoch, da Richtungsänderung in zwei Dimensionen

Ist er scale-invariant? Begründung.



Nein. Wenn man eine Ecke skaliert wird sie möglicher Weise nur mehr als Kante klassifiziert. Siehe Grafik links. Wäre die Skalierung anders oder das Window größer würde eine Ecke erkannt werden. (siehe rechts)

Ransac (Random Sample Consensus)

Wie funktioniert Ransac?

- Man wählt (wenige) zufällige Punkte aus (z.b. nur genau so viele wie man mindestens braucht, für Homographie beispielsweise 4 Matches)
- Mit diesen Punkten wird eine Lösung (Hypothese) berechnet
- Diese Hypothese wird auf alle anderen Punkte angewandt, die Differenz zu den richtigen Werten berechnet und mit einem Threshold zwischen 'Inliers' und Ausreißern unterschieden. Um so mehr 'Inliers' die Hypothese stützen, um so besser die Lösung.
- Das ganze wird sehr oft wiederholt (z.b. 1000 Mal) und die beste Lösung verwendet. Mit dieser kann man dann noch einmal mit allen Inliern des besten Durchlaufs eine genauere Lösung berechnen.

Lösung bei Falschen Korrespondenzen?

Liefert er auch bei sehr vielen Falschen Korrespondenzen eine Lösung (mit Begründung)?
Ja, weil eine Lösung gesucht wird die von möglichst vielen Korrespondenzen unterstützt wird. Durch die Unterscheidung von Ausreißern und 'Inliers' kann eine Lösung gefunden werden die gar nicht von den falschen Korrespondenzen gestört wird.

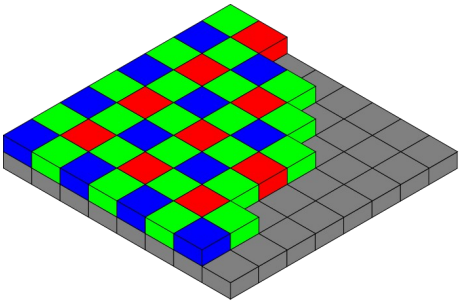
Ransac Single Choice Fragen

6 Statements zu RANSAC (correct/incorrect ankreuzen, falsche Antwort -1 Punkt!)

- Der Algorithmus ist deterministisch?
  Nein, RANSAC nimmt nur zufällig Samples und führt im Allgemeinen jedes Mal zu einem anderen Ergebnis.
- Er findet immer eine optimale Lösung?
  Nein, findet im Allgemeinen nie eine optimale Lösung (dafür müsste er alle Kombinationen mit dem Modell vergleichen).
- Es ist besser, nach allen Runs das Modell aus allen gefundenen Inliern zu bilden?
  Ja, wenn damit gemeint ist, dass alle Inlier der besten Homographie (des besten Runs) für eine neue Schätzung zu verwenden besser ist als die Homographie selbst als Ergebnis zu nehmen.
- Mindestens 50% richtige Übereinstimmungen sind erforderlich.
  Nein.
- Es wird ein Threshold definiert, um zwischen Outlier und Inlier zu unterscheiden.
  ja
- Nach allen Runs wird das Modell mit den meisten Inliern gewählt.
  ja

Demosaicing

<u>Beschreibe eine der 3 Methoden, die nach Demosaicing eingesetzt werden</u>
Demosaicing: Die meisten Digitalkameras verwenden das Bayer-Pattern zur Anordnung der Sensoren → Da ein einzelner Sensor nur eine Farbe aufnehmen kann (eigentlich sowieso nur Helligkeitswerte, es wird ein Farbfilter vor den Sensor gelegt) werden diese als Mosaic verteilt (G doppelt so oft weil G beim menschlichen Auge einen größeren Beitrag zur Helligkeits-wahrnehmung beiträgt durch die Natur bedingt → am meisten Grün, grünsensitives Sehen)



Um aus dem Bayer-Pattern ein RGB Bild zu bekommen in dem jedes Pixel RGB Werte hat muss man Demosaicing machen, z.b. mittels bilinear filtering die fehlenden Farbkanäle pro Pixel interpolieren (siehe Rechenbeispiele). Kompliziertere Methoden verwenden bestimmte Kernels und rechnen Farbräume um.

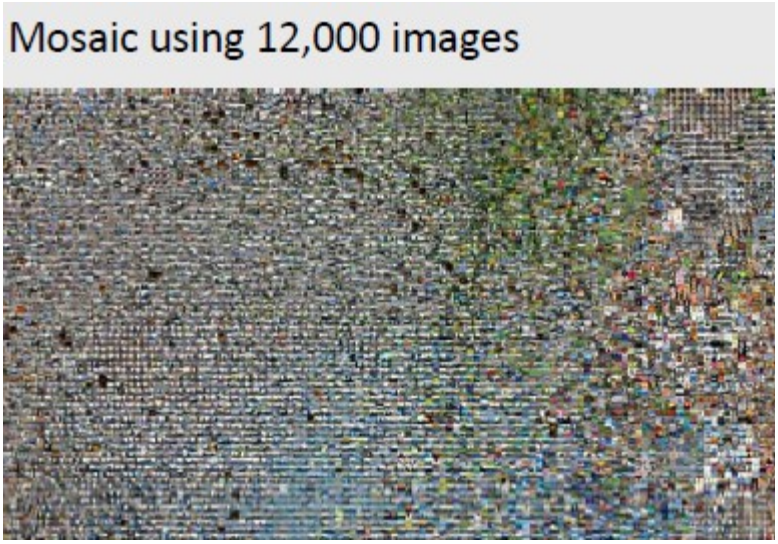
Mosaic / Panorama Images

Algorithmus für Mosaic (Panorama Bilder)

see Image stitching
(bei Mosaic-Bildern normaler Weise kein Blending notwendig, sofern keine Überlappungen.)

Wie erzeugt man ein Panorama (oder Mosaic) Image?

Indem man die Einzelbilder mit der gefundenen Homographie in ein gemeinsames Bild / Referenzkoordinatensystem projiziert (natürlich jeweils an die richtige Stelle). Bei Überlappungen (Panorama) werden die Bilder überblendet, dazu kann man verschiedene Alpha-Maps verwenden. (zb feathering) Details: siehe Image Stiching.



Probleme wenn die Kameraposition verändert wird?

Dann kann ein Bild aufs andere nicht mehr perfekt durch eine Projektion abgebildet werden.

Image Stitching

Jeweils paarweise:

- Interest-Points in beiden Bildern finden
- Korrespondenzen Bilden: die gefundenen Interest-Points in den beiden Bildern zueinander matchen
    - Falsche Korrespondenzen ausfiltern, z.b. durch RANSAC (Inlier bleiben über)
- Mit den Korrespondenzen eine Homographie von einem Bild aufs andere Bild finden.

Hat man jeweils die Homographien kann man alle Bilder in ein gemeinsames Bild projezieren und zusammenbauen (überblenden, z.b. mit Alpha-Maps, feathering)

K-means

Zielfunktion

Beim K-means clustering werden alle Pixel eines Bildes zu K clustern zusammengelegt, sodass die Abweichung innerhalb eines Clusters (distortion meassure) möglichst klein ist, also die cluster jeweils ähnliche pixel enthalten.

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r(n,k) \, \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

Zielfunktion:
= sum of the squares of the distances of each data point to its assigned cluster centroid $\boldsymbol{\mu}_k$.

Wie determiniert der Algorithmus?

Man berechnet J, ein Maß für die 'Verzerrung' (distortion meassure) der aktuellen Lösung und vergleicht sie mit dem J des vorherigen durchgangs. Wenn der Faktor, um den das neue J besser als das alte J ist unter einem definierten Threshold liegt terminiert der Algorithmus.

Welche Bedeutung hat K?

K ist die Anzahl der Cluster.

Funktionsweise

1. Choose random starting values for the centroids $\boldsymbol{\mu}_k$.
2. Assign all data points to their nearest cluster centroids.
3. Compute the new cluster centroids as the mean of all data points assigned to that cluster.
4. Compute J and check for convergence. For this purpose, compute the ratio between the old and the new J. If the ratio lies under a given threshold, terminate the clustering, otherwise go to point 2.

Vor / Nachteile von K-Means

+ Relativ effizient O(tkn), t = # of iterations, k = # clusters, n = # of datapoints
+ Findet meist zumindest ein lokales optimum. Globales Optimum kann man mit simulated annealing oder genetischen Algorithmen finden

- Nur anwendbar bei numerischen daten (Wenn man einen Mittelwert und Distanz zwischen Punkten definieren kann), nicht bei kategorischen Daten.
- Anzahl der Cluster k muss im Vorhinein festgelegt werden
- Kann nicht mit Rauschen und Ausreißern umgehen
- Kann nicht Cluster mit beliebigen Formen finden. (also schlecht für object recognition?)

