3D Scanning

contents

OVERVIEW OF 3D ACQUISITION TECHNIQUES

Overview of 3D Acquisition Techniques

Computer Graphics and Visualization

mechanical measurement of individual points

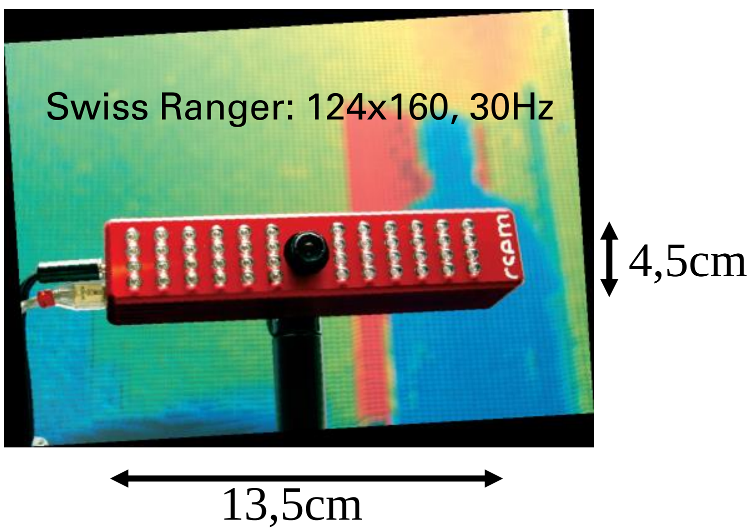
tomographic reconstruction from
x-ray images taken at many view points

Time of flight till detection
of ultrasonic echo

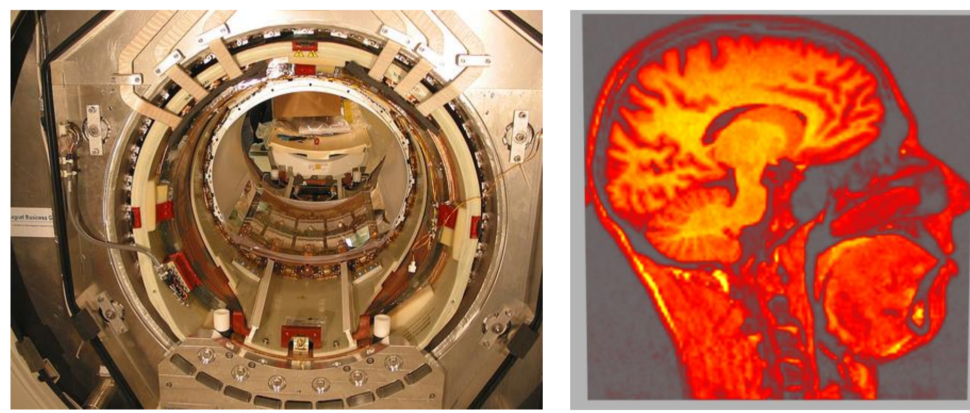
Overview of 3D Acquisition Techniques

precise laser triangulation measurement

spatially varying excitation of hydrogren atoms in magnetic resonance tomography



dynamic phase shift measurement

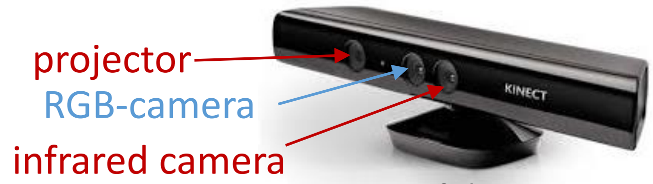


volume visualization of MRI-Volume

RGBD-Cameras

Computer Graphics and Visualization

- today exist affordable 3D cameras from different manufacturers
- plug&play via USB
- joint acquisition of color image and depth map with 30-90 fps
- Hardware architecture:
  - infrared projector projects structured light pattern
  - infrared camera acquires object with projected pattern
  - Reconstruction algorithm computes depth map
  - color camera acquires RGB image



