Shape from X

Readings:

Szeliski: **Chapter 11 (11.1,11.2,11.3,11.4)**
Chapter 12 (12.1,12.2)

Some Slides adapted from Univ. of Washington
http://www.cs.washington.edu/education/courses/cse576/08sp/

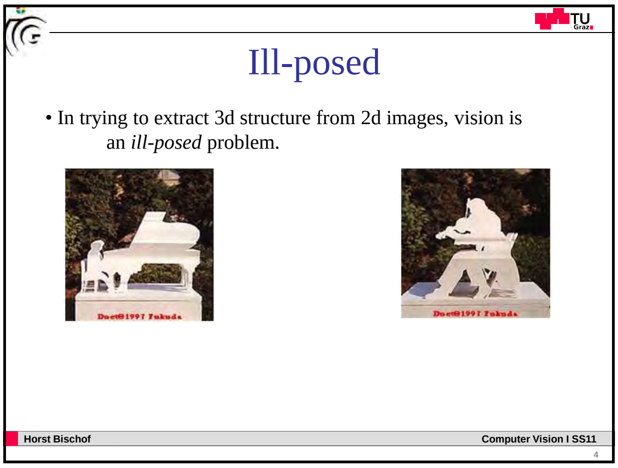
Shape from X

or

How to get 3D Information

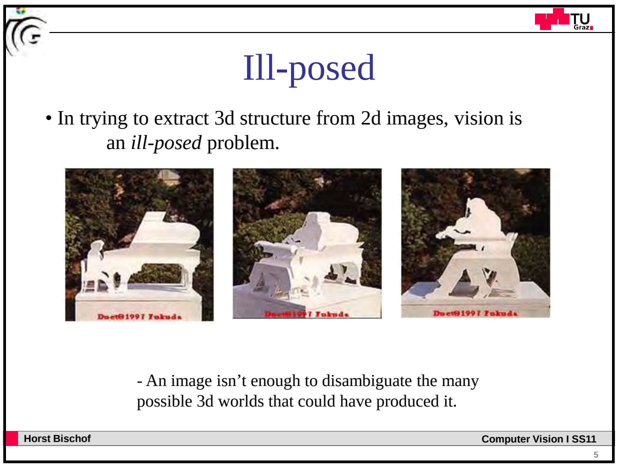
Ill-posed

- In trying to extract 3d structure from 2d images, vision is an *ill-posed* problem.



Ill-posed

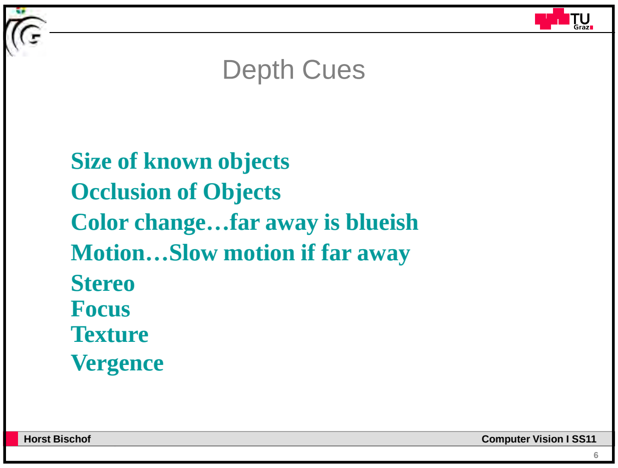
- In trying to extract 3d structure from 2d images, vision is an *ill-posed* problem.



Ill-posed

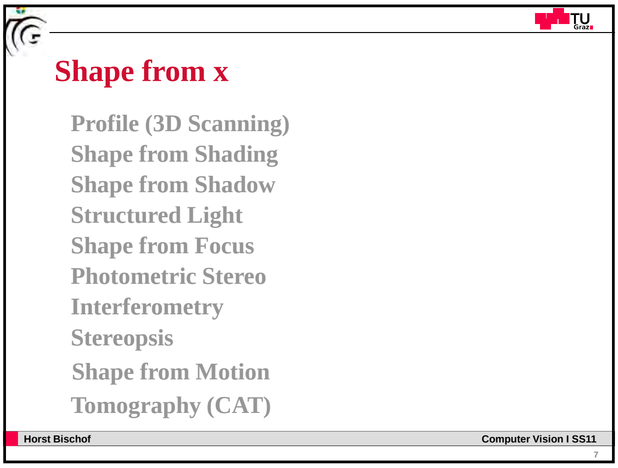
- In trying to extract 3d structure from 2d images, vision is an *ill-posed* problem.

- An image isn't enough to disambiguate the many possible 3d worlds that could have produced it.



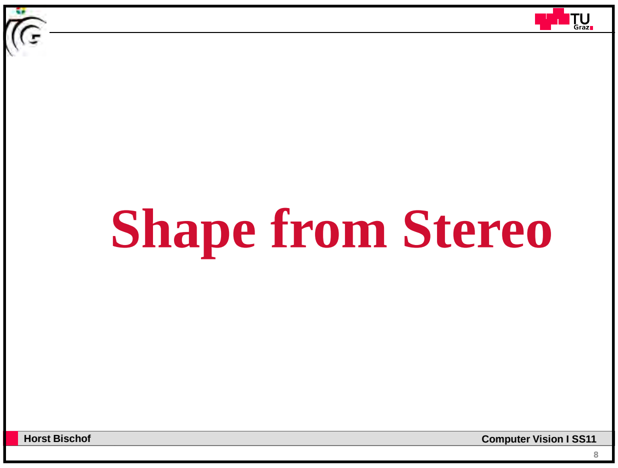
Depth Cues

Size of known objects
Occlusion of Objects
Color change…far away is blueish
Motion…Slow motion if far away
Stereo
Focus
Texture
Vergence

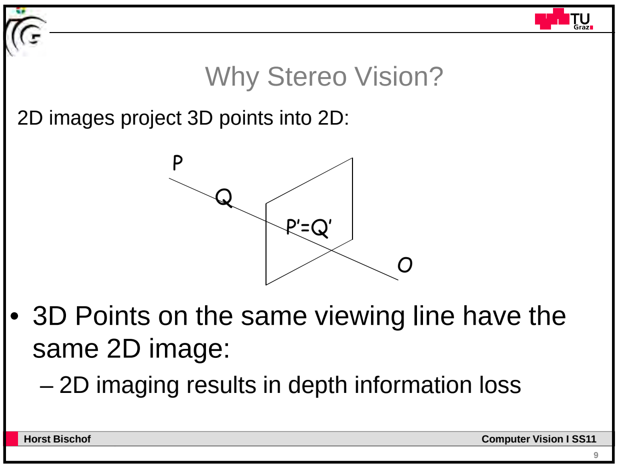


Shape from x

- Profile (3D Scanning)
- Shape from Shading
- Shape from Shadow
- Structured Light
- Shape from Focus
- Photometric Stereo
- Interferometry
- Stereopsis
- Shape from Motion
- Tomography (CAT)



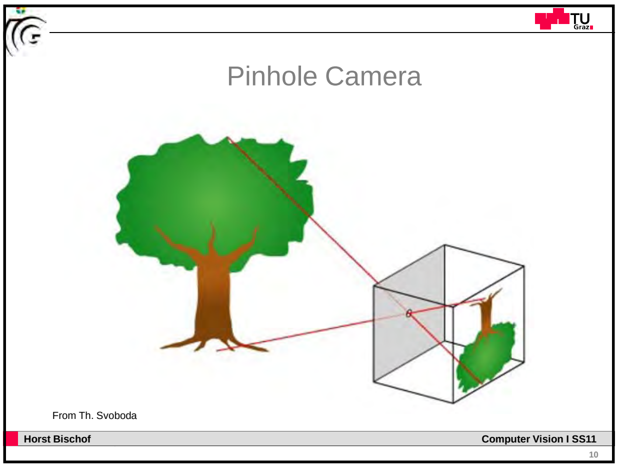
Shape from Stereo



Why Stereo Vision?