Pinhole Camera Model

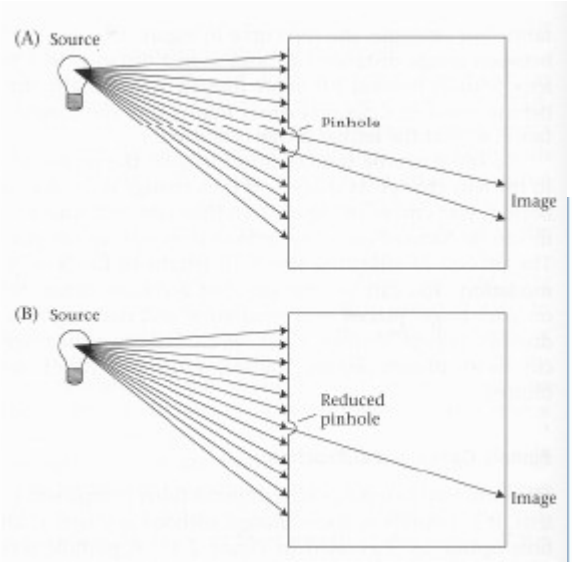
Camera Obscura. Used to observe eclipses.

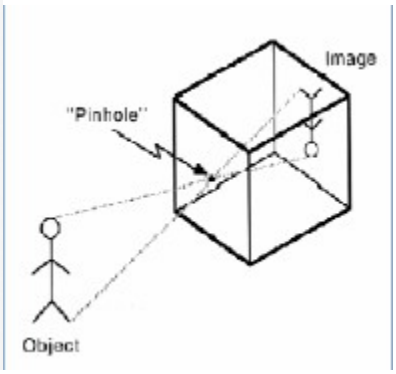
Wie entsteht ein ein image?

Hat sehr kleines Loch (=Apertur), Licht tritt durch Loch ein und formt Bild auf Rückwand (auf dem Kopf stehend)

Was ist das problem wenn das Loch zu groß ist, was wenn es zu klein ist?

zu groß: image rays are not properly converged => blurry picture
zu klein: wieder schlechterer Fokus wegen Diffraction (das haengt mit der tatsache zusammen dass das Licht sowohl Teilchen als auch welle ist --> dualitaet des lichts. Bei Wellen gibt es das Phänomen, dass nach einer kleinen Öffnung sich eben jene Wellen in alle Richtungen ausbreiten und nicht nur, so wie man annehmen könnte, in die in der die Welle durch das Loch kommt.)





Tilt Shift:

Wodurch entsteht der Effekt (dass es wie eine Miniatur aussieht)?

Durch den Tilt wird die Projektionsebene der Linse rotiert und liegt nicht mehr genau auf der Bildebene. Dadurch ist nur mehr ein kleiner Bereich im Bild, der an dem sich die Projektionsebene und die Bildebene nahe genug sind, scharf, der Rest wird unschärfer je weiter man von dem Schnittpunkt wegkommt. (So kann man den Effekt auch simulieren → Scharfen Bereich auswählen, rest mit Abhängikeit der Entfernung vom scharfen Bereich blurren).
Dass es wie eine Miniatur aussieht kommt daher, dass bei Aufnahme von nahen Objekten der scharfe Bereich des depth of field viel geringer ist.

siehe auch: http://www.northlight-images.co.uk/article_pages/tilt_and_shift_ts-e.html

Wieso wird er durch eine höhere color saturation noch verstärkt?

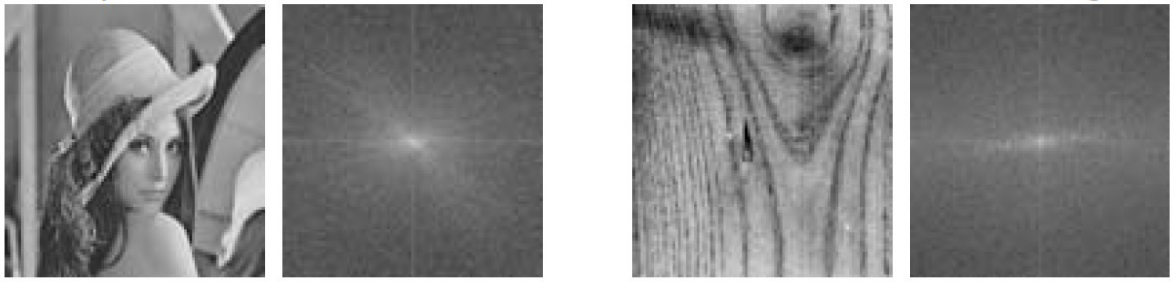
Bei echten Szenen verlieren entfernte Objekte durch die atmosphärische Perspektive (siehe monokulare depth-cues) an Kontrast und Sättigung. Das erhöhen der Sättigung kompensiert diese atmosphärischen Effekte, der depth-cue wird entfernt und es schaut aus als wäre alles nahe beisammen, das ist aber in der Realität nur so wenn alles klein ist.

Fourier Spektrum erklären

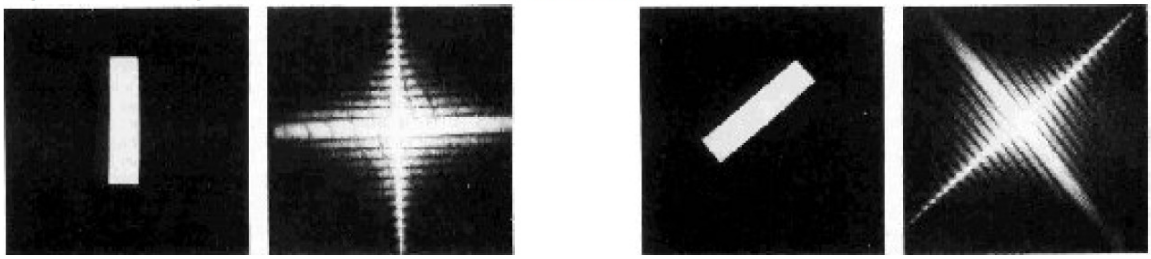
The discrete-time Fourier transform (DTFT) is one of the specific forms of Fourier analysis. As such, it transforms one function into another, which is called the frequency domain representation, or simply the "DTFT", of the original function (which is often a function in the time-domain). The DTFT requires an input function that is discrete. Such inputs are often created by digitally sampling a continuous function, like a person's voice.
The DTFT frequency-domain representation is always a periodic function. Since one period of the function contains all of the unique information, it is sometimes convenient to say that the DTFT is a transform to a "finite" frequency-domain (the length of one period), rather than to the entire real line.

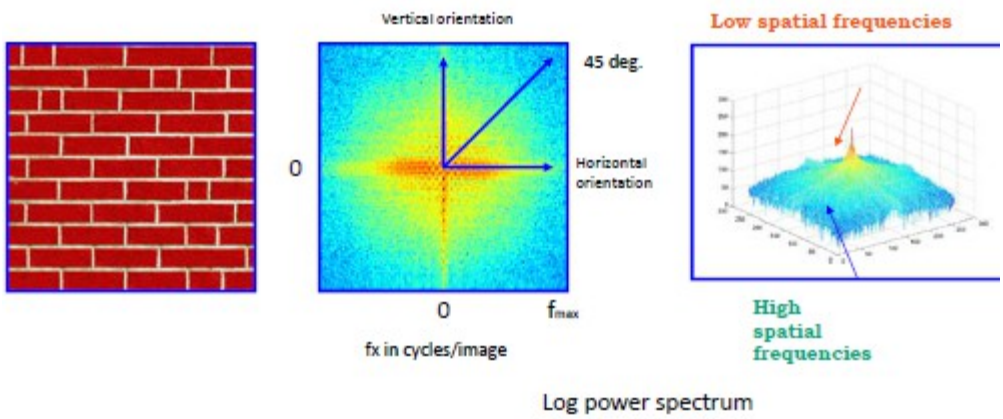
Das Fourier Spektrum eines Bildes ist nichts anderes als die Darstellung des Bildes im Frequenzbereich. Im Fourier Spektrum werden die Frequenzen, die im Bild für jede Richtung auftreten, dargestellt. Treten im Bild niedrige Frequenzen auf, so hat das Spektrum hohe Werte in der Bildmitte, hohe Werte am Rand des Spektrums weisen auf hohe Frequenzen im Bild hin. Das Fourier Spektrum zeigt auch an in welche Richtung die hohen bzw. niedrigen Frequenzen im Bild vorkommen.
Die Spektren natürlicher Bilder unterscheiden sich visuell wenig, die Frequenzen sind eher gleichmäßig verteilt, d.h. in jede Richtung können hohe und niedrige Frequenzen vorkommen. Bei synthethischen Bildern sind die Fourier Spektren sehr unterschiedlich.

Die Spektren natürlicher Bilder unterscheiden sich visuell wenig:



Spektren synthetischer Bilder sind auch visuell unterschiedlich:





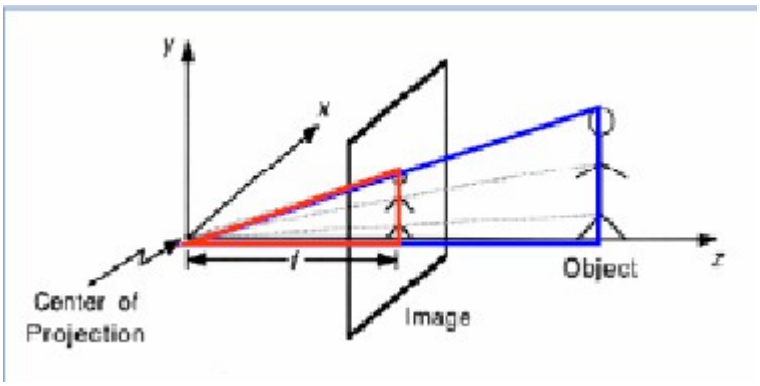
Was ist es, was ist darin enthalten?