Microsoft kinect

Microsoft kinect 2.0

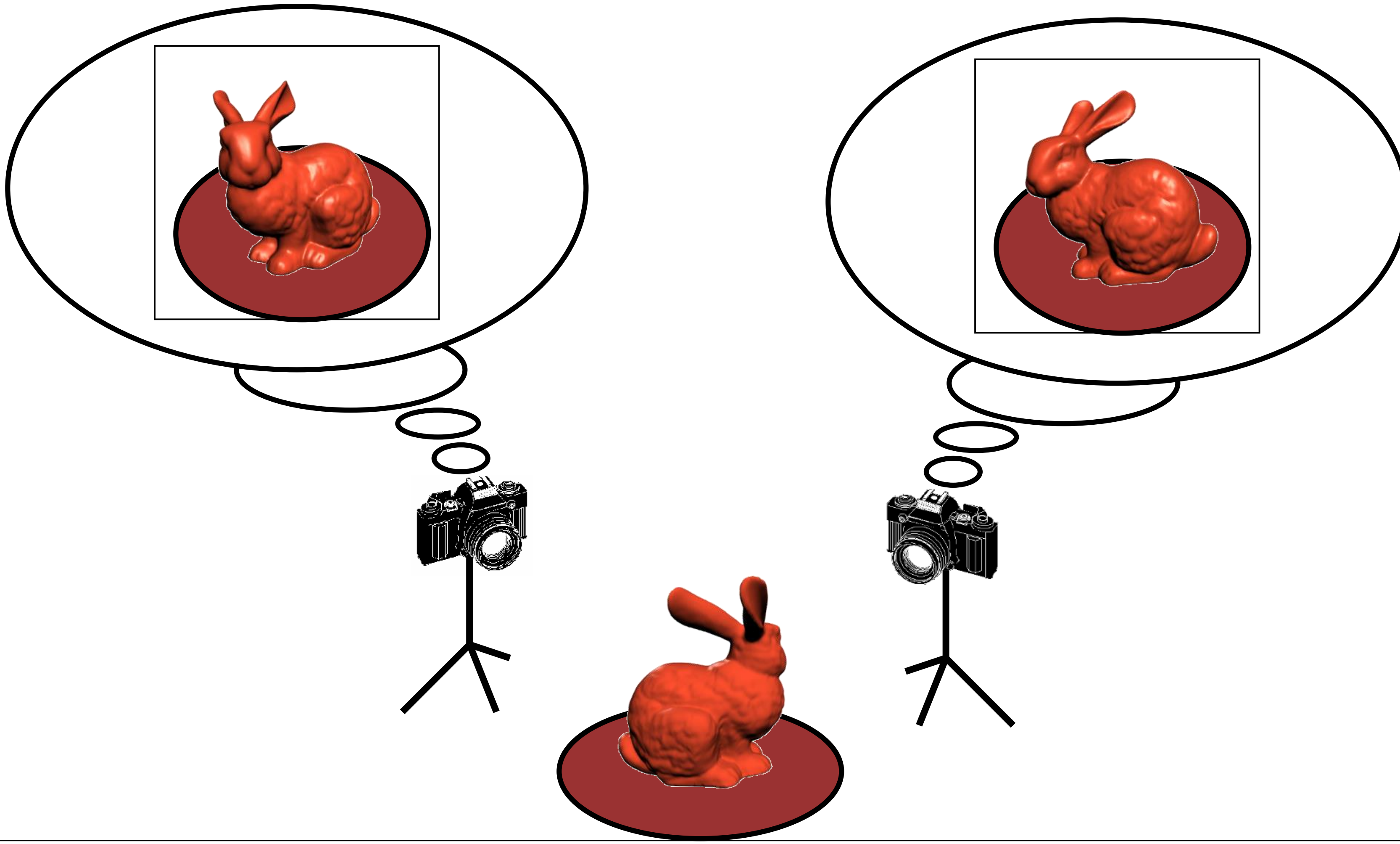
Azure Kinect

Intel® RealSense™ R200

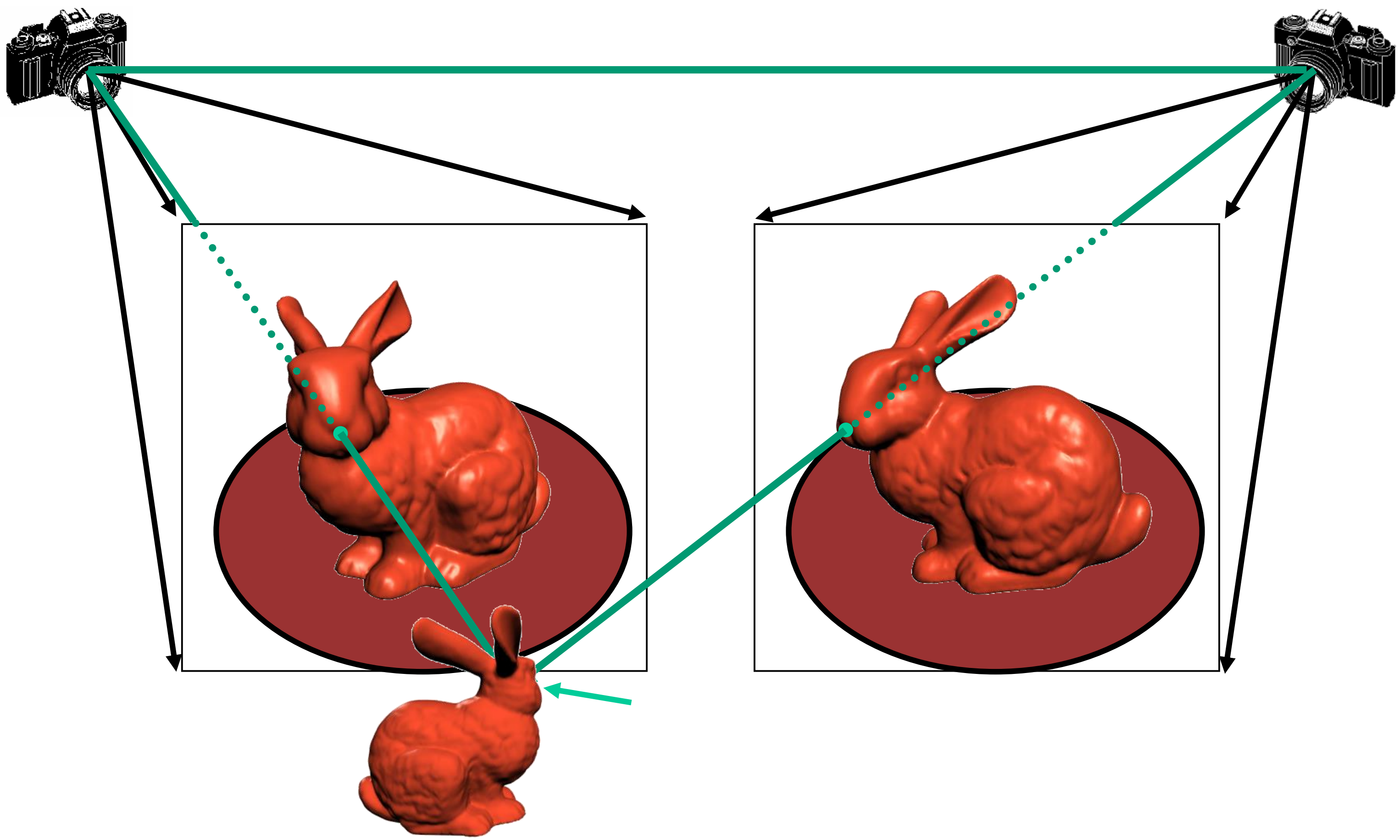
Intel® RealSense™ SR300

Intel® RealSense™ D435i

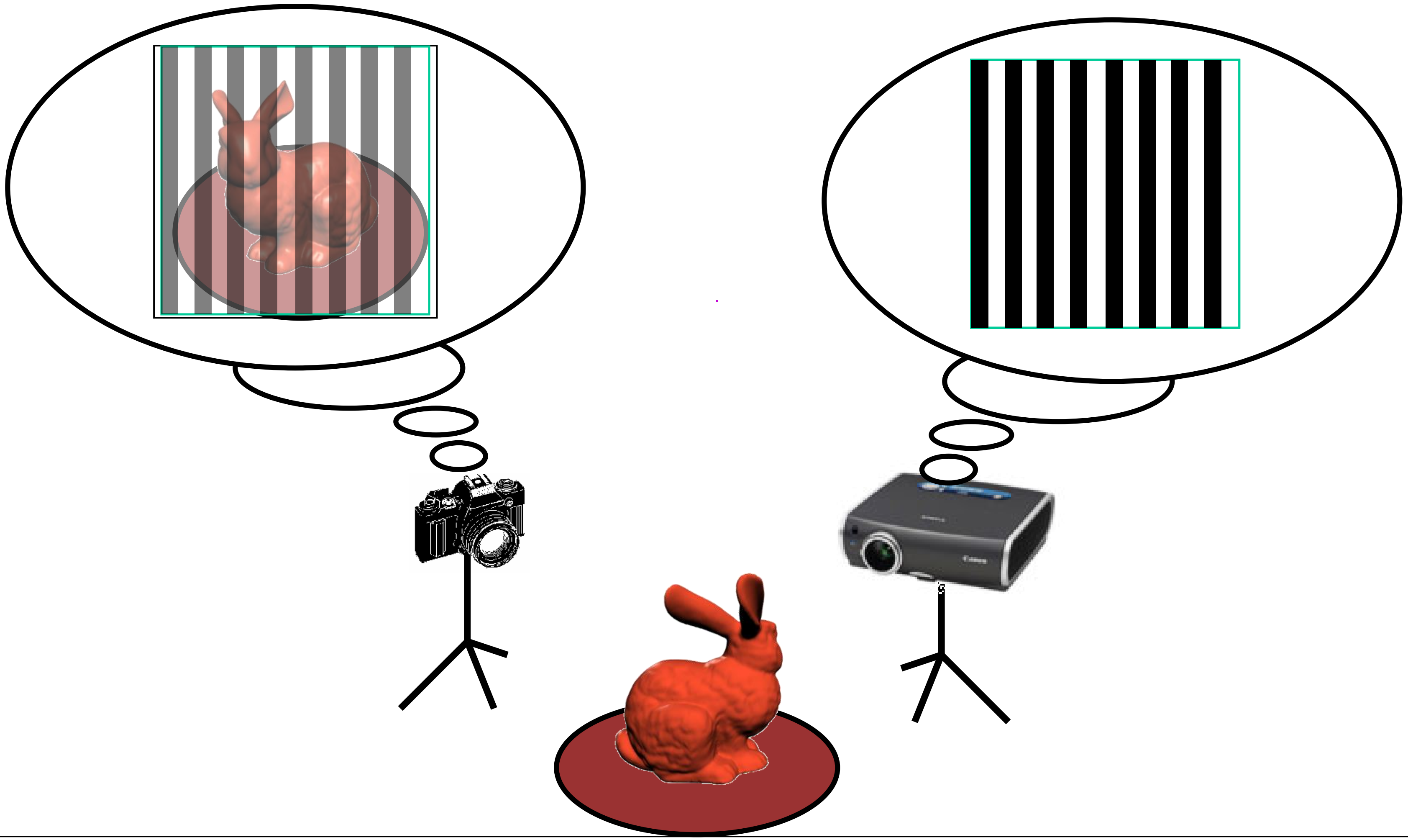
[Multi-view] Stereo Acquisition



system calibration and finding corresponding pixel locations is the basis for 3D point reconstruction via triangulation



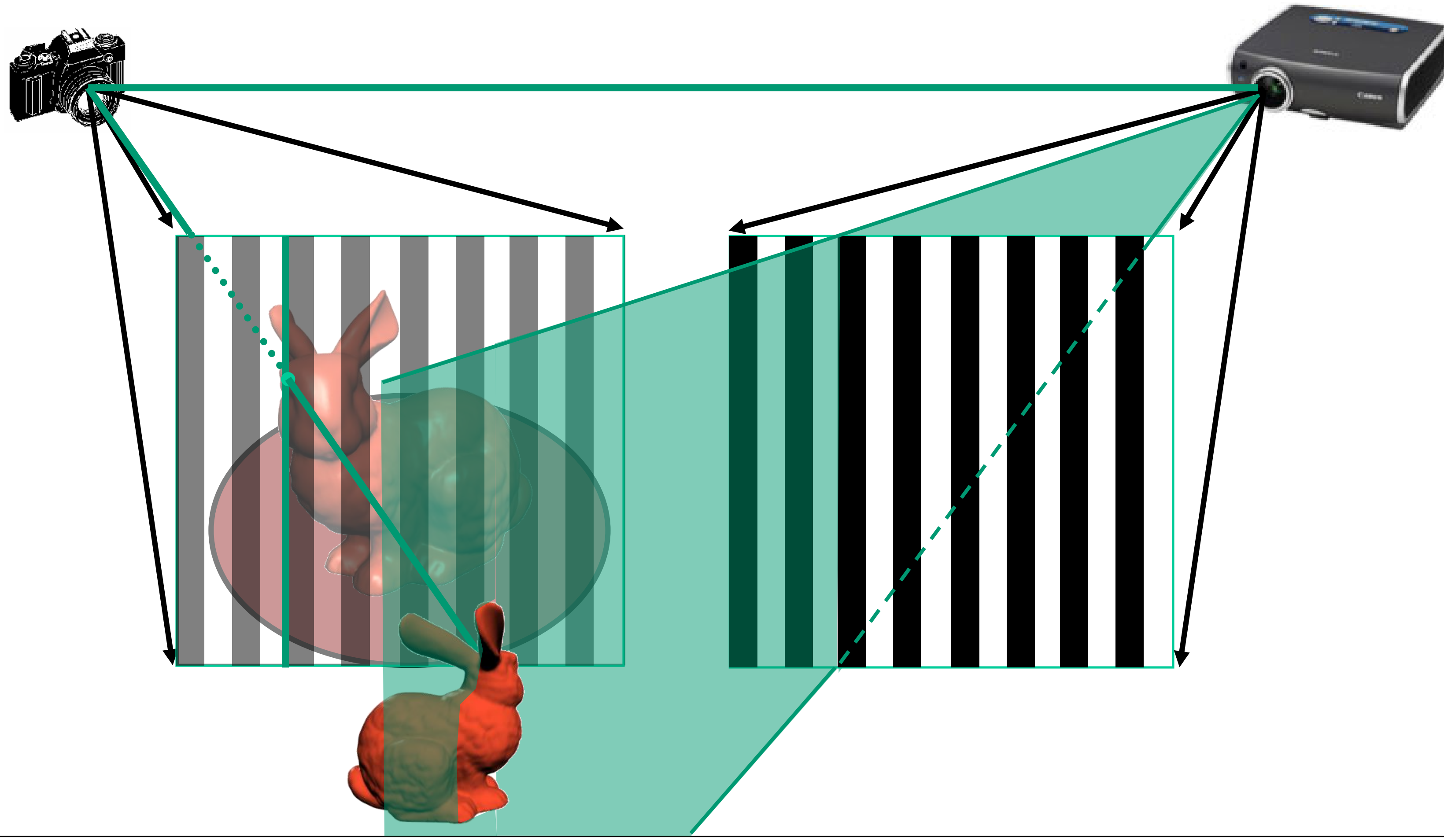
3D Scanning with Structured Light



Overview of 3D Acquisition Techniques

Computer Graphics and Visualization

The projected patterns encode the projector column. For triangulation the ray through the camera pixal is intersected with plane through the projector column.

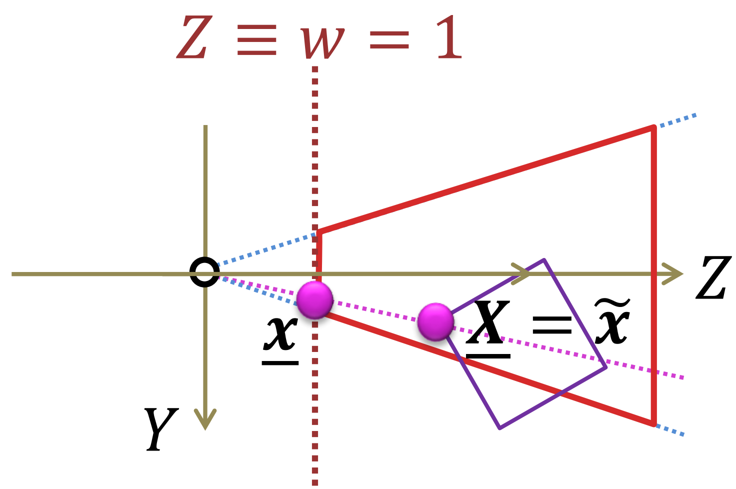


2D PROJECTIVE GEOMETRY AND HOMOGRAPHIES

Camera Projection

Computer Graphics and Visualization

- In computer vision the perspective projection of a pinhole camera is modeled in a coordinate system with the pinhole in the origin and the z-direction corresponding to the view direction (y-direction points downwards) projection to the image plane at $Z = 1$ from division by Z-coordinate (corresponds to w-clip in computer graphics)
- homogeneous image point and 3D point are equal!



$$\underline{\boldsymbol{X}} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \tilde{\boldsymbol{x}}, \qquad \underline{\boldsymbol{x}} = \begin{pmatrix} x = X/Z \\ y = Y/Z \end{pmatrix}$$