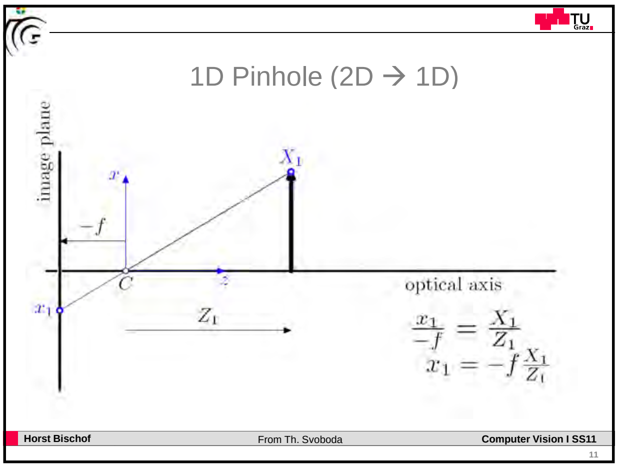
2D images project 3D points into 2D:

- 3D Points on the same viewing line have the same 2D image:
  - 2D imaging results in depth information loss



Pinhole Camera

From Th. Svoboda

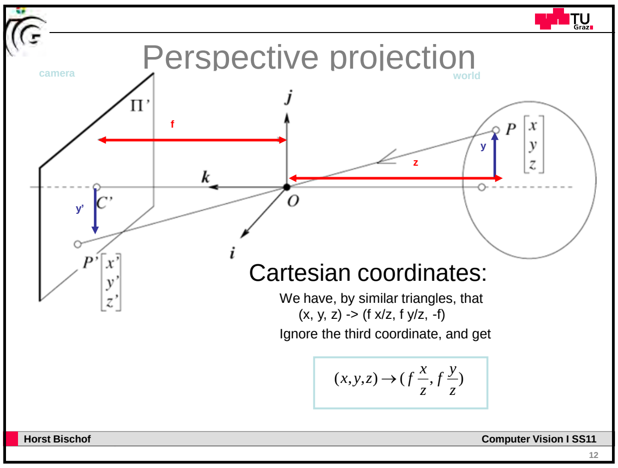


1D Pinhole (2D → 1D)

$$\frac{x_1}{-f} = \frac{X_1}{Z_1}$$

$$x_1 = -f\,\frac{X_1}{Z_1}$$

From Th. Svoboda

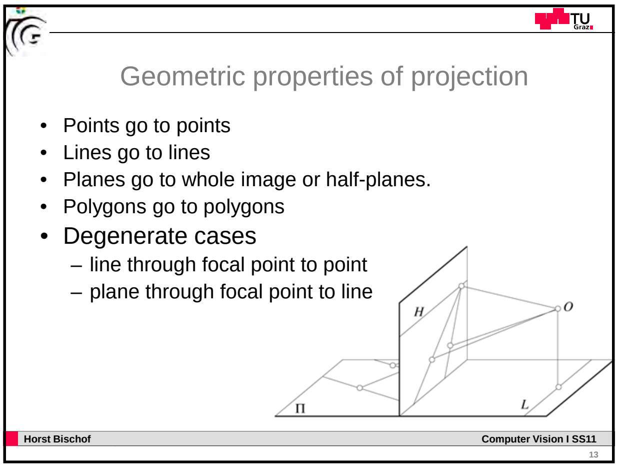


Perspective projection

Cartesian coordinates:

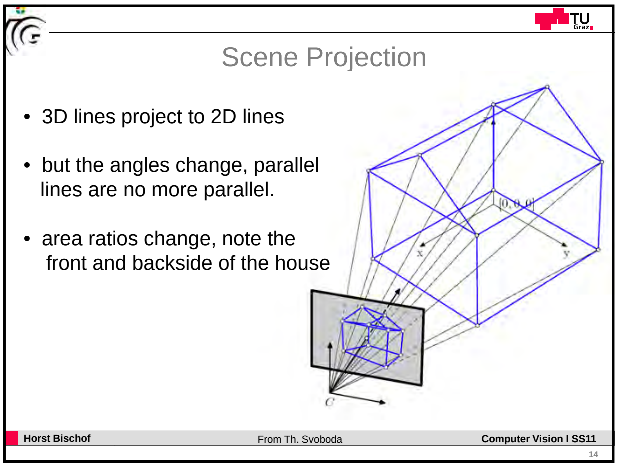
We have, by similar triangles, that
(x, y, z) -> (f x/z, f y/z, -f)
Ignore the third coordinate, and get

$$\boxed{(x,y,z) \rightarrow (f\,\frac{x}{z}, f\,\frac{y}{z})}$$



Geometric properties of projection

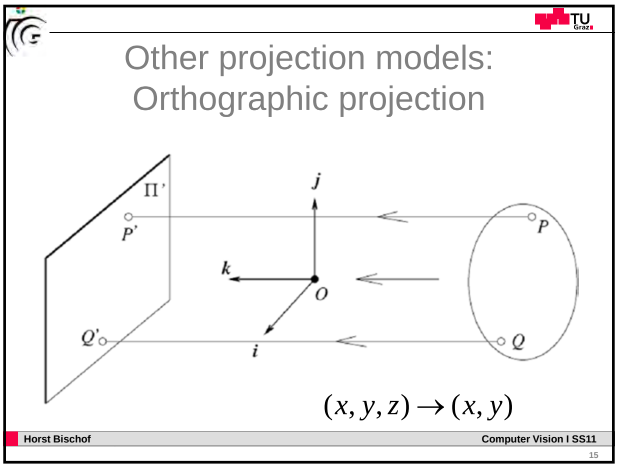
- Points go to points
- Lines go to lines
- Planes go to whole image or half-planes.
- Polygons go to polygons
- Degenerate cases
  - – line through focal point to point
  - – plane through focal point to line



Scene Projection

- 3D lines project to 2D lines
- but the angles change, parallel lines are no more parallel.
- area ratios change, note the front and backside of the house

From Th. Svoboda

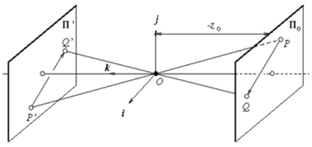


Other projection models: Orthographic projection

$$(x, y, z) \rightarrow (x, y)$$

Other projection models: Weak perspective

- Issue
  - perspective effects, but not over the scale of individual objects
  - collect points into a group at about the same depth, then divide each point by the depth of its group
  - Adv: easy
  - Disadv: only approximate



$$(x, y, z) \rightarrow \left(\frac{fx}{z_0}, \frac{fy}{z_0} \right)$$

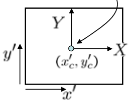
Recap: Camera parameters

A camera is described by several parameters

- Translation T of the optical center from the origin of world coords
- Rotation R of the image plane
- focal length f, principle point (x'_c, y'_c), pixel size (s_x, s_y)
- blue parameters are called "extrinsics," red are "intrinsics"

Projection equation

$$\mathbf{x} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi X}$$



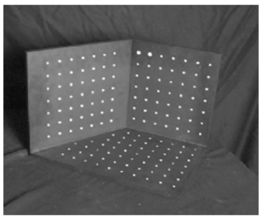
- The projection matrix models the cumulative effect of all parameters
- Useful to decompose into a series of operations

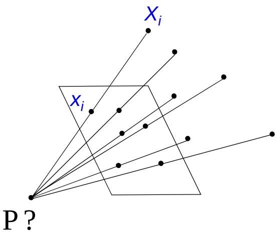
identity matrix

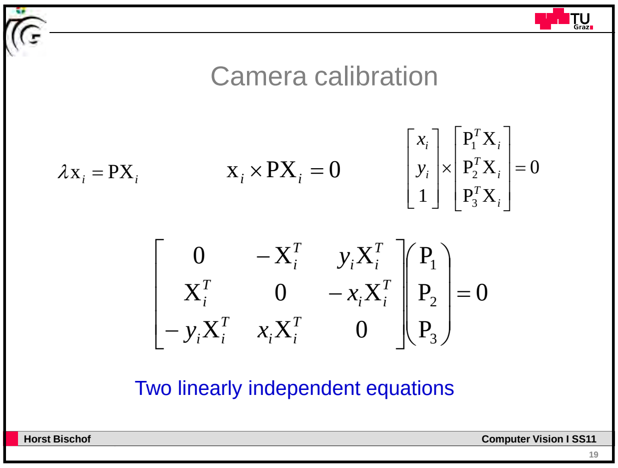
$$\mathbf{\Pi} = \underbrace{\begin{bmatrix} -fs_x & 0 & x'_c \\ 0 & -fs_y & y'_c \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsics}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} \mathbf{R}_{3x3} & \mathbf{0}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{T}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{translation}}$$

Camera calibration

- Given n points with known 3D coordinates X_i and known image projections x_i, estimate the camera parameters