The frequency spectrum of a time-domain signal is a representation of that signal in the frequency domain. The frequency spectrum can be generated via a Fourier transform of the signal, and the resulting values are usually presented as **amplitude and phase**, both plotted versus **frequency**. Any signal that can be represented as an amplitude that varies with time has a corresponding frequency spectrum.

Mögliche Theoriefragen (neu)

Formeln der Zentralprojektion mit Erklärung

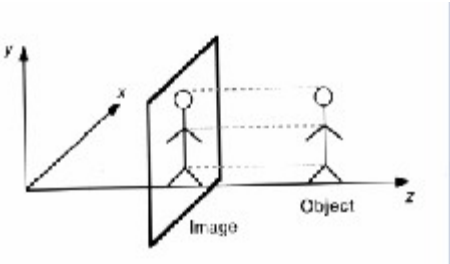
= Perspektivische Projektion (Pinhole perspective). Ist nicht linear! Bildgröße hängt von Entfernung Z ab.
Häufigste Modellierung mittels homogenen Koordinaten.
=> (X,Y,Z,W) = (wx,wy,wz,w)
Jeder Punkt des 3D Raumes wird auf eine Linie im 4D Raum, die durch den Ursprung geht abgebildet.
Berechnung der Bildkoordinaten (x,y) aus Weltkoordinaten (X, Y, Z) und fokaler Länge f:
Verhältnis y/Y = f/Z umformen => x = f/Z * X y = f/Z * Y



$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix}$$

Formeln der Normalprojektion mit Erklärung

Normalprojektion/Orthographische Projektion (wenn normal auf Bildebene) oder Parallelprojektion (wenn nicht normal).
Jeder 3D Punkt wird durch Strahlen die normal zur Bildebene sind abgebildet.
z = ∞, x = X, y = Y



weak perspective

• nur perspektivischer Effekt (perspektivische Projektion approximiert durch Normal- / Parallelprojektion)
• 3D Punkte in Gruppen ähnlicher depth, dann durch Gruppendepth dividieren (=Skalierungsfaktor s)
• (x,y,z) →s(x,y) … s konstant
• Parallele Linien bleiben parallel
→ einfache Mathematik, für kleine und nahe Objekte
→ für recognition

Was ist Radiometry?

Is the science of light and energy measurement

<u>Radiance</u>

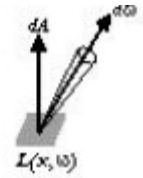
- Energy carried by a ray
- energy / (area solid angle)

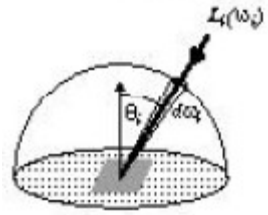
<u>Irradiance</u>

- Energy per unit area falling on a surface

<u>Radiosity</u>

- Energy per unit area leaving a surface





Merkmalsextraktion

Kanten

Edgel, Edgels (Edgel is a term commonly used in computer vision research to refer to a pixel in an image that has the characteristics of an edge, as defined in the image processing sense, and may also be used to refer to the orientation and/or magnitude data of the edge. The term originated as an abbreviation of the term "edge pixel”.)
Diskontinuitäten der Helligkeit können Diskontiniuitäten der <u>Oberflächennormale</u>, der <u>Tiefe</u>, der <u>Textur</u> und der <u>Beleuchtung</u> bedeuten.

Kantenoperatoren

• Roberts
• Prewitt
• Sobel
• Marr- Hildreth Operator (Mexican Hat) (Laplacian of Gaussian LoG)
• Canny

Glättung

Mittelwert, Rauschunterdrückung, hohe Bildfrequenzen werden eliminiert (Tiefpassfilter).

Gabor Wavelets

(Quelle: http://en.wikipedia.org/wiki/Gabor_wavelet#Wavelet_space)

Gabor filter

Gabor filter is named after Dennis Gabor. Frequency and orientation representations of Gabor filters are similar to those of the human visual system.

linear filter for edge detection

= **Gaussian kernel modulated by sinusoidal plane wave**

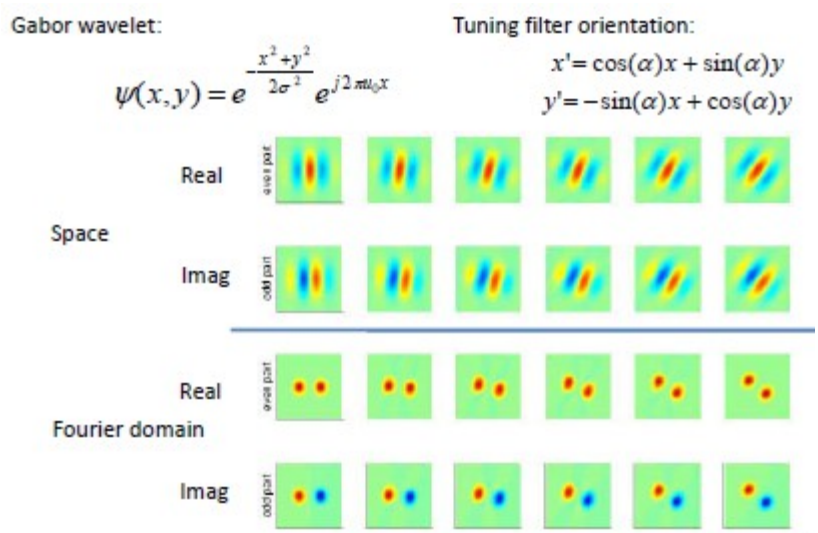
self similar

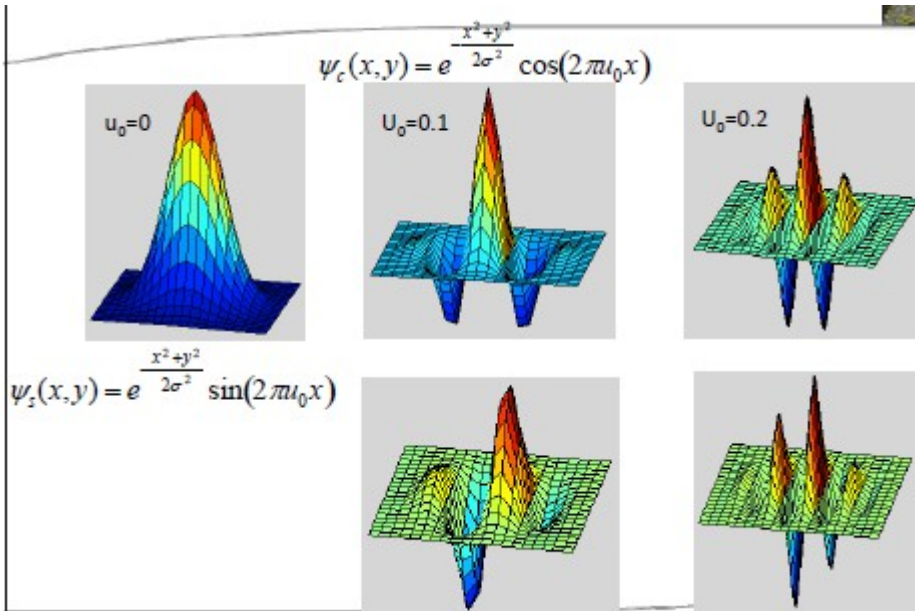
harmonic function multiplied by Gaussian function

has a real and imaginary component representing orthogonal directions (may be used individually)

Gabor wave

A wavelet is a wave-like oscillation with an amplitude that starts out at zero(0), increases, and then decreases back to zero. It can typically be visualized as a "brief oscillation" like one might see recorded by a seismograph or heart monitor. Generally, wavelets are purposefully crafted to have specific properties that make them useful for signal processing. Wavelets can be combined, using a "reverse, shift, multiply and sum" technique called convolution, with portions of an unknown signal to extract information from the unknown signal.





Difference of Gaussian erklären

SIFT verwendet DoG

From the fact that the scale space representation $L(x, y, t)$ satisfies the diffusion equation

$$\partial_t L = \frac{1}{2}\nabla^2 L$$

it follows that the Laplacian of the Gaussian operator $\nabla^2 L(x, y, t)$ can also be computed as the limit case of the difference between two Gaussian smoothed images (scale space representations)

$$\nabla^2_{norm} L(x, y; t) \approx \frac{t}{\Delta t}\left(L(x, y; t + \Delta t) - L(x, y; t - \Delta t)\right).$$

In the computer vision literature, this approach is referred to as the Difference of Gaussians (DoG) approach. Besides minor technicalities, however, this operator is in essence similar to the Laplacian and can be seen as an approximation of the Laplacian operator. In a similar fashion as for the Laplacian blob detector, blobs can be detected from scale-space extrema of differences of Gaussians

Haar Transform erklären

(siehe foliensatz cv_05)

Die Haar-Transformation verwendet 4 Wavelets die aus der Kombination 2er Rechteck-Funktionen aufgebaut sind:

(1, 1), (1, -1), $(1,1)^t$, $(1,-1)^t$

Die resultierenden Wavelets entsprechen bestimmten Filtertypen:

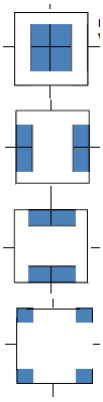
(1, 1; 1, 1) → Tiefpass

(1, -1;1, -1) → Vertikaler Hochpass

(1, 1;-1, -1) → Horizontaler Hochpass

(1, -1;-1, 1) → Diagonaler Hochpass.