2D Projective Geometry

- camera projections map points on a plane with a homography to image plane
- This can be modeled with 2D homogeneous coordinates

$$\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{x}}$$

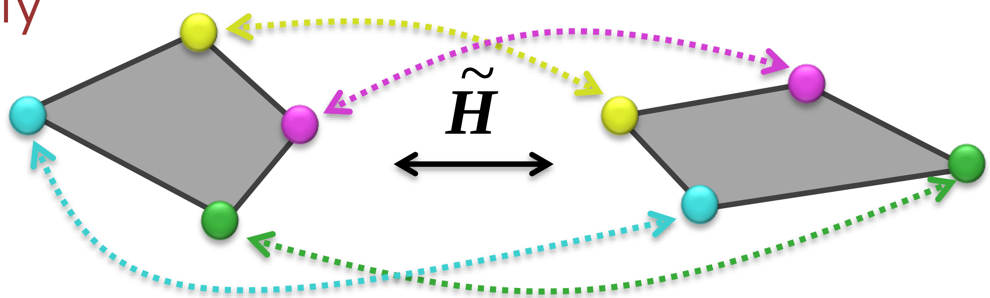
$$\underset{\text{image plane}}{\tilde{\boldsymbol{u}}} = \begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{\lambda} \end{pmatrix} \xleftarrow[\text{homography}]{\tilde{\boldsymbol{H}}} \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{w} \end{pmatrix} = \underset{\text{world plane}}{\tilde{\boldsymbol{x}}}$$

2D Projective Geometry Basics

- a point: $\tilde{\boldsymbol{x}} = (\tilde{x} \quad \tilde{y} \quad \tilde{z})^T \in P^2$
- a line: $a\tilde{x} + b\tilde{y} + c\tilde{w} = 0$
- line as homogeneous vector $\tilde{\boldsymbol{l}} = (a \quad b \quad c)^T \in \bar{P}^2$
- invariance to scalar multiplication $\tilde{\boldsymbol{x}} \sim \lambda \cdot \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{l}} \sim \lambda \cdot \tilde{\boldsymbol{l}}$
- points and lines are dual: $\tilde{\boldsymbol{x}}$ is on $\tilde{\boldsymbol{l}}$ if $\tilde{\boldsymbol{l}}^T\tilde{\boldsymbol{x}}=0$
- line through two points $\tilde{\boldsymbol{l}} = \tilde{\boldsymbol{x}}_1 \times \tilde{\boldsymbol{x}}_2$
- intersection of two lines $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{l}}_1 \times \tilde{\boldsymbol{l}}_2$

2D Projective Transformation

- A homography is defined as a projective transformation that maps from the projective plane to the projective plane bijectively



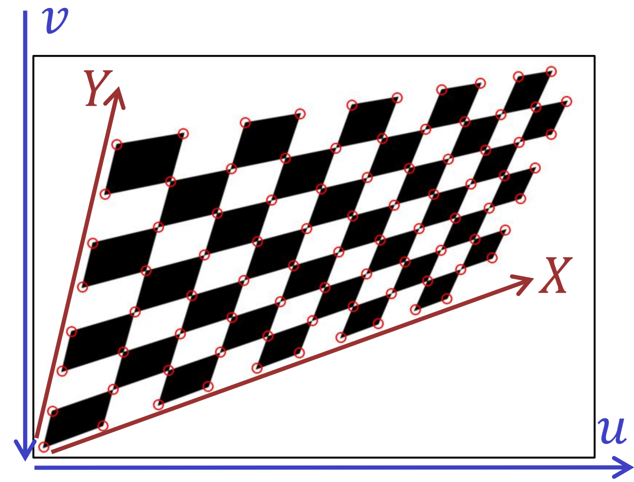
- The homogeneous matrix representation is defined up to scale

$$\tilde{H} \sim \lambda \tilde{H}$$

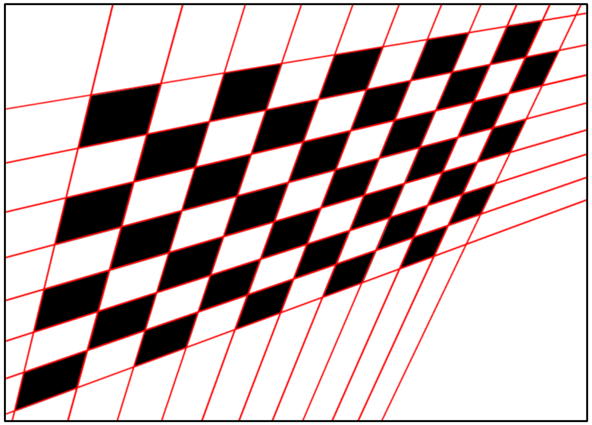
- from 9 parameters, 8 are degrees of freedom (dof)
- 4 corresponding points determine homography
- transformation of lines similar to normals in 3D case: $\tilde{l}' = \left(\tilde{H}^{-1}\right)^T \tilde{l} = \tilde{H}^{-T}\tilde{l}$

Estimation of Homography

- **Input:** point to point $\tilde{\boldsymbol{u}}_i = \lambda_i \cdot \widetilde{\boldsymbol{H}}\tilde{\boldsymbol{x}}_i$ or line to line $\widetilde{\boldsymbol{H}}^T \tilde{\boldsymbol{l}}'_i = \lambda_i \cdot \tilde{\boldsymbol{l}}_i$ correspondences
- at least $n = 4$ correspondences are necessary. *(when mixing point and line correspondences, the case with 2 points & 2 lines is degenerate and does not work)*
- To eliminate λ_i one uses the cross product to derive 3 linear equations in the 9 components of $\widetilde{\boldsymbol{H}}$:
  $\tilde{\boldsymbol{u}}_i \times \widetilde{\boldsymbol{H}}\tilde{\boldsymbol{x}}_i = \vec{\boldsymbol{0}}$ or $\tilde{\boldsymbol{l}}_i \times \widetilde{\boldsymbol{H}}^T \tilde{\boldsymbol{l}}'_i = \vec{\boldsymbol{0}}$.
- per correspondence the 3 equations are linearly dependent and span a 2 dimensional space
- The homogeneous system of $3n$ equations in the 9 components of $\widetilde{\boldsymbol{H}}$ is solved by the singular vector corresponding to the smallest singular value in the SVD.



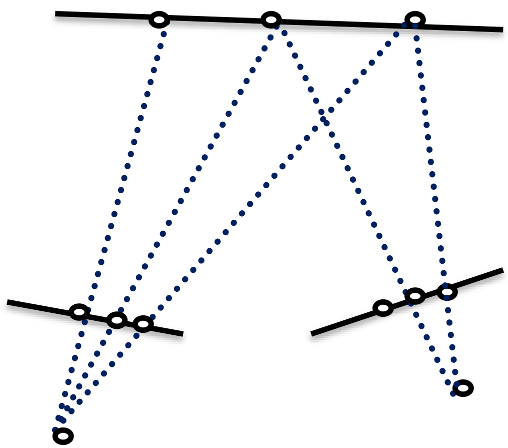
checker board with point features
©Wikipedia



checker board with line features
©Wikipedia

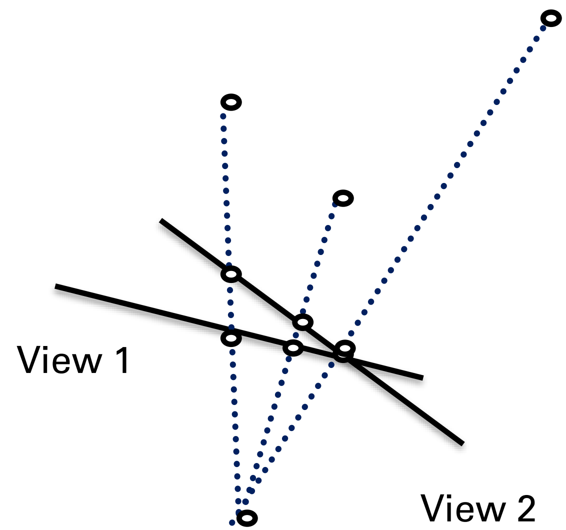
When does Homography relate 2 views?

planar scene homography between 2 views:



View 1 View 2

pure rotation -› same pin hole:



When does Homography relate 2 views?

Computer Graphics and Visualization

Planar Scene

- the part of interest of the scene is planar
- given the plane normal $\hat{\boldsymbol{n}}$ and distance d to origin, one gets the homography from rotation $\boldsymbol{R}$ and translation $\vec{\boldsymbol{t}}$ of camera:

$$\hat{\boldsymbol{n}}^T \underline{\boldsymbol{X}} = d \Rightarrow 1 = \frac{\hat{\boldsymbol{n}}^T \underline{\boldsymbol{X}}}{d}$$

$$\underline{\boldsymbol{X}}' = \boldsymbol{R}\underline{\boldsymbol{X}} + \vec{\boldsymbol{t}} = \underbrace{\left(\boldsymbol{R} + \frac{1}{d}\vec{\boldsymbol{t}}\hat{\boldsymbol{n}}^T\right)}_{\widetilde{\boldsymbol{H}}} \underline{\boldsymbol{X}}$$

$$\widetilde{\boldsymbol{x}}' = \underline{\boldsymbol{X}}' = \widetilde{\boldsymbol{H}}\underline{\boldsymbol{X}} = \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{x}}$$

Pure Rotation

- the camera only rotates around pinhole
- The rotation matrix $\boldsymbol{R}$ can directly be used as homography:

$$\underline{\boldsymbol{X}}' = \boldsymbol{R}\underline{\boldsymbol{X}}$$

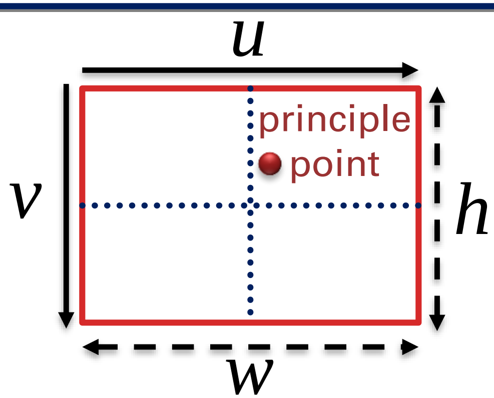
$$\widetilde{\boldsymbol{H}} = \boldsymbol{R}$$