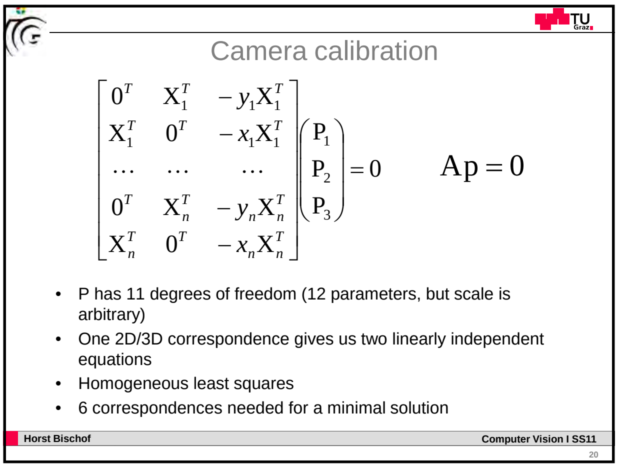


Camera calibration

$\lambda x_i = PX_i$ $\qquad$ $x_i \times PX_i = 0$ $\qquad$ $\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \times \begin{bmatrix} P_1^T X_i \\ P_2^T X_i \\ P_3^T X_i \end{bmatrix} = 0$

$$\begin{bmatrix} 0 & -X_i^T & y_i X_i^T \\ X_i^T & 0 & -x_i X_i^T \\ -y_i X_i^T & x_i X_i^T & 0 \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} = 0$$

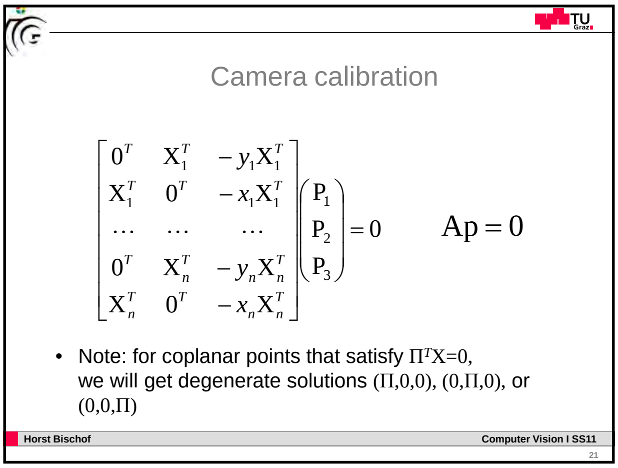
Two linearly independent equations



Camera calibration

$$\begin{bmatrix} 0^T & X_1^T & -y_1 X_1^T \\ X_1^T & 0^T & -x_1 X_1^T \\ \cdots & \cdots & \cdots \\ 0^T & X_n^T & -y_n X_n^T \\ X_n^T & 0^T & -x_n X_n^T \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} = 0 \qquad Ap = 0$$

- P has 11 degrees of freedom (12 parameters, but scale is arbitrary)
- One 2D/3D correspondence gives us two linearly independent equations
- Homogeneous least squares
- 6 correspondences needed for a minimal solution



Camera calibration

$$\begin{bmatrix} 0^T & X_1^T & -y_1 X_1^T \\ X_1^T & 0^T & -x_1 X_1^T \\ \cdots & \cdots & \cdots \\ 0^T & X_n^T & -y_n X_n^T \\ X_n^T & 0^T & -x_n X_n^T \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} = 0 \qquad Ap = 0$$

- Note: for coplanar points that satisfy $\Pi^T X=0$, we will get degenerate solutions $(\Pi,0,0)$, $(0,\Pi,0)$, or $(0,0,\Pi)$

Camera calibration

- Once we've recovered the numerical form of the camera matrix, we still have to figure out the intrinsic and extrinsic parameters
- This is a matrix decomposition problem, not an estimation problem

---

Alternative: multi-plane calibration

Images courtesy Jean-Yves Bouguet, Intel Corp.

Advantage

- Only requires a plane
- Don't have to know positions/orientations
- Good code available online!
  - Intel's OpenCV library: http://www.intel.com/research/mrl/research/opencv/
  - Matlab version by Jean-Yves Bouget: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html
  - Zhengyou Zhang's web site: http://research.microsoft.com/~zhang/Calib/

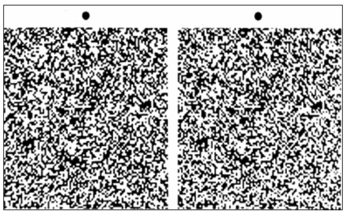
Stereo Vision

Refers to the ability of:

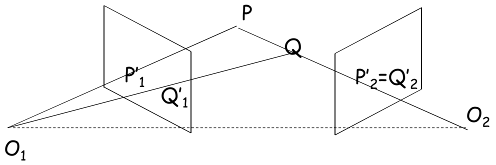
The ability to infer information on the 3D structure and distance of a scene from two or more images taken from different viewpoints.

Random Dot Stereograms

1960 to proof that only Stereo information is necessary [Julez59]



Recovering Depth Information:



Depth can be recovered with two images and triangulation.

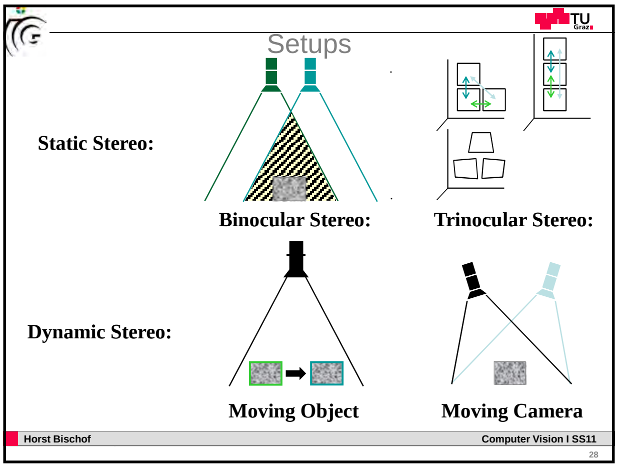
Stereo Vision Problems:

Correspondence Problem:

- Determining which pixel on the left corresponds to which pixel on the right.

Reconstruction Problem:

- Given a number of correspondence pairs and camera geometry information, find location and 3D structure of the observed objects.



Setups

Static Stereo:

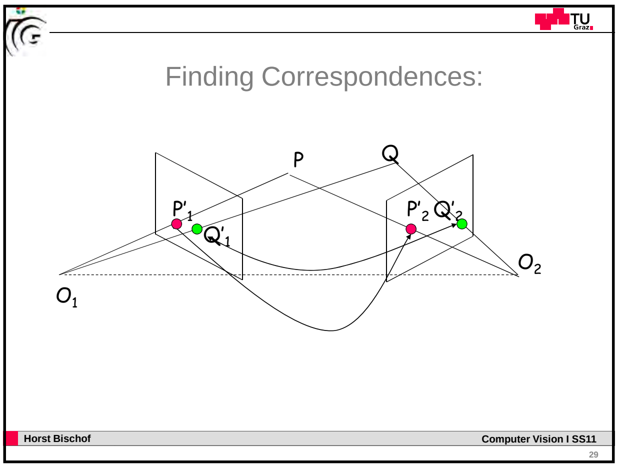
Binocular Stereo:

Trinocular Stereo:

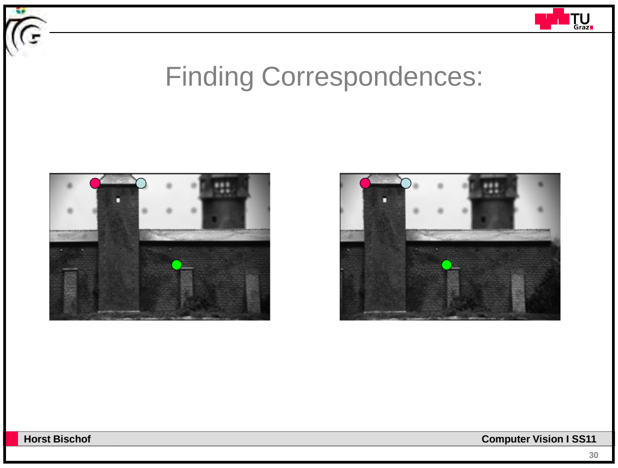
Dynamic Stereo:

Moving Object

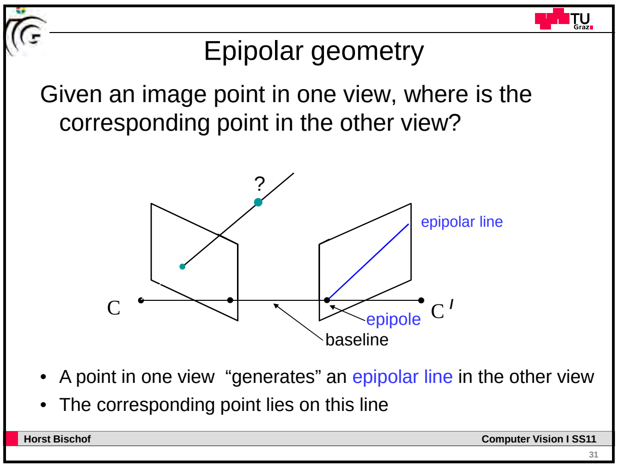
Moving Camera



Finding Correspondences:



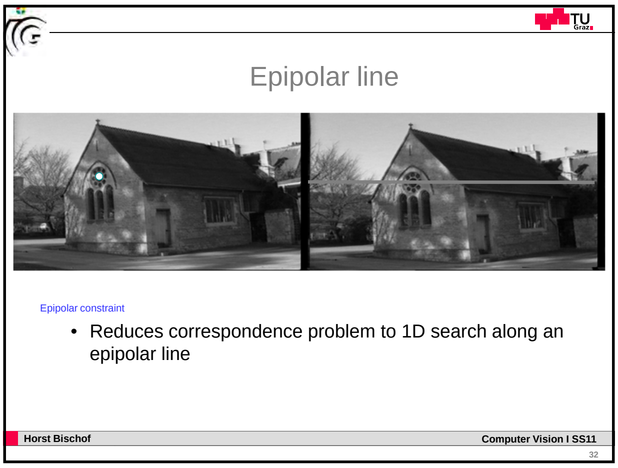
Finding Correspondences:



Epipolar geometry

Given an image point in one view, where is the corresponding point in the other view?

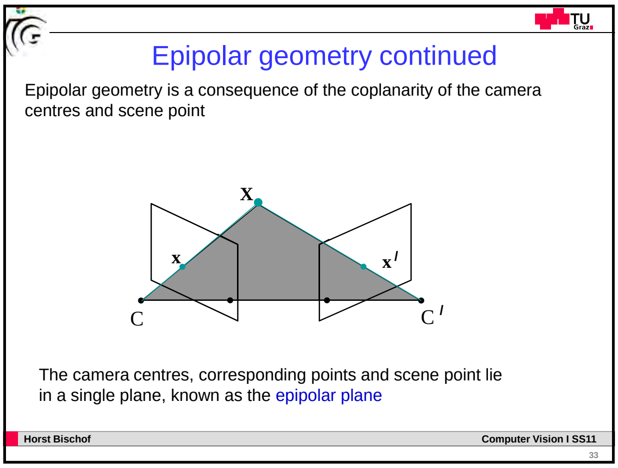
- A point in one view "generates" an epipolar line in the other view
- The corresponding point lies on this line



Epipolar line

Epipolar constraint

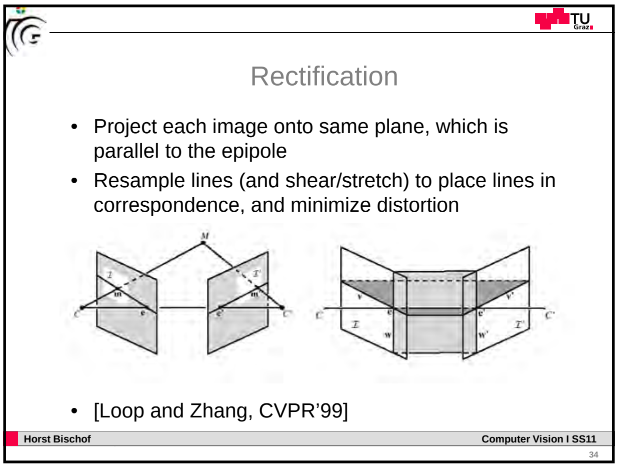
- Reduces correspondence problem to 1D search along an epipolar line



Epipolar geometry continued

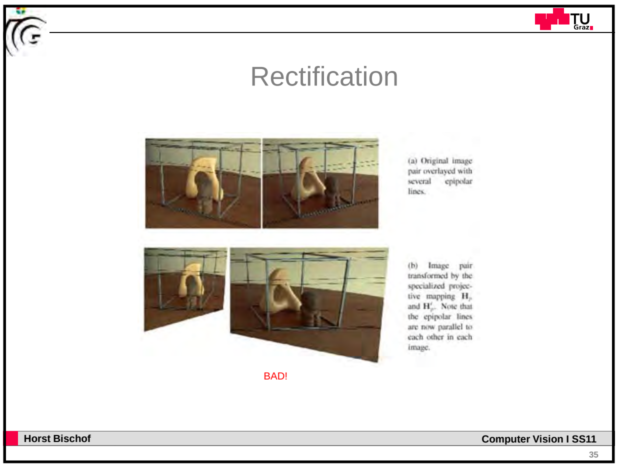
Epipolar geometry is a consequence of the coplanarity of the camera centres and scene point

The camera centres, corresponding points and scene point lie in a single plane, known as the epipolar plane



Rectification

- Project each image onto same plane, which is parallel to the epipole
- Resample lines (and shear/stretch) to place lines in correspondence, and minimize distortion
- [Loop and Zhang, CVPR'99]

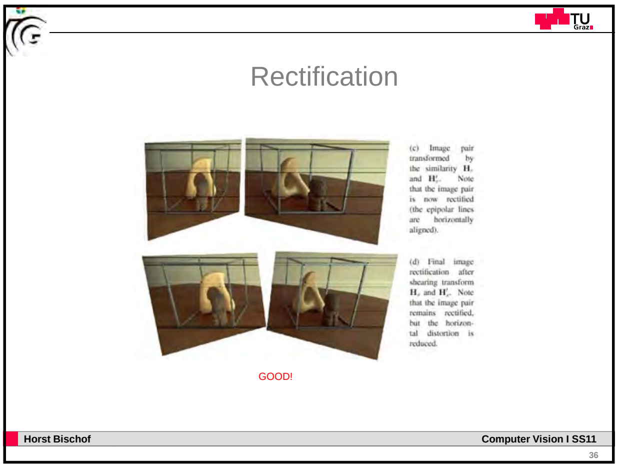


Rectification

(a) Original image pair overlayed with several epipolar lines.

(b) Image pair transformed by the specialized projective mapping H_p and H'_p. Note that the epipolar lines are now parallel to each other in each image.

BAD!

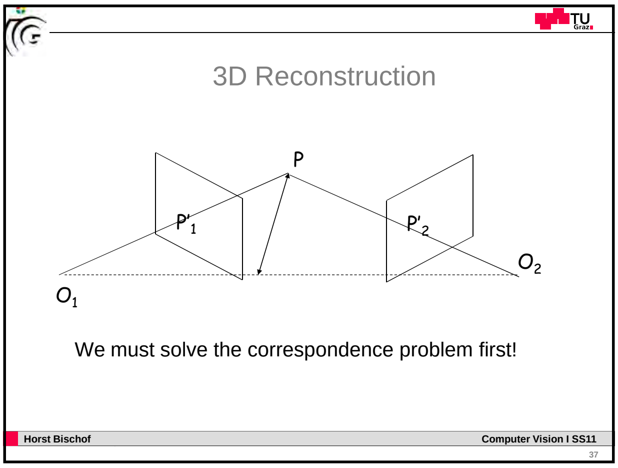


Rectification

(c) Image pair transformed by the similarity H_r and H'_r. Note that the image pair is now rectified (the epipolar lines are horizontally aligned).

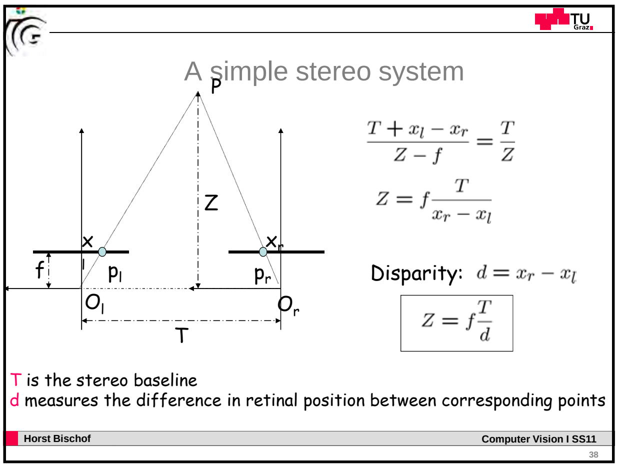
(d) Final image rectification after shearing transform H_s and H'_s. Note that the image pair remains rectified, but the horizontal distortion is reduced.