Betrachtet man die Filter im Spektralbereich, decken sie das gesamte Spektrum ab.



Man kann mit diesen Haar-Wavelets auch eine Pyramide erzeugen indem man die 4 resultierenden Teilbilder um den Faktor 2 Downsampled, gemeinsam im Bereich des originalen Bildes anordnet, und das ganze iterativ mit dem Teilbild, das tiefpassgefiltert wurde, wiederholt. In der Pyramide finden sich dann alle Frequenzkomponenten separiert (Subband-Coding), aus der Pyramide kann man das original wieder vollständig rekonstruieren.

Moravec Corner Detector

Change of intensity in a small window that is shifted around

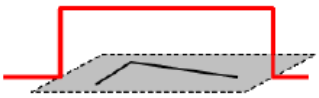
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

Intensity

Window function $w(x,y)$ =



Four shifts: (u,v) = (1,0), (1,1), (0,1), (-1, 1)

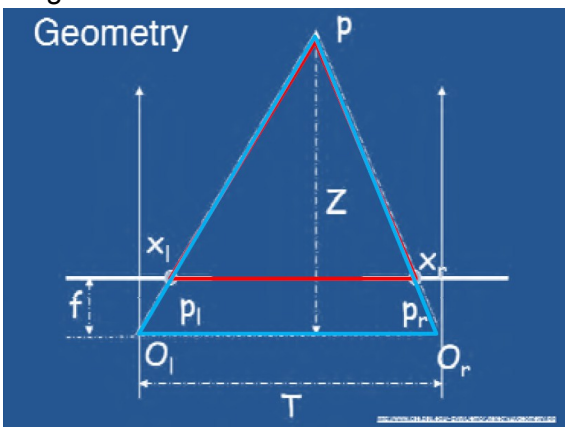
Look for local maxima in min{E}

3D Rekonstruktion aus stereoskopischen Bildern

Aus dem foliensatz 11

Bei paralleler optischer Achse

geometrischer zusammenhang:



mathematische berechnung:

$$Z = f\frac{T}{x_r - x_l}$$

Disparity

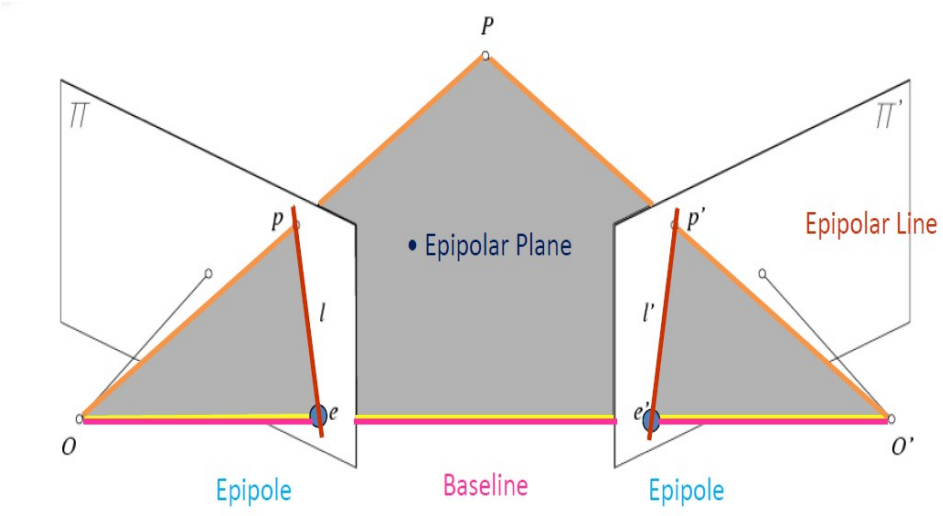
- T ... Basis
- f ... fokal laenge
- x_r - x_l ...disparity (auch als d angegeben)

Beispiel (reverse) von den Rechenbsps unten:
Gegeben: T = 60cm; d = 6mm; f = 8mm
Gesucht: Z
Ergebnis: 80

Bei nicht paralleler optischer Achse

Begriff wrap up:

- Baseline: line joining the camera centers
- Epipole: point of intersection of baseline with the image plane
- Epipolar plane: plane containing baseline and world point
- Epipolar line: intersection of epipolar plane with the image plane



!! Ein punkt im linken Bild muss auf der Epipolar Line im richten Bild liegen !!!

Beispiel der epi-linien und wo der punkt liegen könnte:

This is useful because it reduces the correspondence problem to a 1D search along an epipolar line. Lösungsansatz siehe Frage “Lösungen für das Korrespondenzproblem?” (bzw darüber)

‘Bag of Words’

Wozu verwendet man den ‘Bag of Words’-Approach

Das ist eine Methode, um Szenen zu erkennen. Aus einem Bild werden eine große Anzahl an lokalen Features, ‘Codewords’ (Deskriptoren), extrahiert. Diese werden in einem Histogram untersucht und dieses in einem Vektor codiert. Diesen Vektor kann man dann mit einem ‘Codeword-Dictionary’ abgleichen und so die Szene klassifizieren. (In dem Kontext wird in den Folien K-means clustering eingeschoben, das clustering selbst braucht man denk ich beim lernen um das codeword-dictionary zu erzeugen. In diesem Dictionary sind dann die Cluster-centroide gespeichert und beim Klassifizieren sucht man für ein Bild den nähesten Cluster.)
Analog zu Dokumenten: Schlüsselwörter, Auftreten von bestimmten Schlüsselwörtern erlaubt es auf den Inhalt eines Dokuments zu schließen.

Wie funktioniert der Bag of Words-Approach

2 Anwendungen: Lernen und Klassifizieren(Erkennen, Recognition).
Lernen:

- Lokale Features aus (vielen) Bildern extrahieren und codieren
- Cluster bilden → Category models (and/or) classifiers

Recognition:

- Aus einem Bild die lokalen Features extrahieren und codieren
- Den nähesten Cluster im codeword-dictionary suchen → zu dieser Kategorie gehört das untersuchte Bild.

Schwierigkeit dabei: Welche Features verwendet man?

- Regular Grid (Bild wird in regelmäßige Teilbilder zerlegt)
- Interest point detector (SIFT)
- Other methods
    - Random sampling
    - segmentation based patches

Schwächen von Bag of Words

- keine fundierte geometrische Information über die Objekte und Komponenten
- Intuitiv: Objekte bestehen aus Teilen → diese Information wird nicht berücksichtigt
- noch nicht ausführlich getestet für
    - viewpoint invariance
    - Scale invariance
- Gute segmentierung und lokalisierung noch nicht klar
- (Das ganze ist wohl noch nicht ausgereift für Bilder)

Depth Cues

What kind of Depth Cues (in Genreal) are there?

And describe the difference!

- Binocular: innate, biological (angeboren)
- Monocular: learned, environmental (angelernt)

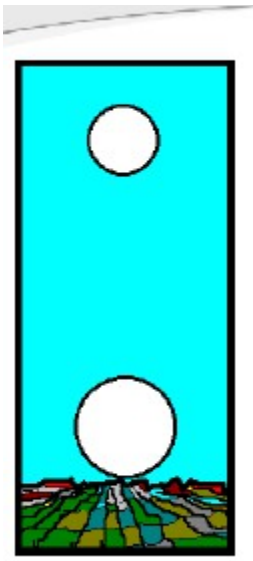
Followup: Categorize Monocular Depth Cues and describe them:

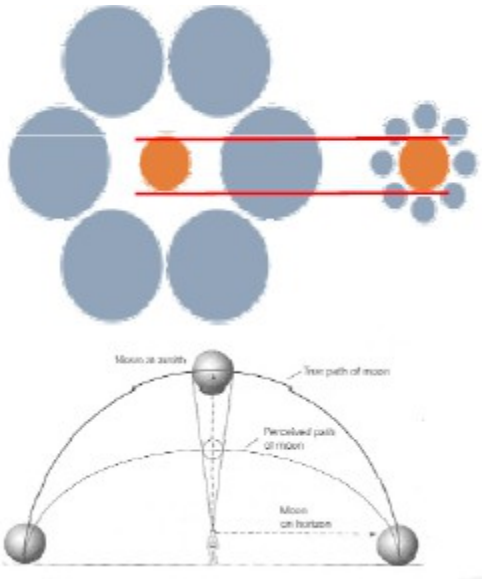
- Absolute: information about the absolute depth between the observer and elements
- Relative: relative information about depth between elements in the scene (provides us with a sense of which object (out of at least two objects) is farther away)

Welche (Monokularen) Cues gibt es für die Tiefenwahrnehmung?