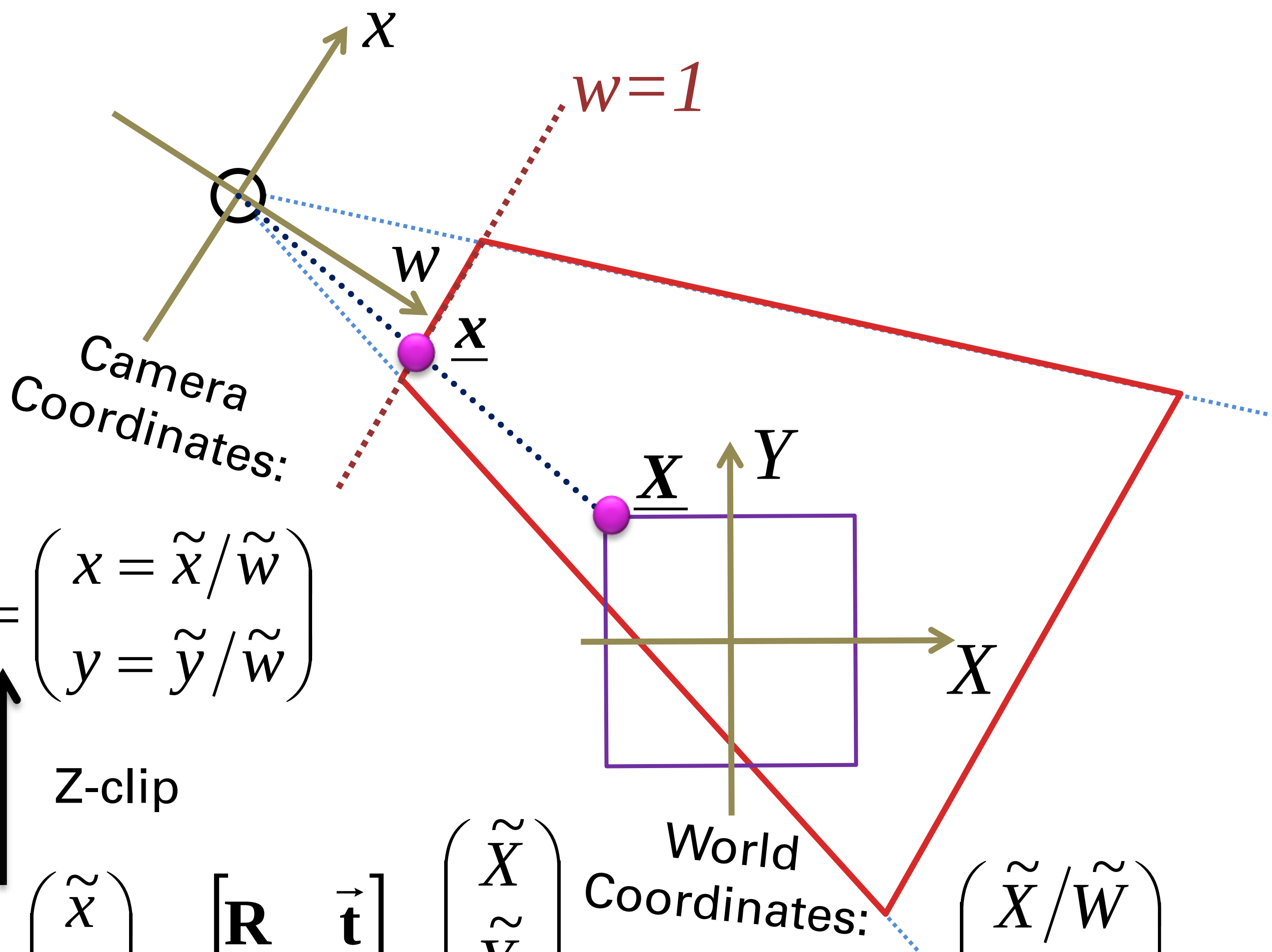
$$\widetilde{\boldsymbol{x}}' = \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{x}}$$

- in both cases the depth of the scene points cannot be recovered from 2 views

GEOMETRIC CAMERA MODEL

Geometric Camera Model





Pixel Coordinates:

$$\underline{\boldsymbol{u}} = \begin{pmatrix} u = \tilde{u}/\tilde{\lambda} \\ v = \tilde{v}/\tilde{\lambda} \end{pmatrix} \qquad \underline{\boldsymbol{x}} = \begin{pmatrix} x = \tilde{x}/\tilde{w} \\ y = \tilde{y}/\tilde{w} \end{pmatrix}$$

Z-clip

Z-clip

$$\tilde{\boldsymbol{u}} = \begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{\lambda} \end{pmatrix} \xleftarrow[\text{intrinsic calibration}]{\tilde{\boldsymbol{K}}_C} \tilde{\boldsymbol{x}} = \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{w} \end{pmatrix} \xleftarrow[\text{extrinsic calibration}]{[\mathbf{R} \quad \vec{\mathbf{t}}]} \begin{pmatrix} \tilde{X} \\ \tilde{Y} \\ \tilde{Z} \\ \tilde{W} \end{pmatrix} = \tilde{\boldsymbol{X}}$$

$$\underline{\boldsymbol{X}} = \begin{pmatrix} \tilde{X}/\tilde{W} \\ \tilde{Y}/\tilde{W} \\ \tilde{Z}/\tilde{W} \end{pmatrix}$$

Extrinsic Parameters

- The camera's position and orientation with respect to the world coordinate system is defined by a 3x3 dim rotation matrix and a 3D translation vector.
- rotation and translation each have 3 degrees of freedom (dof) together these are 6 dof
- Extrinsic calibration determines $\boldsymbol{R}$ and $\vec{\boldsymbol{t}}$ and corresponds to localization of the camera in the scene

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{R} & \vec{\boldsymbol{t}} \end{bmatrix} \tilde{\boldsymbol{X}}$$

Intrinsic Parameters

- intrinsic parameters specify the internal geometry of the camera
- the simplest model is a pinhole camera defined with the camera matrix $\boldsymbol{K}_C$
  - s_x, s_y … focal length in pixel width and height (often assumed to be equal)
  - c_x, c_y … principle point (often close to image center)
  - h … skew strength (often assumed to be zero)
- This results in 3 up to 5 intrinsic parameters
- As last row of $\boldsymbol{K}_C$ is trivial, one can equivalently do the Z-clip on $\tilde{\boldsymbol{x}}$.

$$\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{K}}_C \tilde{\boldsymbol{x}}$$

$$\tilde{\boldsymbol{K}}_C = \begin{pmatrix} s_x & h & c_x \\ 0 & s_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$\underline{\boldsymbol{u}} = \boldsymbol{K}_C \begin{pmatrix} \underline{\boldsymbol{x}} \\ 1 \end{pmatrix}, \boldsymbol{K}_C = \begin{pmatrix} s_x & h & c_x \\ 0 & s_y & c_y \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} s_x x + hy + c_x \\ s_y y + c_y \end{pmatrix}$$

Focal length

- computation of focal length from metric quantities:

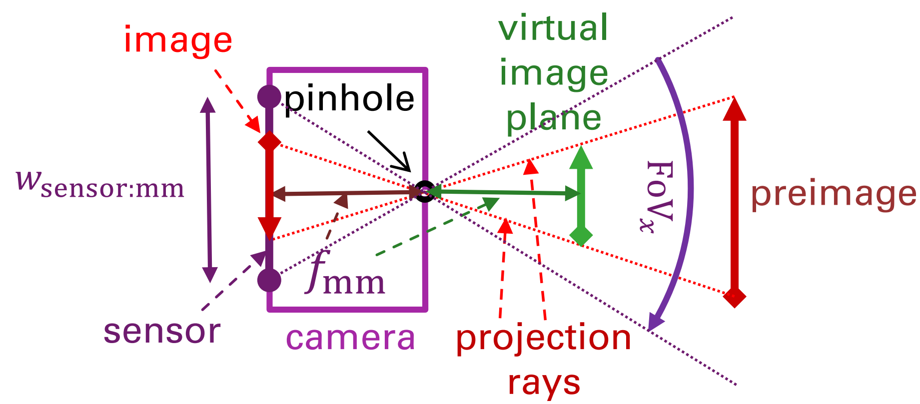
$$s_x = \frac{w_{\text{pixel}}}{w_{\text{sensor:mm}}} f_{\text{mm}}$$

$$s_y = \frac{h_{\text{pixel}}}{h_{\text{sensor:mm}}} f_{\text{mm}}$$

- computation of focal length from field of view and pixel quantities:

$$s_x = \frac{w_{\text{pixel}}}{2\tan\frac{\text{FoV}_x}{2}}$$

$$s_y = \frac{h_{\text{pixel}}}{2\tan\frac{\text{FoV}_y}{2}}$$



- values from kinect 1/RGB:

$$w_{\text{pixel}} \times h_{\text{pixel}} = 640 \times 480$$
$$\text{FoV}_x \times \text{FoV}_y = 62° \times 48.6°$$
$$s_x = 526.37013657$$
$$s_y = 526.37013657$$
$$c_x = 313.68782938$$
$$c_y = 259.01834898$$

Dissecting the Camera Matrix

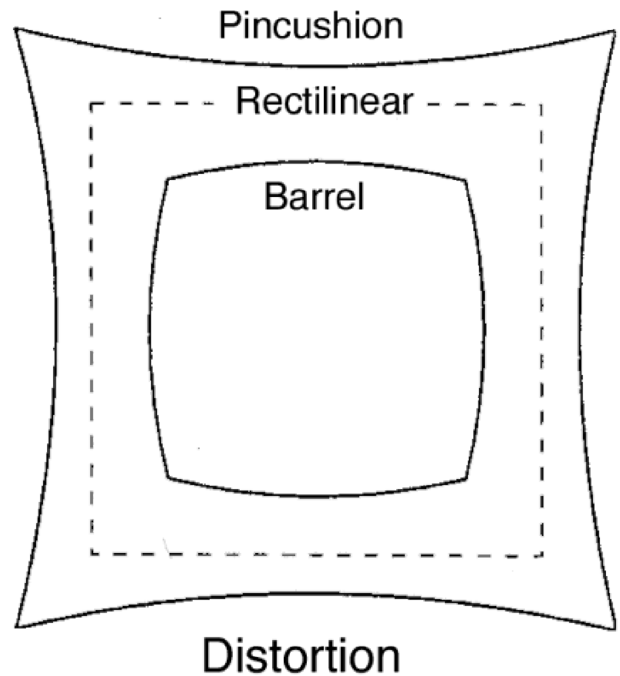
- slight change of notation: $K = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$
- Intrinsic and extrinsic

$$P = \overbrace{K}^{\text{Intrinsic Matrix}} \times \overbrace{[R \mid \mathbf{t}]}^{\text{Extrinsic Matrix}}$$

$$= \overbrace{\underbrace{\begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{2D Translation}} \times \underbrace{\begin{pmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{2D Scaling}} \times \underbrace{\begin{pmatrix} 1 & s/f_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{2D Shear}}}^{\text{Intrinsic Matrix}} \times \overbrace{\underbrace{\left(\begin{array}{c|c} I & \mathbf{t} \end{array} \right)}_{\text{3D Translation}} \times \underbrace{\left(\begin{array}{c|c} R & 0 \\ \hline 0 & 1 \end{array} \right)}_{\text{3D Rotation}}}^{\text{Extrinsic Matrix}}$$