GOOD!



3D Reconstruction

We must solve the correspondence problem first!



A simple stereo system

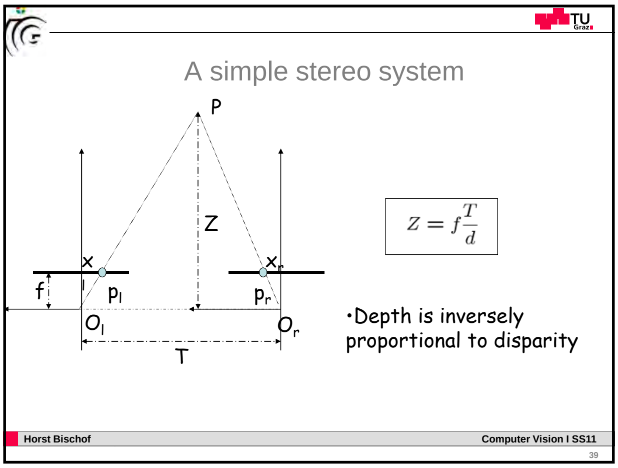
$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z}$$

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

Disparity: $d = x_r - x_l$

$$\boxed{Z = f\frac{T}{d}}$$

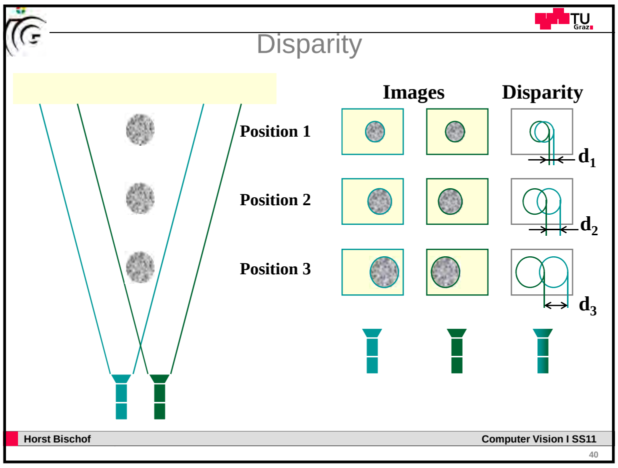
T is the stereo baseline
d measures the difference in retinal position between corresponding points



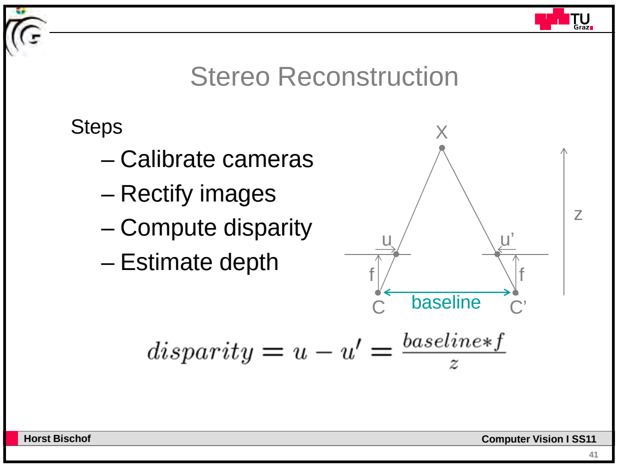
A simple stereo system

$$\boxed{Z = f\frac{T}{d}}$$

- Depth is inversely proportional to disparity



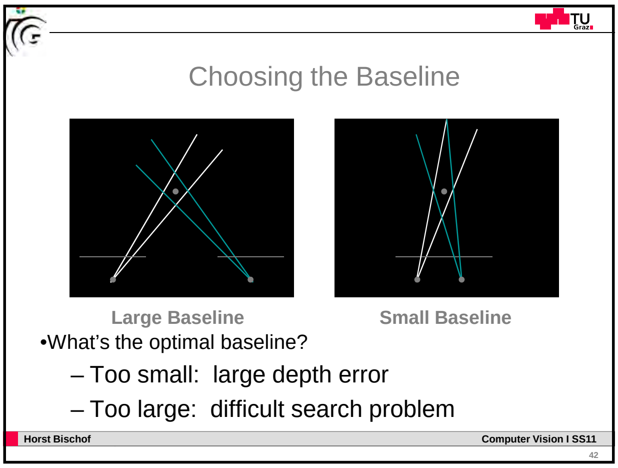
Disparity



Stereo Reconstruction

Steps
- Calibrate cameras
- Rectify images
- Compute disparity
- Estimate depth

$$disparity = u - u' = \frac{baseline * f}{z}$$

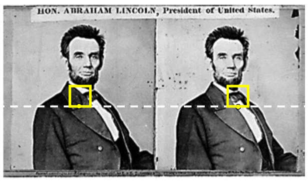


Choosing the Baseline

Large Baseline Small Baseline

- What's the optimal baseline?
  - Too small: large depth error
  - Too large: difficult search problem

Your basic stereo algorithm



For each epipolar line

  For each pixel in the left image

  - compare with every pixel on same epipolar line in right image
  - pick pixel with minimum match cost

Improvement: match ***windows***

- This should look familar...

The Correspondence Problem

Basic assumptions:

- Most scene points are visible in both images
- Corresponding image regions are similar

These assumptions hold if:

- The distance of the fixation point from the cameras is much larger than the stereo baseline: Z >> T

The Correspondence Problem

Is a “search” problem:

- Given an element in the left image, search for the corresponding element in the right image.

We must choose:

- Elements to match
- A similarity measure to compare elements

Correspondence Problem

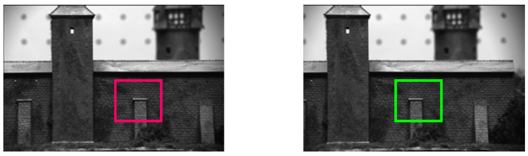
Two classes of algorithms:

- Correlation-based algorithms
  - Produce a DENSE set of correspondences
- Feature-based algorithms
  - Produce a SPARSE set of correspondences

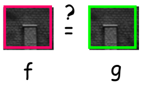
Correlation-based Algorithms

Elements to be matched:

- image WINDOWS of fixed size.



Comparing Windows:



Some possible measures:

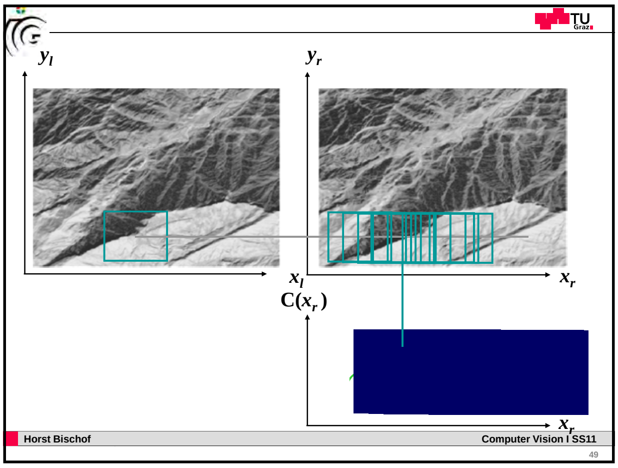
$$\max_{[i,j]\in R} |f(i,j) - g(i,j)|$$

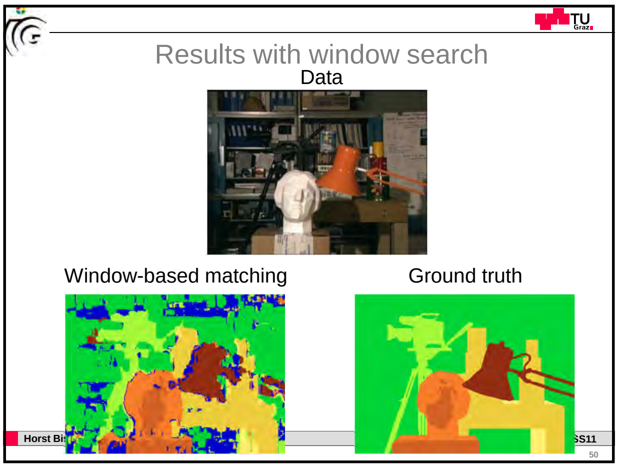
$$\sum_{[i,j]\in R} |f(i,j) - g(i,j)|$$

$$SSD = \sum_{[i,j]\in R} (f(i,j) - g(i,j))^2$$

$$C_{fg} = \sum_{[i,j]\in R} f(i,j)g(i,j)$$

Most popular



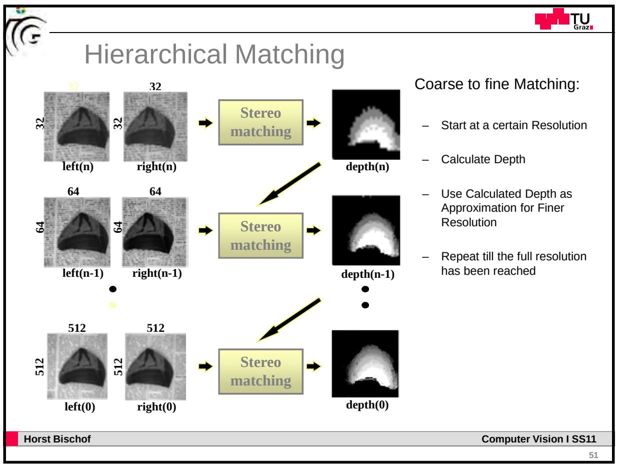


Results with window search

Data

Window-based matching

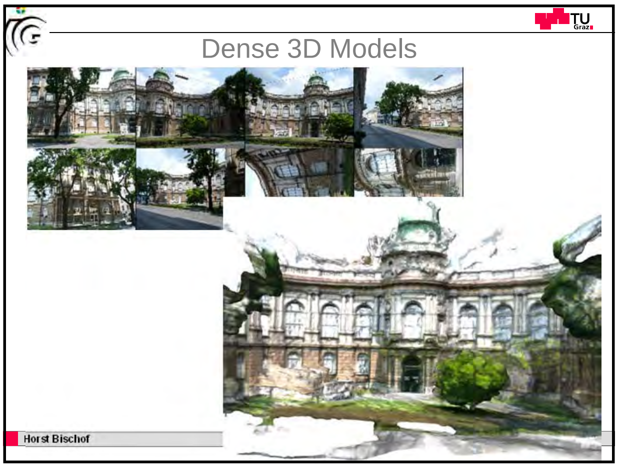
Ground truth



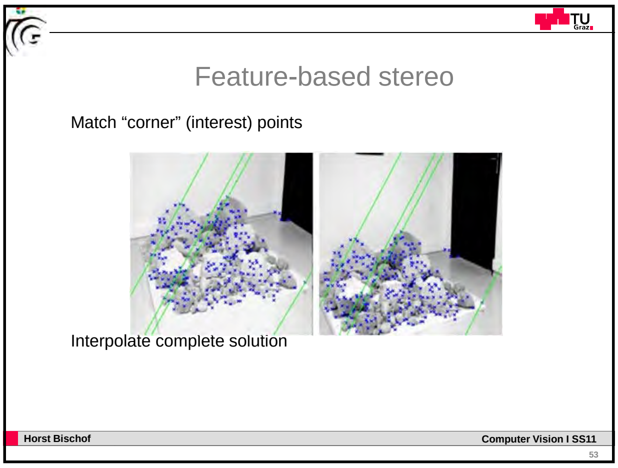
Hierarchical Matching

Coarse to fine Matching:

- Start at a certain Resolution
- Calculate Depth
- Use Calculated Depth as Approximation for Finer Resolution
- Repeat till the full resolution has been reached



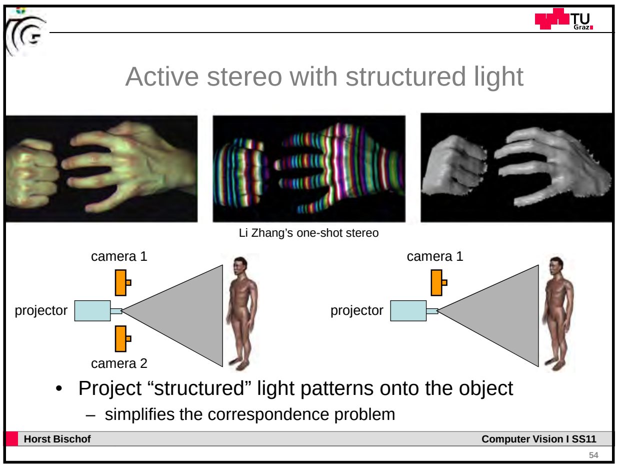
Dense 3D Models



Feature-based stereo

Match “corner” (interest) points

Interpolate complete solution

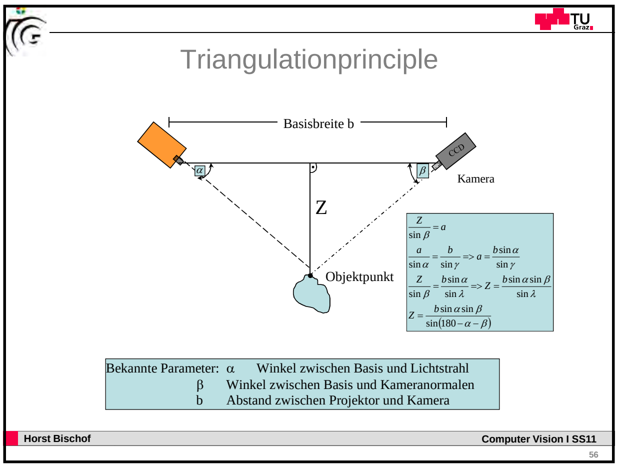


Active stereo with structured light

Li Zhang's one-shot stereo

- Project “structured” light patterns onto the object
  - simplifies the correspondence problem

Microsoft Kinect



Triangulationprinciple

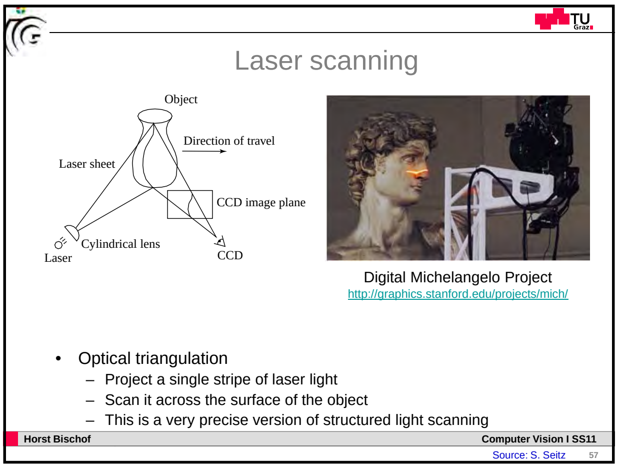
$$\frac{Z}{\sin\beta} = a$$

$$\frac{a}{\sin\alpha} = \frac{b}{\sin\gamma} => a = \frac{b\sin\alpha}{\sin\gamma}$$

$$\frac{Z}{\sin\beta} = \frac{b\sin\alpha}{\sin\lambda} => Z = \frac{b\sin\alpha\sin\beta}{\sin\lambda}$$

$$Z = \frac{b\sin\alpha\sin\beta}{\sin(180-\alpha-\beta)}$$

Bekannte Parameter: α Winkel zwischen Basis und Lichtstrahl
β Winkel zwischen Basis und Kameranormalen
b Abstand zwischen Projektor und Kamera

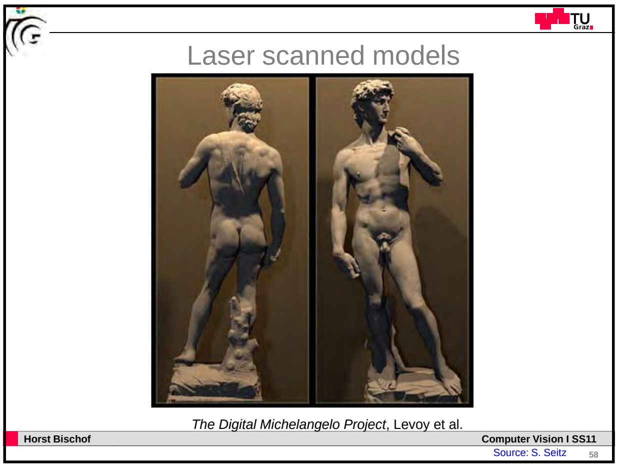


Laser scanning

Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/

- Optical triangulation
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning

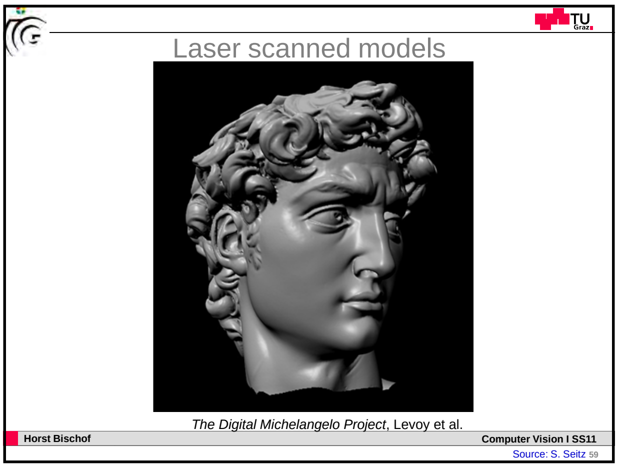
Source: S. Seitz



Laser scanned models

The Digital Michelangelo Project, Levoy et al.