- Überdeckung
  - Wenn ein Objekt ein anderes verdeckt kann man davon ausgehen dass es näher als das andere ist
- Textur-Gradient
  - Texturen sind gröber in der Nähe und werden kleiner und feiner in der Entfernung
- Beleuchtung / Schattierung / Schatten
  - Das Auge nimmt immer an, dass das Licht von oben kommt
- Lineare Perspektive
  - Fluchtpunkte von parallelen Linien
- Atmospherische Perpektive
  - Dinge in der Ferne haben weniger Kontrast und sind bläulicher durch die Atmosphäre
- Relative Größe
  - Wenn bei Objekten von denen wir wissen, dass sie etwa gleich groß sind eines kleiner erscheint, nehmen wir an es ist weiter weg
- Höhe auf der Horizontalen Ebene
  - Um so höher ein Objekt auf der horizontalen Ebene (bzw. um so näher am Horizont), um so weiter ist es entfernt
- Bewegungsparallaxe
  - Objekte die näher sind bewegen sich schneller/weiter

Moon Illusion:





Absolute:

- Bekannte Größen
    - Bekannte Objekte bekommen von unserem Auge automatisch eine bestimmte Größe zugeteilt. Wenn zwei bekannte Größen im Widerspruch zu anderen Cues stehen, dann wird von uns angenommen, dass die umgebung/szene die korrekte Größe hat und das andere objekt verfälscht ist

In english:

List and describe different Monocular Depth Cues:

- **Superimposition**: we have learned that several things are usually in front of others, such as the teacher in front of the blackboard
- **Texture Gradient**: we assume a certain pattern on the ground texture is kind-of-repeated. The varying size of the textures features give us a clue to the (relative) depth
- **Illumination**: The shadow gradient lets us perceive the objects structure -> depth cues on the object. Shadows also give a clue on where the object is related to the ground plane. Be aware: shading clues might trick you, as you can only perceive the gradient, but objects can be inside-out or outside-in. (optical illusion thingi)
- **Linear Perspective**: convergence of parallel lines give us a clue as well (keyword: vanishing point etc.)
- **Atmospheric Perspective**: the change in atmospheric effects (such as fog) give us clues as well
- **Relative size**: Several identical objects have a different size on the screen. we therefore make the assumption, that the bigger objects are closer to us
- **Height in the horizontal plane**: When an object on the image begins higher (on the vertical axis), we assume it is further away. This thing is base on the assumption that there is a almost flat ground plane and that the vanishing point lies uppwards. Example: Moon illusion: The moon seems to be bigger, when it is close to the horizon
- **Motion parallax**: Motion gives us a clue as well
- **Familiar Size**: Familiar sizes are assumed to be as high as the experience tells us. If two familiar size contradict each other, we assume that the environment/scene has the familiar size and that the other object is distoreted.

3D Object Categorization

Abhängig davon, von wo (aus welcher Perspektive) ein Bild von einem Objekt aufgenommen wurde, können wir es unterschiedlich schnell kategorisieren. Dabei hat sich herausgestellt, dass die “Canonical Perspective” (seitlich von vorn, sodass man das Objekt teilweise von vorn und von der Seite sieht) eines Objekts am schnellsten erkannt wird. Warum? Dafür gibt es zwei Hypothesen: die Frequenz Hypothese und die Maximal Information Hypothese.
Es hat sich herausgestellt, dass wahrscheinlich beide Hypothesen korrekt sind.

Was sagt uns die Canonical Perspective / Canonical Viewpoint

Phenomen, das sagt, dass wir bekannten Objekten immer auch einer bekannte Perspektive zuordnen. Sprich: bekannte Objekte aus unbekannten Blickwinkel wirken unnatürlich.

Was ist die Frequency Hypothesis?

Die Frequency Hypothesis besagt, dass die Schnelligkeit in der wir ein Objekt wahrnehmen in Beziehung dazu steht wie oft wir ein Objekt aus verschiedenen Perspektiven gesehen haben.
(easiness of recognition is related to the number of times we have see the objects from each viewpoint)

Maximal Information Hypothese

Manche Perspektiven geben mehr Information her als andere. Die besten Perspektiven zeigen mehrere Seiten des Objekts.

Was ist der Priming Effect?

Besagt, dass die Erkennbarkeit des Objekts steigt, wenn man es schon mal gesehen hat. Außerdem kann das Objekt in anderen Positionen und sogar ähnliche Objekte schneller erkannt werden.

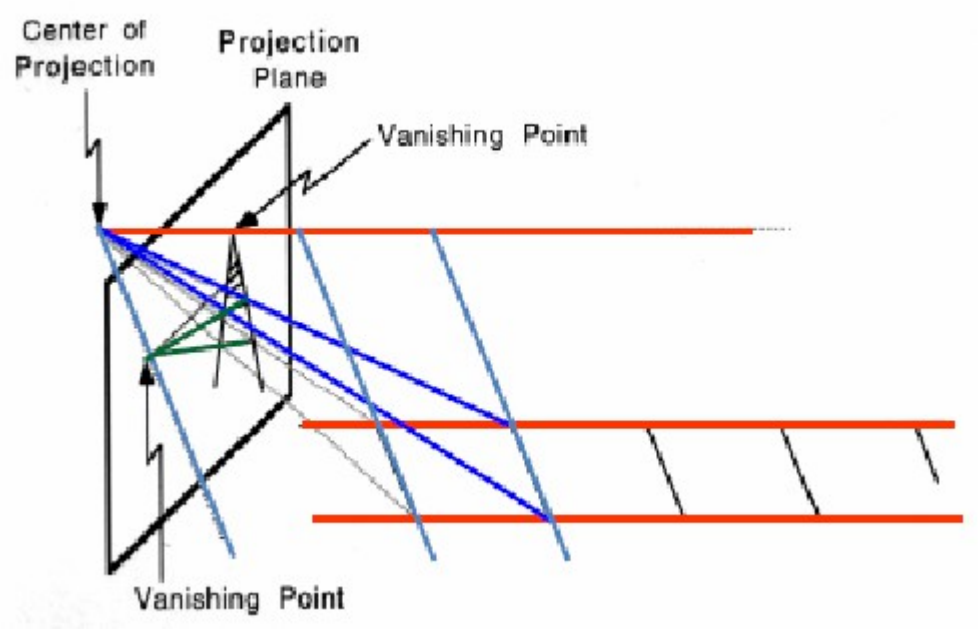
Fluchtpunkt? Fluchtlinie?

Der Fluchtpunkt einer Linie in Zentralprojektion ist jener Punkt im Bild, an dem die Projektion der Linie nicht fortgesetzt werden kann. Eine unendlich lange Linie flüchtet im Fluchtpunkt aus dem Bild. Der Fluchtpunkt einer Linie hängt von dessen Orientierung ab (nicht von ihrer Länge).

Parallele Linien treffen sich im Fluchtpunkt der Richtung. Gruppen von parallelen Linien auf der gleichen Ebene haben kollineare Fluchtpunkte.

Fluchtpunkt der Linie ist jener Punkt in der Bildebene in dem eine parallele Linie, die durch das Projektionszentrum geht, die Bildebene schneidet.

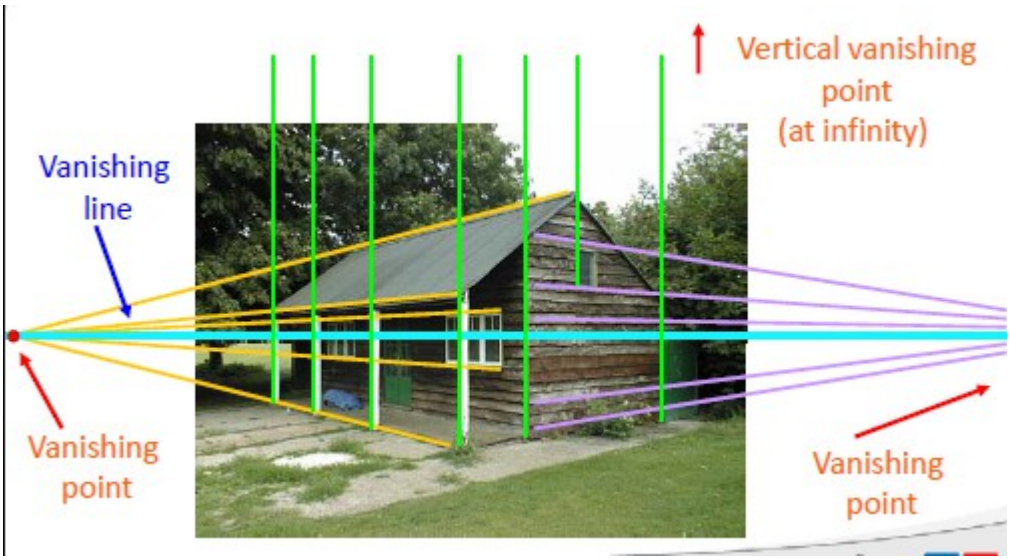
Fluchtlinie = Ebene durch Projektionszentrum