- taken from http://ksimek.github.io/2013/08/13/intrinsic
(see also the interactive tool there)

Lens Distortion



Brown-Conrady Non-linear Lens Distortion Model

- extend linear intrinsic camera model by non-linear radial and tangential distortion model
- map from distorted image coordinates $\underline{x}_d$ to undistorted ones $\underline{x}$ with radial / tangential parameters $k_{1\ldots 6}$ / $p_{1/2}$:

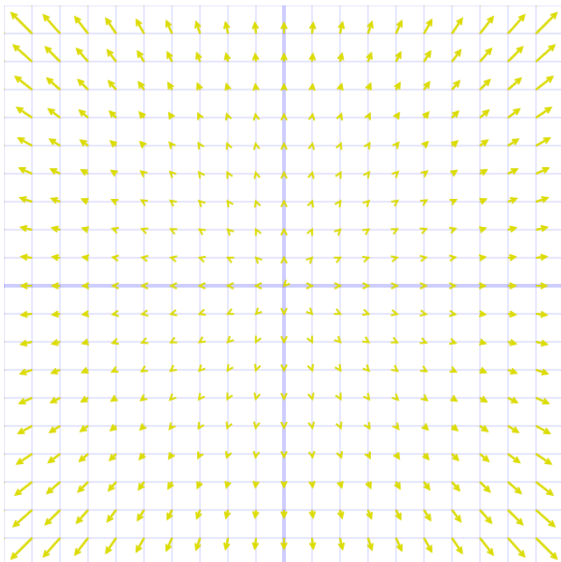
$$\underline{u} \xleftarrow[\text{intrinsic calibration}]{K_C} \underline{x} \xleftarrow[\text{non-lin. lens distortion}]{\boldsymbol{L}} \underline{x}_d \xleftarrow[\text{projection}]{\text{Z-clip}} \tilde{x}_d \xleftarrow[\text{extrinsic calibration}]{\left[\boldsymbol{R}\;\; \vec{t}\right]} \tilde{X}$$

$$\underline{x} = \underline{L}(\underline{x}_d) = \underline{L}_{\mathrm{rad}}(\underline{x}_d) + \vec{L}_{\tan}(\underline{x}_d), \text{ with } r_d^2 = x_d^2 + y_d^2$$

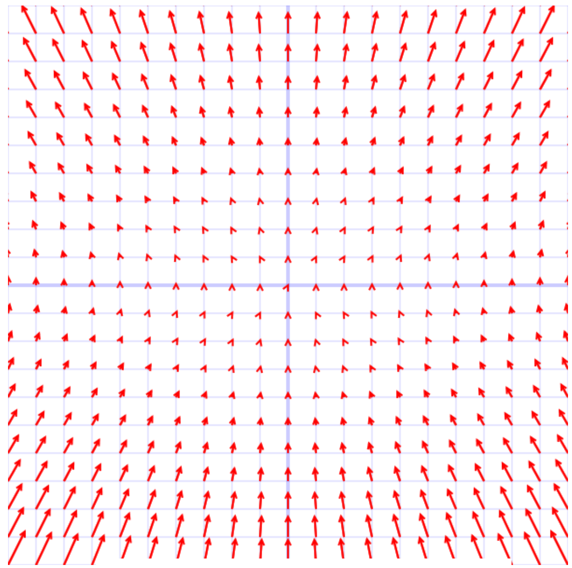
$$\underline{L}_{\mathrm{rad}}(\underline{x}_d) = \frac{1 + k_1 \cdot r_d^2 + k_2 \cdot r_d^4 + k_3 \cdot r_d^6}{1 + k_4 \cdot r_d^2 + k_5 \cdot r_d^4 + k_6 \cdot r_d^6} \cdot \underline{x}_d$$

$$\vec{L}_{\tan}(\underline{x}_d) = p_1 \begin{pmatrix} 2x_d y_d \\ r_d^2 + 2y_d^2 \end{pmatrix} + p_2 \begin{pmatrix} 2x_d^2 + r_d^2 \\ 2x_d y_d \end{pmatrix}$$

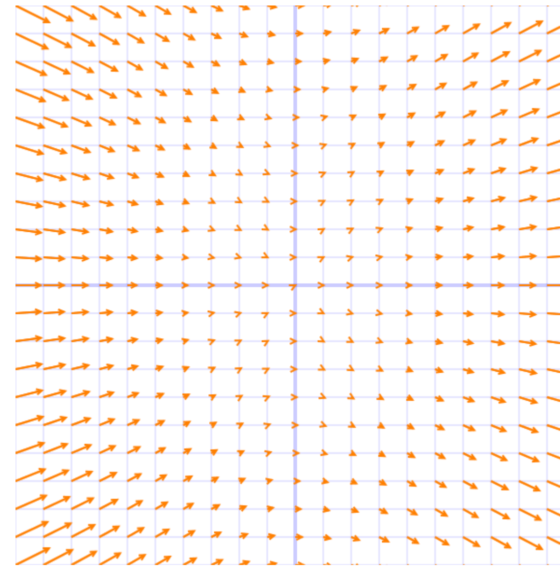
Influence of Lens Distorsion Parameters



$$k_1 : r_d^2 \cdot \underline{\mathbf{x}}_d$$



$$p_1 : \begin{pmatrix} 2x_d y_d \\ r_d^2 + 2y_d^2 \end{pmatrix}$$



$$p_2 : \begin{pmatrix} 2x_d^2 + r_d^2 \\ 2x_d y_d \end{pmatrix}$$

click on graphs for link to vector field plot tool by Kevin Mehall (©2010)

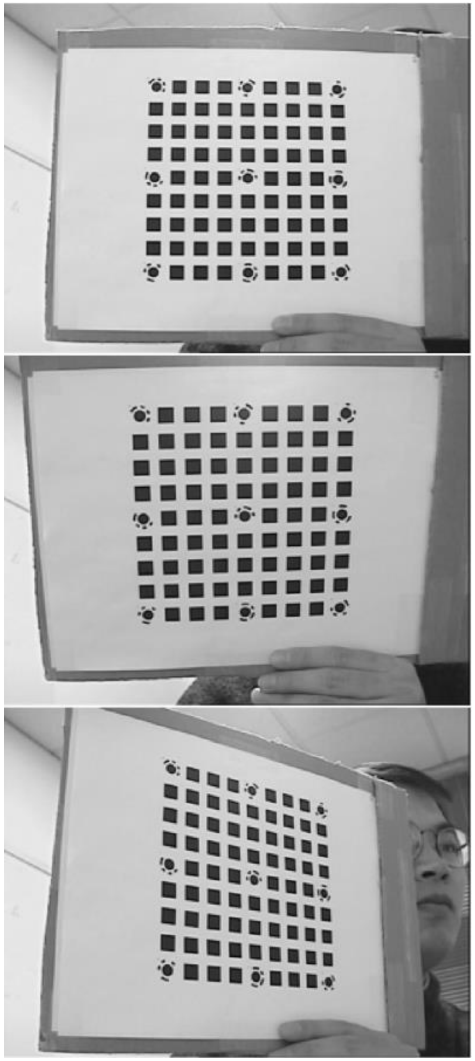
or copy the following link for specific plots:

http://kevinmehall.net/p/equationexplorer/vectorfield.html#!2xyi+%28x^2+3y^2%29j|!%283*x^2+y^2%29i+2xyj|%28x^2+y^2%29*%28xi+yj%29|[-1,1,-1,1]

CAMERA CALIBRATION

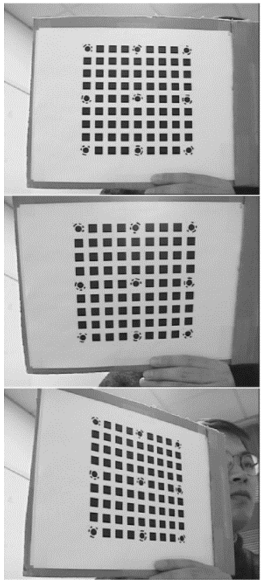
Camera Calibration by Zhang

1. construct calibration plate: print a checker board pattern and attach it to a planar surface.
2. take ($\geq$ 3) images of calibration plate under different orientations by moving either the plate or the camera (no pure translation!!).
3. detect the feature points in the images.
4. estimate the five intrinsic parameters and all the extrinsic parameters using the closed-form solution from paper
5. refine all parameters, including lens distortion parameters, by minimizing re-projection error



sample calibration images taken from Zhang 2000

Graphical Abstract Camera Calibration



$(\tilde{\boldsymbol{u}}_{1j}, \underline{\boldsymbol{X}}_{1j}) \Rightarrow \tilde{\boldsymbol{H}}_1 \Rightarrow$

$(\tilde{\boldsymbol{u}}_{2j}, \underline{\boldsymbol{X}}_{2j}) \Rightarrow \tilde{\boldsymbol{H}}_2 \Rightarrow$