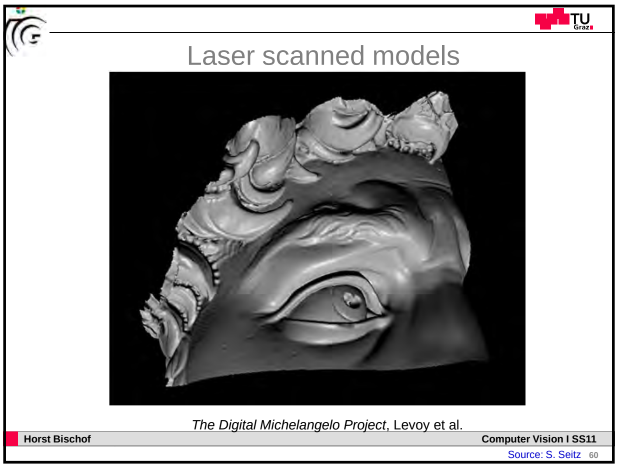
Source: S. Seitz



Laser scanned models

The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz



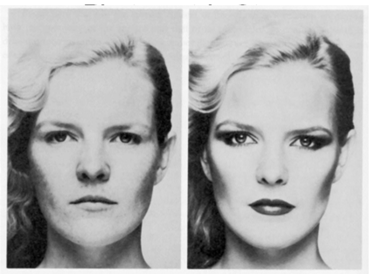
Laser scanned models

The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz

Shape from x

- Profilmessung eines Objektes
- **Shape from Shading**
- Shape from Shadow
- Structured Light
- Shape from Focus
- **Photometric Stereo**
- Interferometry
- Stereopsis
- Tomography

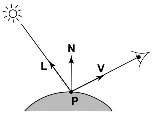


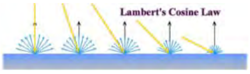
Merle Norman Cosmetics, Los Angeles

Readings

- R. Woodham, *Photometric Method for Determining Surface Orientation from Multiple Images*. Optical Engineering 19(1)139-144 (1980). (PDF)

Diffuse reflection



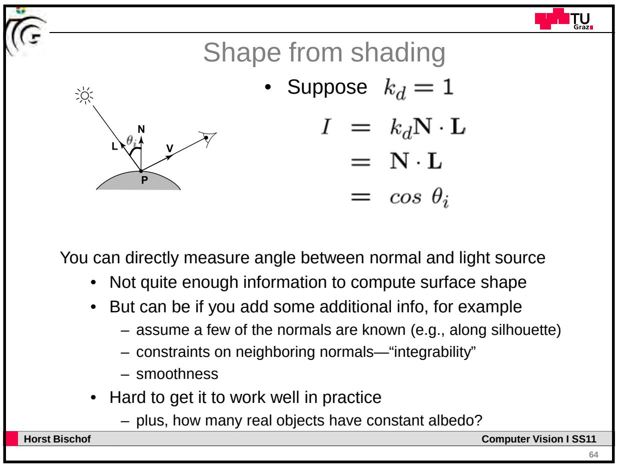


$$R_e = k_d \mathbf{N} \cdot \mathbf{L} R_i$$

image intensity of **P** $\longrightarrow$ $I = k_d \mathbf{N} \cdot \mathbf{L}$

Simplifying assumptions

- I = R_e: camera response function f is the identity function:
  - can always achieve this in practice by solving for f and applying f^{-1} to each pixel in the image
- R_i = 1: light source intensity is 1
  - can achieve this by dividing each pixel in the image by R_i



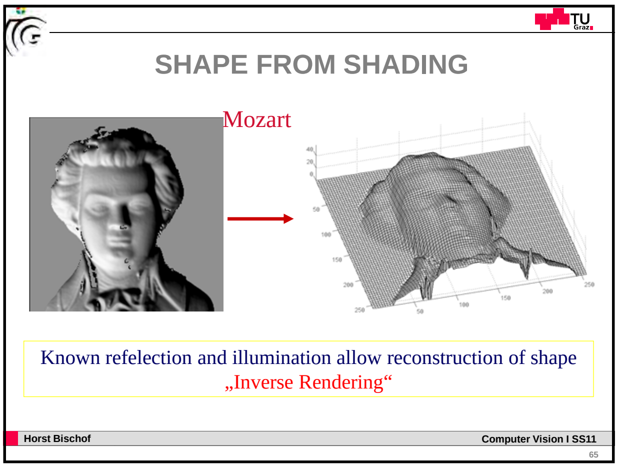
Shape from shading

- Suppose $k_d = 1$

$$\begin{aligned} I &= k_d \mathbf{N} \cdot \mathbf{L} \\ &= \mathbf{N} \cdot \mathbf{L} \\ &= cos\ \theta_i \end{aligned}$$

You can directly measure angle between normal and light source

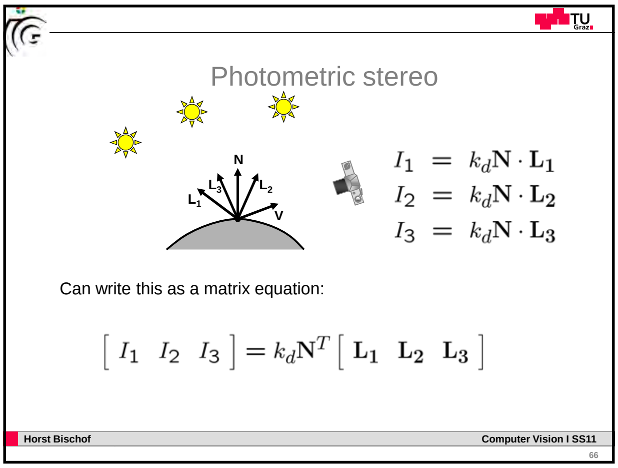
- Not quite enough information to compute surface shape
- But can be if you add some additional info, for example
  - assume a few of the normals are known (e.g., along silhouette)
  - constraints on neighboring normals—“integrability”
  - smoothness
- Hard to get it to work well in practice
  - plus, how many real objects have constant albedo?



SHAPE FROM SHADING

Mozart

Known refelection and illumination allow reconstruction of shape
„Inverse Rendering“



Photometric stereo

$$\begin{aligned} I_1 &= k_d \mathbf{N} \cdot \mathbf{L_1} \\ I_2 &= k_d \mathbf{N} \cdot \mathbf{L_2} \\ I_3 &= k_d \mathbf{N} \cdot \mathbf{L_3} \end{aligned}$$

Can write this as a matrix equation:

$$\left[\begin{array}{ccc} I_1 & I_2 & I_3 \end{array}\right] = k_d \mathbf{N}^T \left[\begin{array}{ccc} \mathbf{L_1} & \mathbf{L_2} & \mathbf{L_3} \end{array}\right]$$

Solving the equations

$$\underbrace{\begin{bmatrix} I_1 & I_2 & I_3 \end{bmatrix}}_{\substack{\mathbf{I} \\ 1\times 3}} = \underbrace{k_d \mathbf{N}^T}_{\substack{\mathbf{G} \\ 1\times 3}} \underbrace{\begin{bmatrix} \mathbf{L_1} & \mathbf{L_2} & \mathbf{L_3} \end{bmatrix}}_{\substack{\mathcal{L} \\ 3\times 3}}$$

$$\mathbf{G} = \mathbf{I}\mathbf{L}^{-1}$$

$$k_d = \|\mathbf{G}\|$$
$$\mathbf{N} = \frac{1}{k_d}\mathbf{G}$$

More than three lights

- Get better results by using more lights

$$\begin{bmatrix} I_1 & \dots & I_n \end{bmatrix} = k_d \mathbf{N}^T \begin{bmatrix} \mathbf{L_1} & \dots & \mathbf{L_n} \end{bmatrix}$$