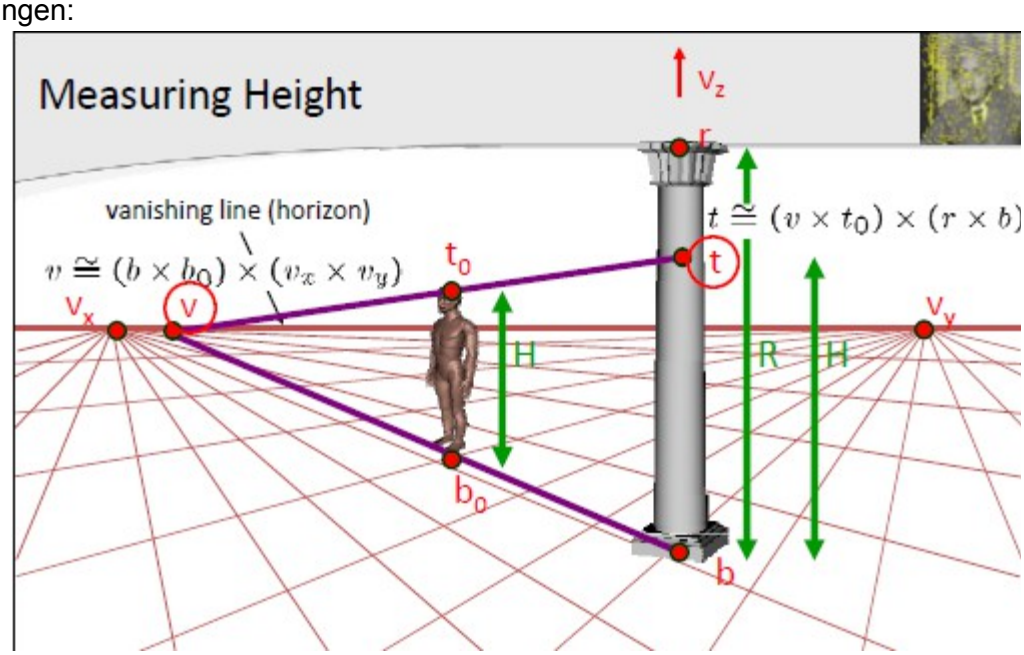
Single View Reconstruction

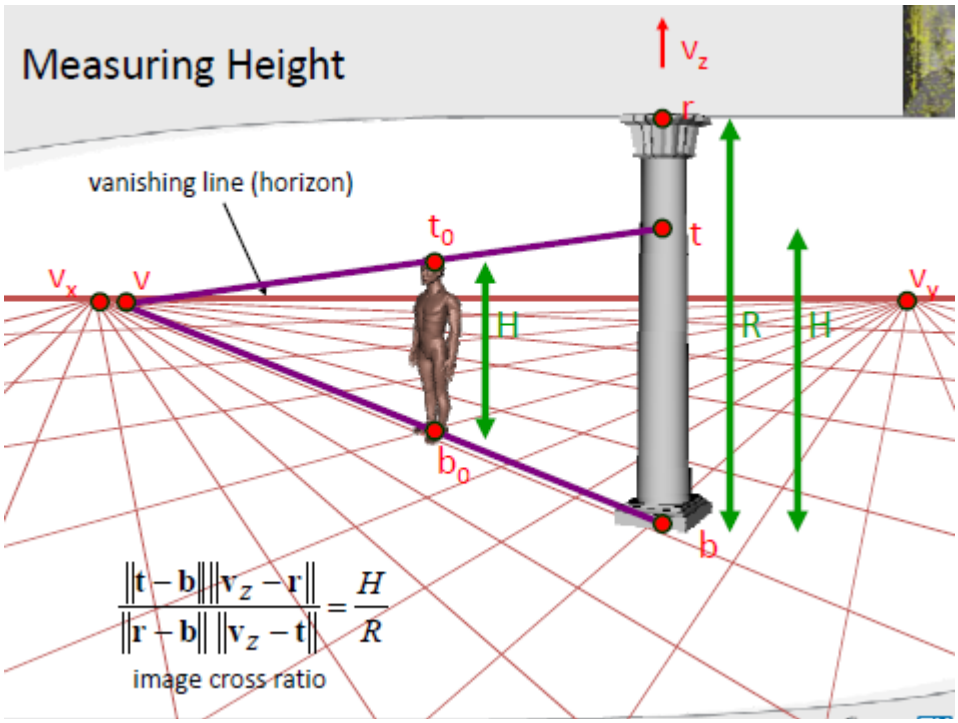
Calculate by Vanishing cues

Basics:



Berechungen:

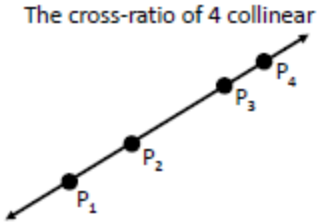




The Cross Ratio

- A Projective Invariant
  - Something that does not change under projective transformations (including perspective projection)

The cross-ratio of 4 collinear points



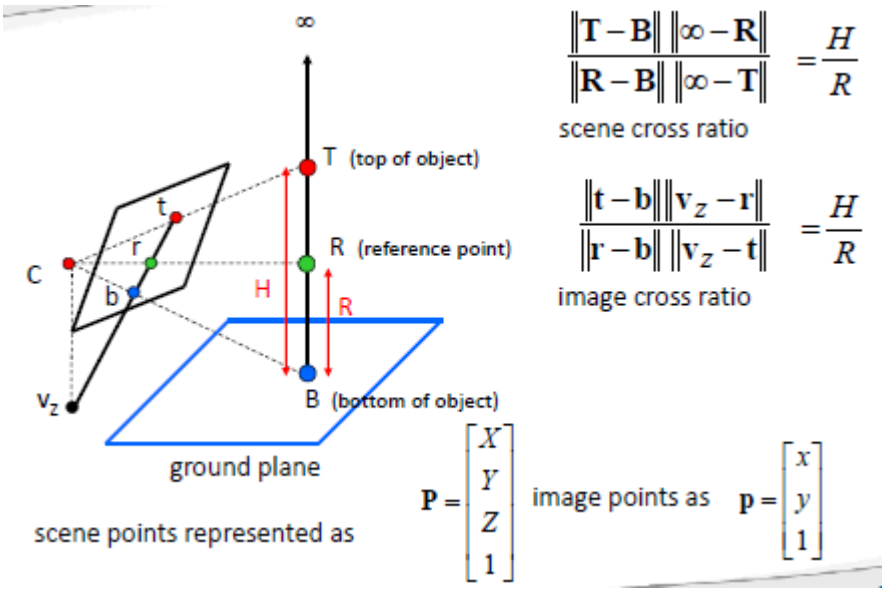
$$\frac{\|\mathbf{P}_3-\mathbf{P}_1\|\,\|\mathbf{P}_4-\mathbf{P}_2\|}{\|\mathbf{P}_3-\mathbf{P}_2\|\,\|\mathbf{P}_4-\mathbf{P}_1\|} \qquad \mathbf{P}_i=\begin{bmatrix}X_i\\Y_i\\Z_i\\1\end{bmatrix}$$

Can permute the point ordering $\frac{\|\mathbf{P}_1-\mathbf{P}_3\|\,\|\mathbf{P}_4-\mathbf{P}_2\|}{\|\mathbf{P}_1-\mathbf{P}_2\|\,\|\mathbf{P}_4-\mathbf{P}_3\|}$

- 4! = 24 different orders (but only 6 distinct values)

This is the fundamental invariant of projective geometry

Calculate height without a ruler



$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|} = \frac{H}{R}$$

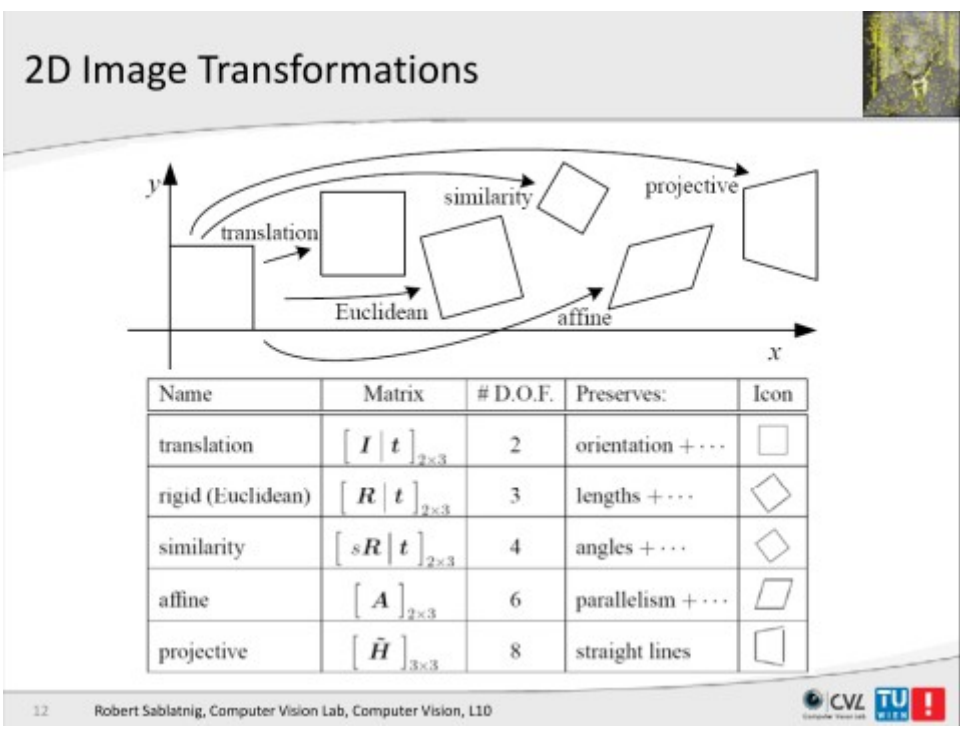
scene cross ratio

$$\frac{\|\mathbf{t}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|} = \frac{H}{R}$$

image cross ratio

needs more than the vanishing points - that is the groundplane and the camera orientation towards it

2D Transformationen (# D.o.F = Anzahl Freiheitsgrade)



Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} I \mid t \end{bmatrix}_{2\times3}$	2	orientation + ···	
rigid (Euclidean)	$\begin{bmatrix} R \mid t \end{bmatrix}_{2\times3}$	3	lengths + ···	
similarity	$\begin{bmatrix} sR \mid t \end{bmatrix}_{2\times3}$	4	angles + ···	
affine	$\begin{bmatrix} A \end{bmatrix}_{2\times3}$	6	parallelism + ···	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3\times3}$	8	straight lines	

Foliensatz 12 - Theoriefragen:

Basic Stereo Algorithmus

Beschreibung in Pseudocode

Für jede epipolare Linie {
    Für jeden Pixel im linken Bild {
        - vergleiche mit jedem Pixel im rechten Bild auf der selben epipolaren Linie
        - wähle Pixel mit minimaler match cost
    }
}

Wie kann der Algorithmus verbessert werden?