$(\tilde{\boldsymbol{u}}_{3j}, \underline{\boldsymbol{X}}_{3j}) \Rightarrow \tilde{\boldsymbol{H}}_3 \Rightarrow$

$$\left.\begin{array}{l}
0 = \tilde{\boldsymbol{h}}_{1,1}^T \boldsymbol{B} \tilde{\boldsymbol{h}}_{1,2} \\
0 = \tilde{\boldsymbol{h}}_{1,1}^T \boldsymbol{B} \tilde{\boldsymbol{h}}_{1,1} - \tilde{\boldsymbol{h}}_{1,2}^T \boldsymbol{B} \tilde{\boldsymbol{h}}_{1,2} \\
0 = \tilde{\boldsymbol{h}}_{2,1}^T \boldsymbol{B} \tilde{\boldsymbol{h}}_{2,2} \\
0 = \tilde{\boldsymbol{h}}_{2,1}^T \boldsymbol{B} \tilde{\boldsymbol{h}}_{2,1} - \tilde{\boldsymbol{h}}_{2,2}^T \boldsymbol{B} \tilde{\boldsymbol{h}}_{2,2} \\
0 = \tilde{\boldsymbol{h}}_{3,1}^T \boldsymbol{B} \tilde{\boldsymbol{h}}_{3,2} \\
0 = \tilde{\boldsymbol{h}}_{3,1}^T \boldsymbol{B} \tilde{\boldsymbol{h}}_{3,1} - \tilde{\boldsymbol{h}}_{3,2}^T \boldsymbol{B} \tilde{\boldsymbol{h}}_{3,2}
\end{array}\right\} \boldsymbol{B} \Rightarrow \boldsymbol{K}_{C,0}(\boldsymbol{B}) \Rightarrow \boldsymbol{K}_C$$

corner detection and grid location extraction

homography estimation

homogeneous system of equations used to extract $\boldsymbol{B}$ from singular vector of smallest singular value in SVD

compute internal calibration matrix (slide 32)

optimization of non-linear camera model to minimize reprojection error

Camera Calibration by Zhang

- **Input:** images $I_{i=1\ldots n\geq 3}$
- **Output:** camera matrix $\boldsymbol{K}_C$ and poses $\begin{bmatrix}\boldsymbol{R}_i & \vec{\boldsymbol{t}}_i\end{bmatrix}_{i=1\ldots n}$
- in each image detect checker board corners and construct correspondences $\left(\tilde{\boldsymbol{u}}_{ij}, \underline{\boldsymbol{X}}_{ij}\right)_{j=1\ldots m_i}$
- assume that checker board is in $Z=0$ plane:

$$\begin{pmatrix}u\\v\\1\end{pmatrix} = \tilde{\boldsymbol{u}} \sim \boldsymbol{K}_C[\boldsymbol{R} \quad \vec{\boldsymbol{t}}]\tilde{\boldsymbol{X}} = \boldsymbol{K}_C[\vec{\boldsymbol{r}}_1 \quad \vec{\boldsymbol{r}}_2 \quad \vec{\boldsymbol{r}}_3 \quad \vec{\boldsymbol{t}}]\begin{pmatrix}X\\Y\\0\\1\end{pmatrix}$$

- we get homography $\widetilde{\boldsymbol{H}}$ with resp. to 2D homogeneous $\tilde{\tilde{\boldsymbol{X}}}$:

$$\tilde{\boldsymbol{u}} \sim \boldsymbol{K}_C[\vec{\boldsymbol{r}}_1 \quad \vec{\boldsymbol{r}}_2 \quad \vec{\boldsymbol{t}}]\begin{pmatrix}X\\Y\\1\end{pmatrix} = \widetilde{\boldsymbol{H}}\tilde{\tilde{\boldsymbol{X}}}$$

- for each image compute $\widetilde{\boldsymbol{H}}_i$ from $\tilde{\tilde{\boldsymbol{X}}}_{ij}$.

Camera Calibration by Zhang

Computer Graphics and Visualization

- write homography in columns:

$$\left[\tilde{\boldsymbol{h}}_{1,i} \quad \tilde{\boldsymbol{h}}_{2,i} \quad \tilde{\boldsymbol{h}}_{3,i}\right] = \lambda_i \cdot \boldsymbol{K}_C\left[\vec{\boldsymbol{r}}_{1,i} \quad \vec{\boldsymbol{r}}_{2,i} \quad \vec{\boldsymbol{t}}_i\right]$$

- exploit that columns of $\boldsymbol{R}$ are orthonormal:

$$\vec{\boldsymbol{r}}_{1,i}^T\vec{\boldsymbol{r}}_{2,i} = 0 = \tilde{\boldsymbol{h}}_{1,i}^T\boldsymbol{K}_C^{-T}\boldsymbol{K}_C^{-1}\tilde{\boldsymbol{h}}_{2,i} \qquad (1)$$

$$\vec{\boldsymbol{r}}_{1,i}^T\vec{\boldsymbol{r}}_{1,i} = \vec{\boldsymbol{r}}_{2,i}^T\vec{\boldsymbol{r}}_{2,i} \Rightarrow \tilde{\boldsymbol{h}}_{1,i}^T\boldsymbol{K}_C^{-T}\boldsymbol{K}_C^{-1}\tilde{\boldsymbol{h}}_{1,i} = \tilde{\boldsymbol{h}}_{2,i}^T\boldsymbol{K}_C^{-T}\boldsymbol{K}_C^{-1}\tilde{\boldsymbol{h}}_{2,i} \quad (2)$$

- define symmetric matrix $\boldsymbol{B} = \boldsymbol{K}_C^{-T}\boldsymbol{K}_C^{-1}$ and represent it as 6D vector $\vec{\boldsymbol{b}} = (B_{11}, B_{12}, B_{13}, B_{22}, B_{23}, B_{33})$
- for each homography $\tilde{\boldsymbol{H}}_i$ the constraints (1) and (2) define two linear equations on $\vec{\boldsymbol{b}}$, thus 3 images sufficient
- from $\boldsymbol{B}$ we can reconstruct $\boldsymbol{K}_C$ and poses $\left[\boldsymbol{R}_i \quad \vec{\boldsymbol{t}}_i\right]_{i=1\ldots n}$:

$$c_x = \frac{a_1 = B_{12}B_{23} - B_{13}B_{22}}{a_2 = B_{11}B_{22} - B_{12}^2}, c_y = \frac{a_3 = B_{12}B_{13} - B_{11}B_{23}}{a_2}, \lambda = B_{33} - \frac{B_{13}^2 + c_y \cdot a_3}{B_{11}}$$

$$\boldsymbol{K}_C = \begin{pmatrix} \sqrt{\lambda/B_{11}} & -\sqrt{\lambda B_{12}^2/B_{11}a_2} & c_x \\ 0 & \sqrt{\lambda B_{11}/a_2} & c_y \\ 0 & 0 & 1 \end{pmatrix}, \left[\vec{\boldsymbol{r}}_{1,i} \quad \vec{\boldsymbol{r}}_{2,i} \quad \vec{\boldsymbol{t}}_i\right] = \nu\boldsymbol{K}_C^{-1}\tilde{\boldsymbol{H}}_i, \nu = \frac{1}{\left\|\boldsymbol{K}_C^{-1}\tilde{\boldsymbol{h}}_1\right\|}$$

Camera Calibration by Zhang

- in case of no shearing ($h = 0$) $\boldsymbol{B}$ simplifies to

$$B = \frac{1}{s_x^2 s_y^2}\begin{pmatrix} s_y^2 & 0 & -c_x s_y^2 \\ 0 & s_x^2 & -c_y s_x^2 \\ -c_x s_y^2 & -c_y s_x^2 & c_x^2 s_y^2 + c_y^2 s_x^2 + s_x^2 s_y^2 \end{pmatrix}$$

  such that $B_{12} = 0$ resulting in a linear constraint on the vector $\vec{\boldsymbol{b}}$: $(0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0)\vec{\boldsymbol{b}} = 0$, which can be incorporated into the linear equation system.
- In a final non-linear optimization problem the re-projection error is minimized with Levenberg-Marquardt algorithm and previous result as initial guess for camera matrix and poses:

$$\min_{\boldsymbol{K}_C, [\boldsymbol{R}_i \quad \vec{\boldsymbol{t}}_i]_{i=1\ldots n}} \sum_{ij} \left\| \underline{\boldsymbol{u}}_{ij} - \mathrm{Zclip}\left(\boldsymbol{K}_C [\boldsymbol{R}_i \quad \vec{\boldsymbol{t}}_i] \tilde{\boldsymbol{X}}_{ij}\right) \right\|^2$$

Camera Calibration with OpenCV

1. print a checker board pattern
2. take images
3. detect checker board corners:
```
bool findChessboardCorners(
   InputArray image, Size patternSize,
   OutputArray corners, int flags)
```
4. estimate intrinsic and extrinsic parameters and (undistorted case: fit one homography per image, decompose homographies into joint camera matrix and rotations / translations per image)
5. including lens distortion
```
double calibrateCamera(
   InputArrayOfArrays objectPoints, imagePoints, Size imageSize,
   InputOutputArray cameraMatrix, distCoeffs,
   OutputArrayOfArrays rvecs, tvecs, int flags, TermCriteria crit)
```

PROJECTOR MODEL AND CALIBRATION

Projector Model

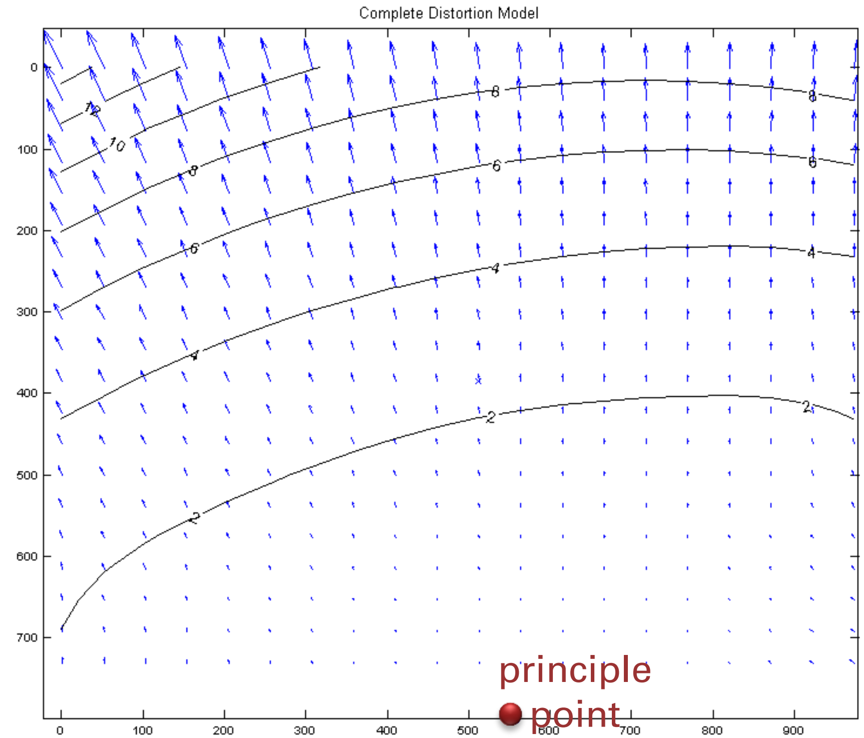
- use pinhole with radial and tangential lens distortion to describe projector

$$\mathbf{K}_P, k_{P,1..6}, p_{P,1..2}$$

- calibrate projector with same technique as camera
- for this we need correspondences of checker board corners j with projector image plane

$$\underline{\boldsymbol{u}}_{P,ij} \leftrightarrow \widetilde{\boldsymbol{X}}_{ij}$$