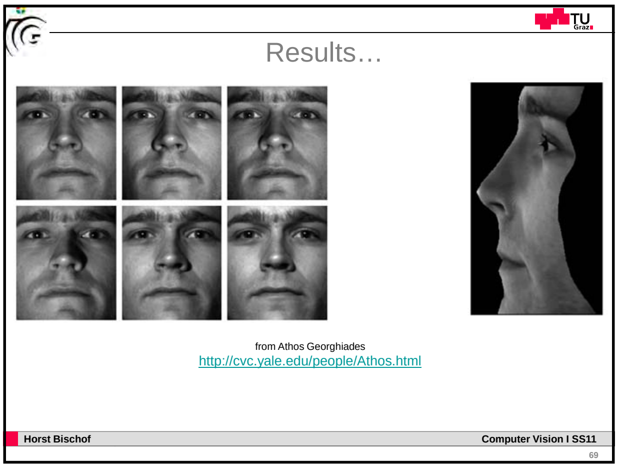
Least squares solution:

$$\begin{aligned} \mathbf{I} &= \mathbf{G}\mathbf{L} \\ \mathbf{I}\mathbf{L}^{\mathbf{T}} &= \mathbf{G}\mathbf{L}\mathbf{L}^{\mathbf{T}} \\ \mathbf{G} &= (\mathbf{I}\mathbf{L}^{\mathbf{T}})(\mathbf{L}\mathbf{L}^{\mathbf{T}})^{-1} \end{aligned}$$

Solve for N, k_d as before

What's the size of **LL**$^{\mathbf{T}}$?

Results…



from Athos Georghiades
http://cvc.yale.edu/people/Athos.html

Limitations

- Big problems
  - doesn’t work for shiny things, semi-translucent things
  - shadows, inter-reflections
- Smaller problems
  - camera and lights have to be distant
  - calibration requirements
    - measure light source directions, intensities
    - camera response function
- Newer work addresses some of these issues

Shape from x

Profilmessung eines Objektes
Shape from Shading
Shape from Shadow
Structured Light
Shape from Focus
Photometric Stereo
Interferometry
Stereopsis
Tomography

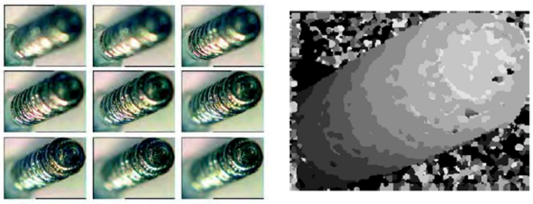
Shape f. Focus/Defocus

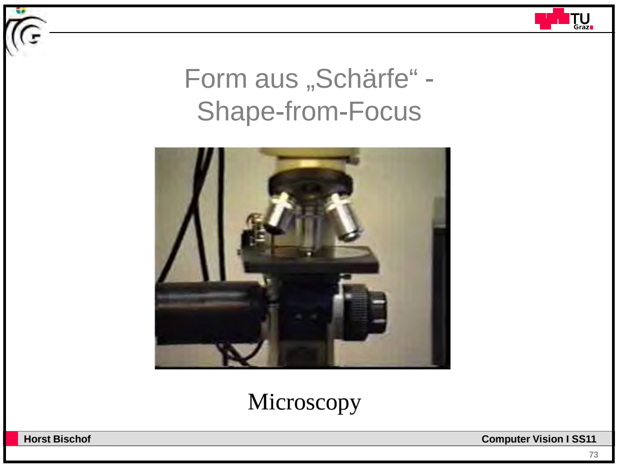
Shape from Focus:

Objekt bewegt entlang x/y/z, bis Oberfläche im Focus
Wie Misst man Focus?

Shape from Defocus:

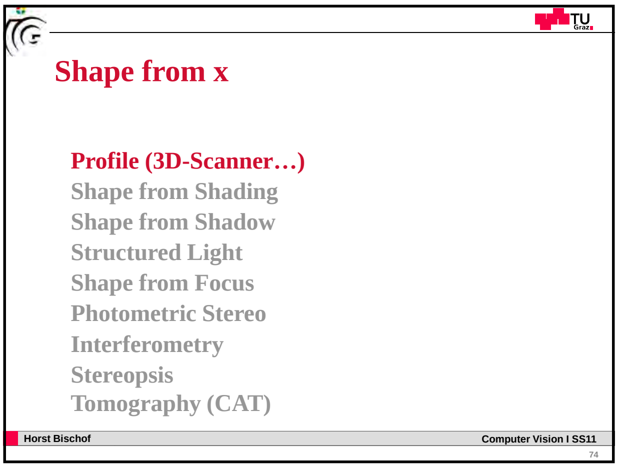
Tiefeninformation durch Blurr (mindestens 2 Focus Einstellungen)





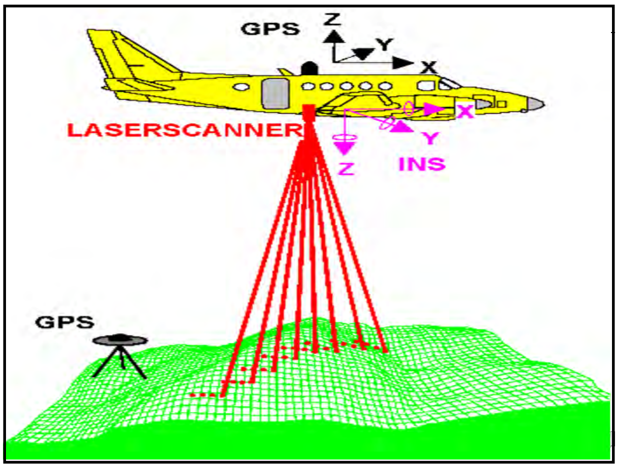
Form aus „Schärfe“ - Shape-from-Focus

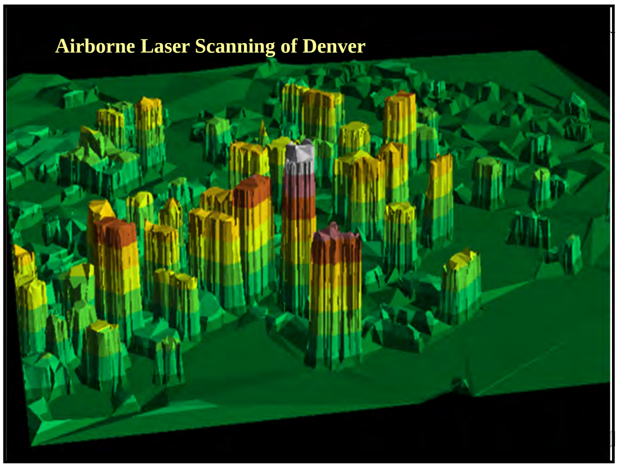
Microscopy



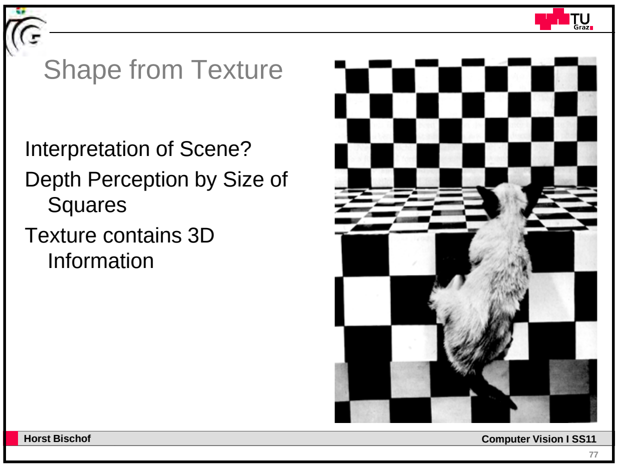
Shape from x

Profile (3D-Scanner…)
Shape from Shading
Shape from Shadow
Structured Light
Shape from Focus
Photometric Stereo
Interferometry
Stereopsis
Tomography (CAT)





Airborne Laser Scanning of Denver



Shape from Texture

- Interpretation of Scene?
- Depth Perception by Size of Squares
- Texture contains 3D Information