Match Windows Improvment für Basic Stereo Algorithmus:

Für jeden Pixel{
    Für jede Disparität{
        Für jeden Pixel im Fenster{
- Berechne Differenz
        }
    }
    - Finde die Disparität mit der minimalen SSD (sum of squared differences)
}

Was ist bei der Auswahl der Fenstergröße zu beachten?

- kleines Fenster: mehr Details - mehr Rauschen
- großes Fenster: weniger Details - robuster (weniger Rauschen)
- Kompromiss: großes Fenster - mit höherer Gewichtung in Richtung Fensterzentrum (Gauss)

Algorithmus: Berechne gewichtete SSD

Energie Minimierung allgemein

Wenn Daten und Energie Terme kontinuierlich, smooth etc. sind versuche Minimierung unter Verwendung der Gradientensteigung.
In der Praxis sind die Disparitäten immer nur Stückweise smooth.
Stelle smoothness Funktion auf welche große Sprünge in der Smoothness nicht zu stark gewichtet/benachteiligt.

Globale Minima sind durch Non-Smooth-Funktionen schwer zu finden:

- viele globale Minima
- möglicherweise NP-schweres Problem

Praktikable Algorithmen suchen nach Annäherungen an Minima.

Formel zur Energie Minimierung

$E_{data} + [?][?] E_{smoothness}$

E_{data}.....Qualität der Übereinstimmung zwischen Daten und Disparität
$E_{smoothness}$.....Qualität der Übereinstimmung zwischen den Disparitäten der Nachbarn (Regularization)

Schritte und Fehlerursachen der Stereo Reconstruction Pipeline

Schritte:

- Kamerakallibrierung
- Begradigen der Bilder
- Berechnen der Disparität
- Bestimmen der Tiefe

Fehlerursachen:

- Kamerakallibrierungs - Fehler
- Geringe Auflösung der BIlder
- Überdeckungen in den Bildern
- Spiegelungen
- Starke Bewegungen
- Bildregionen mit geringem Kontrast

Was ist das Depth-of-Field Problem?

- Mehere Fokalebenen erzeugen verschwommene Bildregionen
- Kann bei stereoskopischen Sehen unnatürlich wirken bzw. Kopf und Augenschmerzen erzeugen.
- Die ideale stereoskopische Kamera würde eine unendliche Tiefenunschärfe haben:
    - kleiner Rahmen/Sensor -> verringerte Qualität
    - verringerte Blende
- Ideal kann bei animierten Filmen erreicht werden

Übersicht Gauss / Laplace -Stuff

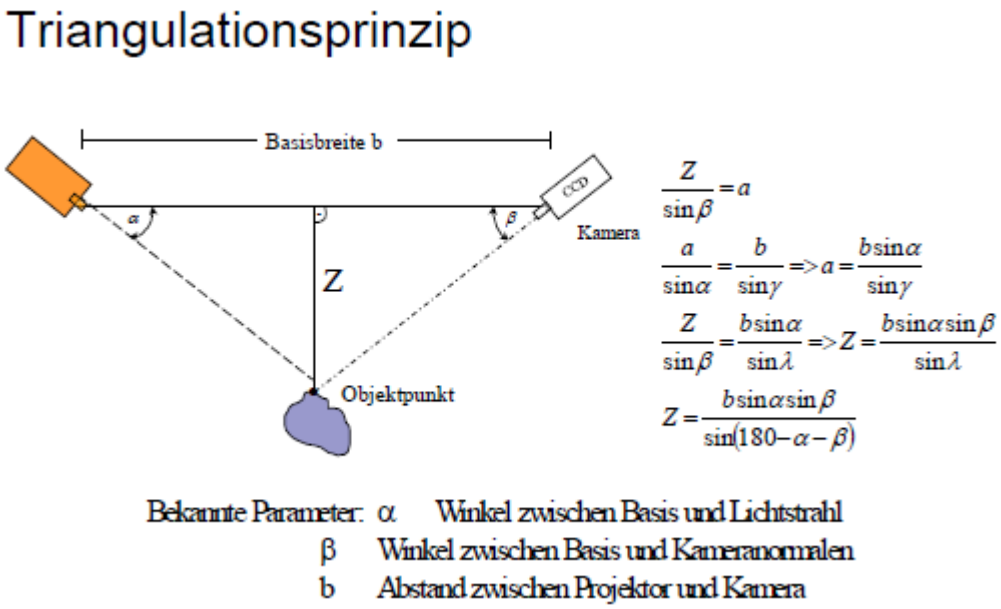
Gauss-Pyramide	Difference of Gaussian	Laplacian of Gaussian	Laplace Pyramide
Bild wird auf jeder Stufe mit gleichem Gauss-Kernel geblurred / komprimiert (Size wird um jeweils ¼ verringert, d.h. nur jedes 2. Pixel in x-Richtung bzw. in y-Richtung wird berücksichtigt)	Zwei Bilder die mit unterschiedlichen Gauss-Kerneln geblurred wurden werden von einander abgezogen	=Summe der 2.Ableitungen der Gauss-Funktion (Mexican Hat, durch abziehen der zweiten Ableitung invertiert man ihn))	DoG wird angewendet. 2 Level der Gauss-Pyramide werden von einander abgezogen (das kleinere muss auf die gleiche Größe aufskaliert werden)

(Size wird um jeweils ¼ verringert bei der Gauss-Pyramide???? Die Skizze weiter oben zeigt was anderes)

Rechenbeispiele

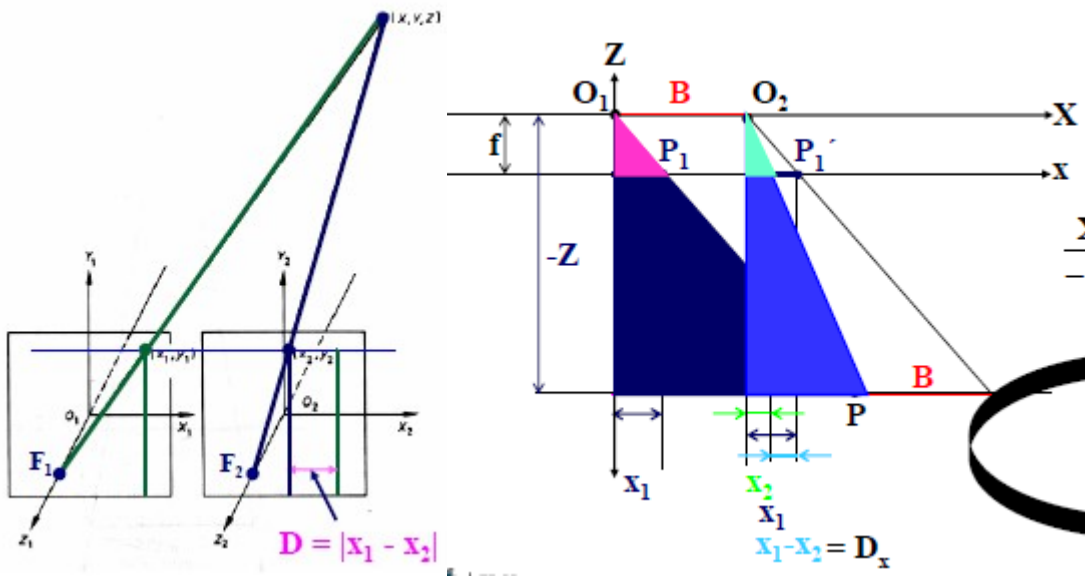
Triangulation

Pojektor & Kamera



law of sines: a/sin(alpha) = b/sin(beta) = c/sin(gamma)

Stereoauswertung



Fokale Länge f gesucht, gegeben waren Basisbreite b, Disparität D, -Z

Ähnliche Dreiecke:

$$\boxed{\frac{X}{-Z} = \frac{x_1}{f} \qquad \frac{X-B}{-Z} = \frac{x_2}{f}}$$

$$X = -\frac{Zx_1}{f} \qquad X = -\frac{Zx_2}{f} + B$$

$$\boxed{-\frac{Zx_1}{f} = -\frac{Zx_2}{f} + B}$$

$$-B = \frac{Zx_1}{f} - \frac{Zx_2}{f}$$

$$-B = Z\frac{x_1 - x_2}{f} \qquad \boxed{D = x_1 - x_2}$$

$$-B = Z\frac{D}{f}$$

$$f = \frac{ZD}{-B}$$

Gegeben:fokale Länge f (8mm),Objekttiefe z (80cm), Disparität d (6mm). Gesucht: Basis b

b = (z * d) / f oder -b = (z * d) / f oder b = -(z * d) / f
(kommt auf formulierung an ob -Z oder Z)
Herleitung siehe vorherige Frage
(Lösung: 60cm).