- calibrate camera first and use it to calibrate projector

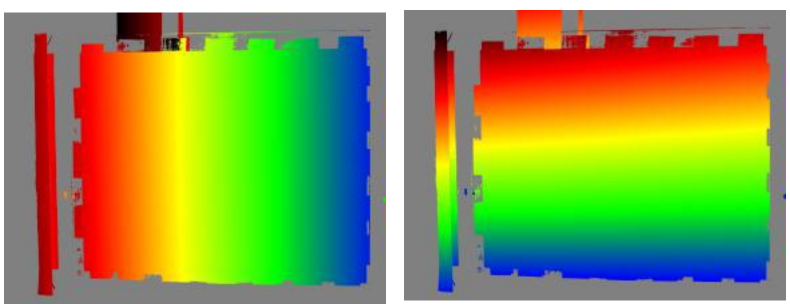


example for projector distortion
taken from D. Moreno and G. Taubin, 2012

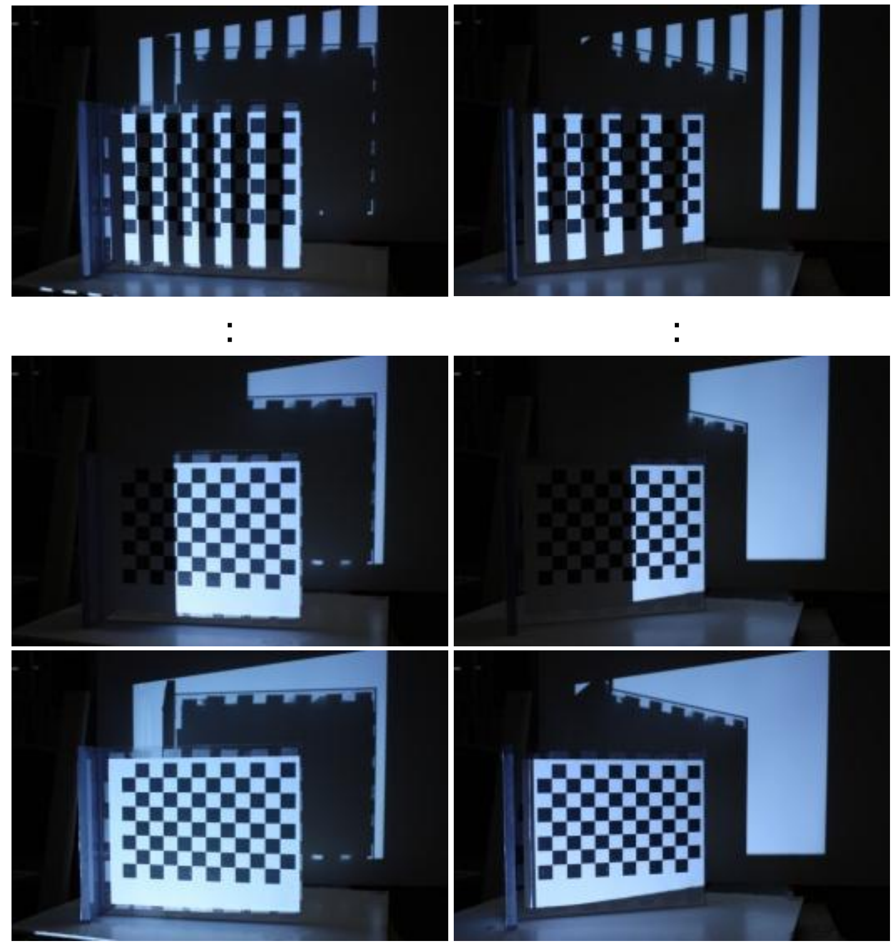
- careful: projectors have principle point on top or bottom edges, not in image center!

Projector to Camera Correspondence

- for each pose i of calibration plate use standard procedure (i.e. gray-code pattern) to encode projector row and column in logarithmic number of images



$u_{P,i}$ $v_{P,i}$



$i=1$ $i=2$ …

all images from D. Moreno and G. Taubin, 2012

Estimate Local Homographies

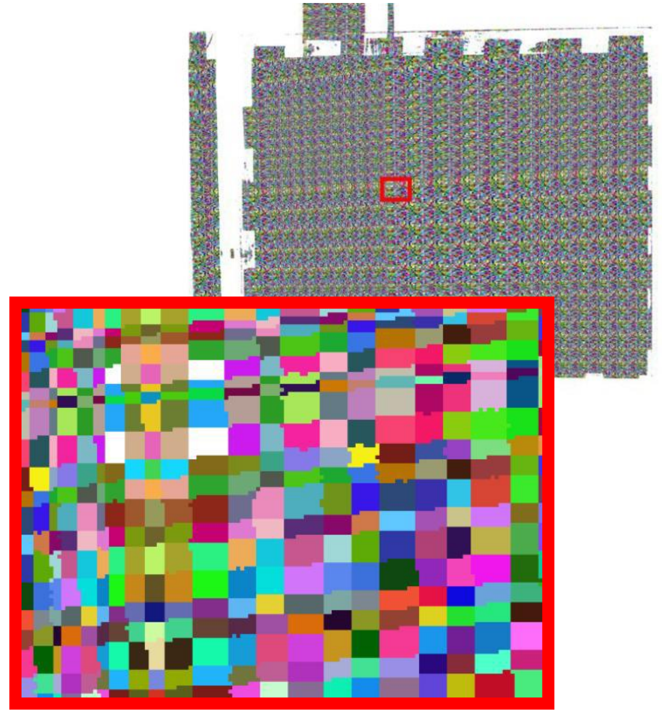
Computer Graphics and Visualization

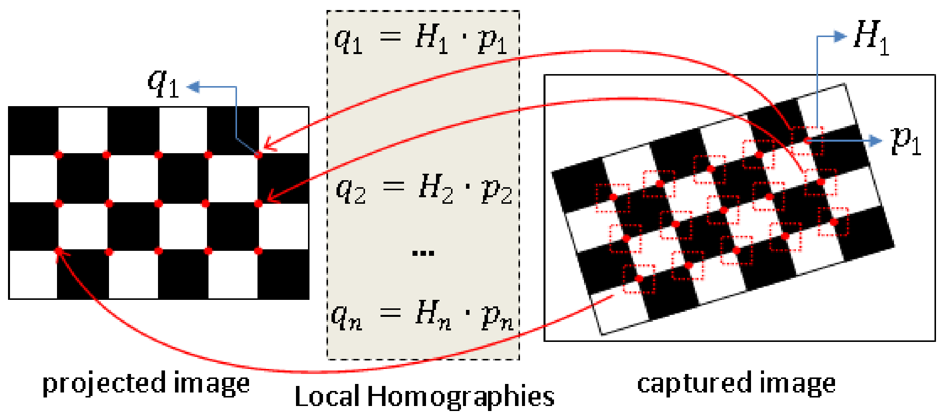
- There are more camera $\underline{\boldsymbol{u}}_C$ pixels than projector pixels $\underline{\boldsymbol{u}}_P$
- per pixel correspondences $\underline{\boldsymbol{u}}_P(\underline{\boldsymbol{u}}_C)$ are not accurate
- therefore perform fit of homography $\tilde{\boldsymbol{H}}_{ij}$ locally around each checker board corner $\underline{\boldsymbol{u}}_{C,ij}$ to a pixel neighborhood $U(\underline{\boldsymbol{u}}_{C,ij})$ of 47x47 pixel:

$$\tilde{\boldsymbol{H}}_{ij} = \min_{\tilde{\boldsymbol{H}}} \sum_{\underline{\boldsymbol{u}}_C \in U(\underline{\boldsymbol{u}}_{C,ij})} \left\| \tilde{\boldsymbol{u}}_P(\underline{\boldsymbol{u}}_C) - \tilde{\boldsymbol{H}}\tilde{\boldsymbol{u}}_C \right\|^2$$

- and finally:

$$\tilde{\boldsymbol{X}}_{ij} \leftrightarrow \tilde{\boldsymbol{u}}_{P,ij} = \tilde{\boldsymbol{H}}_{ij}\tilde{\boldsymbol{u}}_{C,ij}$$





all images from D. Moreno and G. Taubin, 2012

Summary of Camera-Projector Calib

Computer Graphics and Visualization

1. Detect checkerboard corner locations in camera image for each plane orientation
2. Decode projector row and column correspondences
3. Per checkerboard corner in cam image compute local homography (cam image ->proj image)
4. Transform corner locations to projector coordinates
5. Find camera intrinsics with OpenCV's implementation of Zhang's method
6. Find projector intrinsics with OpenCV's implementation of Zhang's method
7. Fix camera and projector intrinsics and use world, camera, and projector corner locations to estimate stereo extrinsic parameters.
8. Optimize all intrinsic and extrinsic parameters to minimize the total re-projection error

CONCLUSION AND REFERENCES

Summary / Take Home Message

- camera and projector can be modeled with pinhole extended by radial and tangential lens distortion
- intrinsic camera parameters can be determined by acquisition of checker board in 3 and more poses
- an iterative non-linear optimization is performed with parameters estimated from linear model as initial guess
- projector can be calibrated in the same way by projecting binary coded stripe images for projector-camera pixel correspondences, for which homographies are fitted locally.

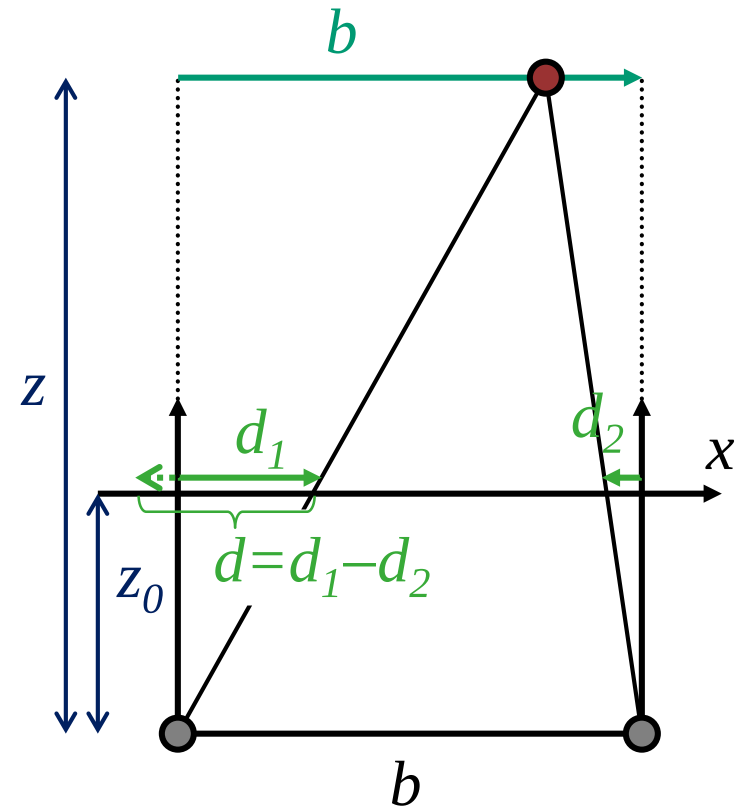
References

- Zhang: A Flexible New Technique for Camera Calibration, TechRep from 1998 and TPAMI **22**(11) 2000
- Daniel Moreno and Gabriel Taubin: Simple, Accurate, and Robust Projector-Camera Calibration, 3DPVT 2012
- OpenCV Reference Manual