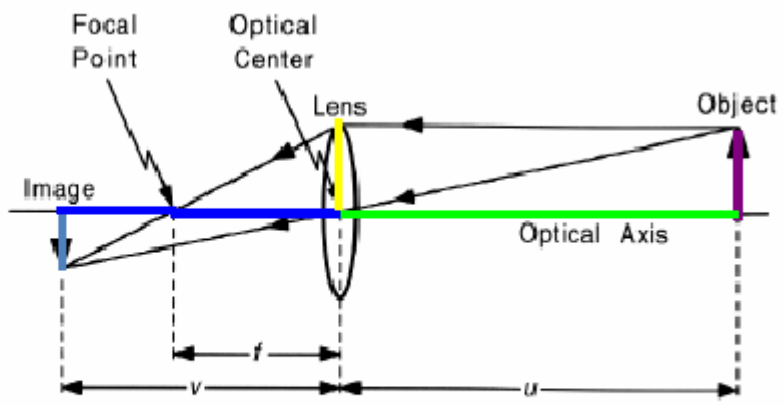
Thin Lens Equation:

<u>a) gegeben u,yo,f gesucht v b) gesucht yi aufbauend auf a)</u>
siehe Theoriefrage “Dünnes Linsenprinzip mit Skizze und Formel für Linsengleichung”.

- Linsengesetz:

$$\frac{y_0}{y_i} = \frac{u}{v}$$

$$\frac{y_0}{y_i} = \frac{f}{v-f}$$



Thin-Lens Equation: $\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$$

$$\frac{1}{v} = \frac{1}{f} - \frac{1}{u} \quad => \quad \frac{1}{v} = \frac{u}{fu} - \frac{f}{fu}$$

$$\frac{1}{\frac{u-f}{fu}} = v$$

$$v = \frac{fu}{u-f}$$

a)

$$\frac{y_0}{y_i} = \frac{u}{v} \qquad \frac{y_0}{y_i} = \frac{f}{v-f}$$

$$y_i = \frac{(y_0 v)}{u}$$

b)

Bayer Pattern

Berechnen der RGB Werte für einen Bildpunkt nach dem Bayer pattern.

3x3 ausschnitt; aufnahme mit CCD-bildsensor; color filter array (bayer filter) verwendet:
GB ...
RG
.
.
berechnen sie den RGB-Farbwert an position *:
G:200 | B:150 | G:190 | 130 | 170
R:110 | G:90 | R:140 | 100 | 130
G:210 | B:120*| G:200 | 140 | 200
R:140 | G:100 | R:170 | 120 | 140
220 | 120 | 210 | 110 | 180

Bilineare Interpolation:
R: (110 + 140) / 2 = 125; (140 + 170) / 2 = 155; (125 + 155) / 2 = 140
bzw.:
R = (110 + 140 + 140 + 170) / 4 = 140
G = (90 + 210 + 200 + 100) / 4 = 150
B = 120

Da die verwendeten Grün-Werte näher sind als die Rot-Werte ist auch die interpolation für Grün genauer als die für Rot.

Disparität mit sliding window ausrechnen (fenster 4 mal angelegt)

Gleich wie das nächste denk ich

Stereoauswertung: Ähnlichkeitsmaß ermitteln

ein gegebener 3x3-Filterkern auf eine gegebene Zeile eines Bildes anwenden und den Punkt mit dem besten Ähnlichkeitsmaß ermitteln (Formel zur Berechnung des Ähnlichkeitsmaß war ebenfalls gegeben). Für diesen Punkt dann die Disparität berechnen.

Hab hier jeweils in Klammern die beiden Pixelwerte die von IL und IR subtrahiert werden
S(37) = |(30 - 80)| + |(10 - 10)| + |(40 - 20)| + |(20 - 20)| + |(100 - 100)| + |(20 - 20)| + |(40 - 20)| + |(30 - 60)| + (20 - 30) = 130
Die restlichen genau so nur dass die Werte ausm rechten Bild um 1 nach rechts geschoben sind.
Dann sieht man S(39) ist das minimum, also die Disparität ist 100 - 39 = 61.

Measuring height

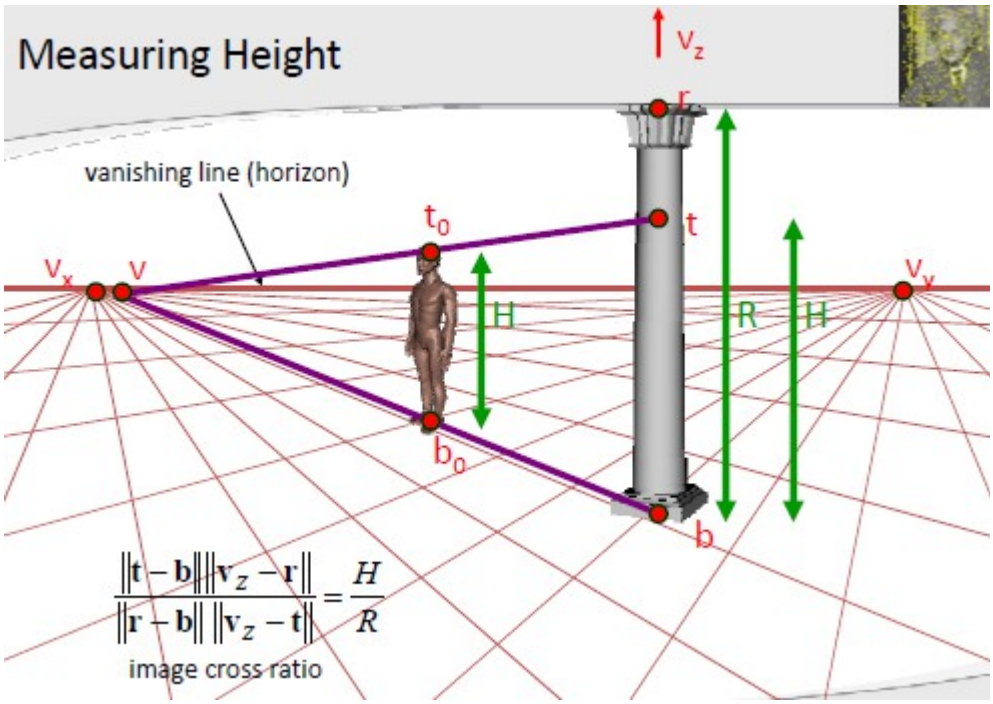
Diagramm mit der Säule und dem Menschen aus den Folien (WS 2011/12: Seite 26 aus *cv_08.pdf*). Gegeben sind die Koordinaten der Punkte v, b, t0 und r, sowie die Höhe der Säule (2,10m). Es ist auch die Formel für das Kreuzprodukt angegeben. Gesucht ist die Höhe des Menschen (Lösung: 1,80m).

$$v = \begin{pmatrix} 4 \\ 3 \\ 1 \end{pmatrix}, b = \begin{pmatrix} 8 \\ 11 \\ 1 \end{pmatrix}, t_0 = \begin{pmatrix} 3 \\ 7 \\ 1 \end{pmatrix}, r = \begin{pmatrix} 1 \\ 11 \\ 1 \end{pmatrix}$$

<u>Rechenweg:</u>
t = ((v x t0) x (r x b)), t = (2, 11, 1)^t, ist dann der Schnittpunkt vom Mensch auf der Säule.
Länge von b->r, also |r - b| = 7 = 2,10m
Länge von b->t, also |t - b| = 6 = x

t = (-56, -308, -28) müsste das sein. Nachdem da mit homogenen Koordinaten gerechnet wird kriegt mas dann wieder so (x,y,z) → (x/z, y/z, 1), also ((-56/-28), (-308/-28), 1) = (2, 11, 1)

Dann nur das Verhältnis ausrechnen: 7/6 = 2,10m / x → x = 2,10 * (7/6) = 1,8m



Prüfungsfragen 29.01.2013

1. Rechenbeispiel zu Stereovision, gegeben: f = 16mm, d = 4mm, Z = 40cm, gesucht Basis B
2. 4D-Lichtfeldkamera, was ist das (was ist der unterschied zu einer normalen Kamera)
    a. Wie Funktioniert das?
    b. warum nennt man das 4D?
3. Reflektion: Wie reflektiert eine Lambertsche und wie reflektiert eine spekulare Oberfläche (Spiegel)?
4. SIFT
    a. Was liefert SIFT für jeden Keypoint
    b. Wie wird bei SIFT die Rotationsinvarianz erreicht?
5. Cues
    a. Was sind Absolute und Relative Depth-Cues
    b. 3 Depth Cues aufzählen und erklären
6. Pinhole-Kamera erklären
    a. Was passiert bei zu großer/zu kleiner Öffnung
7. Gauss-Pyramide
    a. Wie kann man aus der Gauss-Pyramide eine Laplace-Pyramide erzeugen
8. RANSAC die selben Ja/Nein Fragen wie schon bekannt
9. Harris-Corner: 3 Bilder (homogener Bereich, Kante, Ecke), frage bei welchem die Eigenwerten klein, mittel, groß sind.
    a. Ist Harris-Corner Scale-Invariant?
10. Rechenbeispiel zu Höhenmessung, gleiches Bsp, gleiche Zahlen