TRIANGULATION

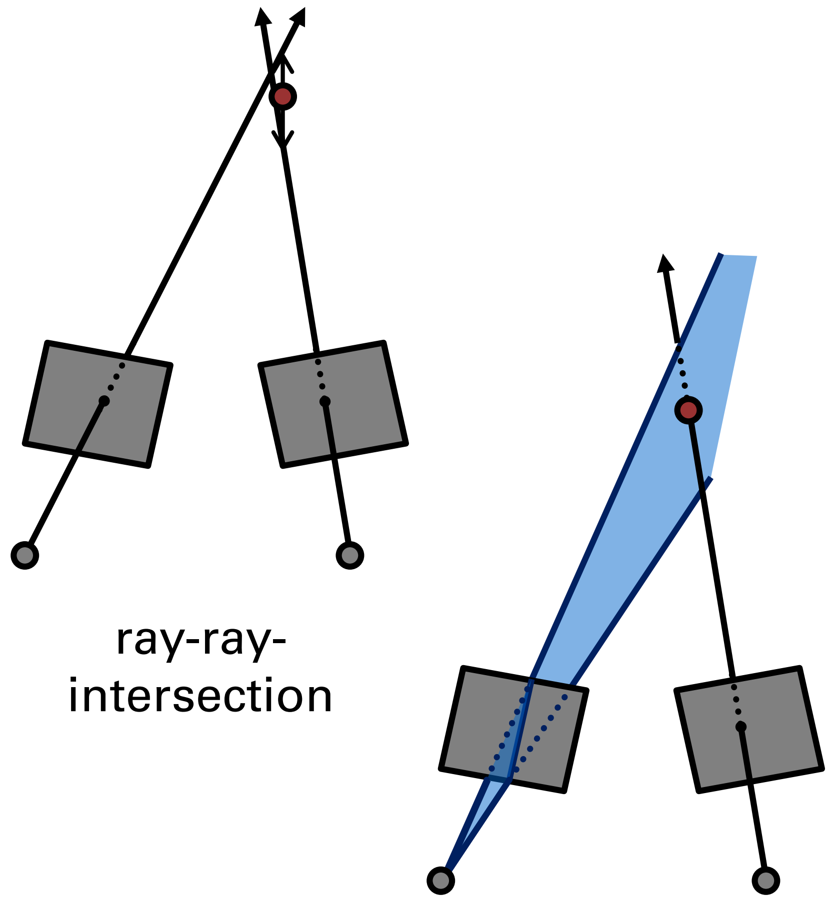
Triangulation – rectified setup

Triangulation in 2D



$$z = \frac{b}{d} z_0 \qquad x = \frac{b}{d} d_1 = b + \frac{b}{d} d_2$$

Triangulation in 3D



ray-ray-intersection

ray-plane-intersection

Triangulation in undistorted Case

$$\underline{u} \xleftarrow{\text{w-clip}} \tilde{u} \xleftarrow[\text{intrinsic}]{\tilde{K}_{C/P}} \tilde{x} \xleftarrow[\text{extrinsic}]{[R_{C/P}|\vec{t}_{C/P}]} \begin{pmatrix} \underline{X} \\ 1 \end{pmatrix} = \tilde{X}$$

Camera

- ray of pixel $\underline{u} = (u, v)$ is line through origin of homogenous vector:

$$\lambda \cdot \tilde{u} = \tilde{K}_C(R_C\underline{X} + \vec{t}_C) \quad (1)$$

- solving for $\underline{X}$ yields ray:

$$\underline{X}_{u,v}(\lambda) = \underline{X}_0 + \lambda \cdot \vec{V},$$
$$\underline{X}_0 = -R_C^T\vec{t}_C \wedge \vec{V} = R_C^T\tilde{K}_C^{-1}\tilde{u}$$

- if camera is coordinate reference, $[R_C|\vec{t}_C] = [\mathbf{1}|\vec{0}]$:

$$\underline{X}_{u,v}(\lambda) = \lambda \cdot \tilde{K}_C^{-1}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

Projector

$$\tilde{l}^T\tilde{u} = 0 \quad (2)$$

2d line equation

- homogenous line of column u_0/row v_0: $\tilde{l} = \begin{pmatrix} -1 \\ 0 \\ u_0 \end{pmatrix} / \begin{pmatrix} 0 \\ -1 \\ v_0 \end{pmatrix}$
- Ansatz for plane

$$\tilde{\Pi}(\tilde{l}) = [R_P|\vec{t}_P]^T\tilde{K}_P^T\tilde{l} \quad (3)$$

- yields plane equation in 3D

$$\tilde{\Pi}^T\tilde{X} = 0$$

- Proof by reduction to line equation:

$$\tilde{\Pi}^T\tilde{X} \overset{(1)}{=} \tilde{l}^T\tilde{K}_P[R_P|\vec{t}_P]\tilde{X} \overset{(3)}{=} \tilde{l}^T\tilde{u} \overset{(2)}{=} 0$$

Triangulation in distorted Case

$$\underline{\boldsymbol{u}} \xleftarrow[\text{intrinsic calibration}]{\boldsymbol{K}_{C/P}} \underline{\boldsymbol{x}} \xleftarrow[\text{non-lin. lens distortion}]{\boldsymbol{L}_{C/P}} \underline{\boldsymbol{x}}_d \xleftarrow[\text{projection}]{\text{w-clip}} \tilde{\boldsymbol{x}}_d \xleftarrow[\text{extrinsic calibration}]{\left[\boldsymbol{R}_{C/P} \quad \vec{\boldsymbol{t}}_{C/P}\right]} \tilde{\boldsymbol{X}}$$

Camera

- $\underline{\boldsymbol{x}}_d(\underline{\boldsymbol{u}})$ can be found iteratively or stored per camera pixel in a map
- the camera ray can be computed to

$$\underline{\boldsymbol{X}}(\tilde{\boldsymbol{x}}_d, \lambda) = \boldsymbol{R}_C^T\left(\lambda \cdot \tilde{\boldsymbol{x}}_d - \vec{\boldsymbol{t}}_C\right)$$

Projector

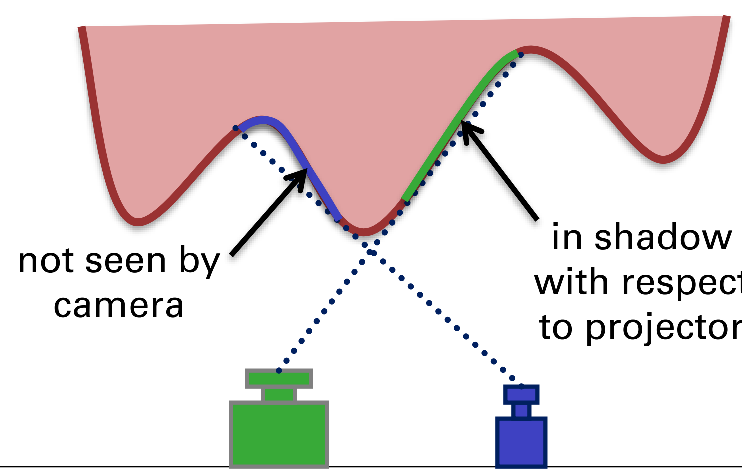
- The planes of a projector column become bent
- Three possible solutions
  - inversely distort projector pattern (can yield stair case artifacts for stripe patterns)
  - iteratively solve the ray surface intersection (multiple intersections possible!!)
  - project full 2D coordinate and intersect rays (slower acquisition)

ACQUISITION SETUPS

Standard Setup

- Uses one camera and one projector
- Calibrate projector camera system, optionally rectify
- Project structured light patterns from projector and acquire images with camera
- Projector and camera need to be synchronized
- Reconstruct points through triangulation
- Only points seen from camera AND projector can be reconstructed
- indirect lighting causes confusion and highlights lead to high range of brightness values

Standard Setup
(taken from David Laserscanner)



More Cameras

adding more cameras …

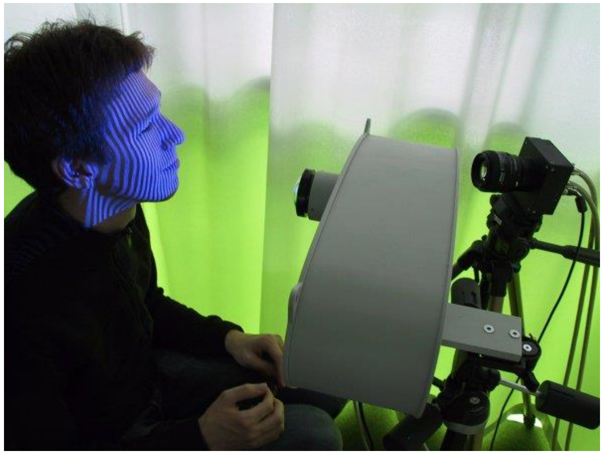
- reduces problems with highlights
- increases surface visibility with respect to cameras
- through multiple measurements of the same surface point the precision can be increased
- professional systems are mostly optimized for shape acquisition and do not reproduce colors at all (like "ATOS II Triple Scan") or in low quality

Setup with two cameras

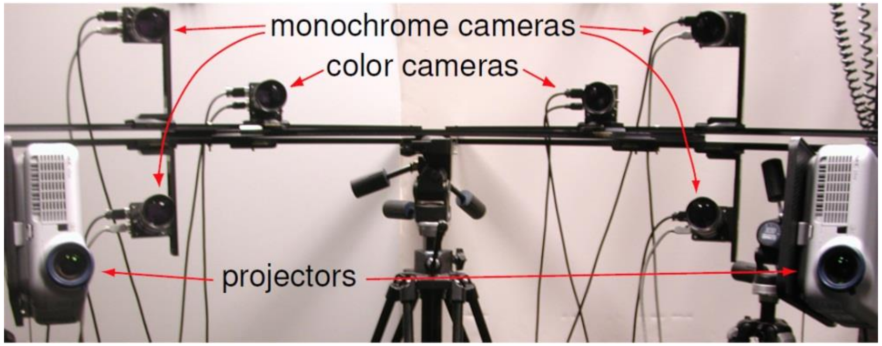
ATOS II Triple Scan (available at KTC chair in school of engineering)

Highspeed Setups

- in standard approaches several patterns need to be projected per 3D scan
- In dynamic setting one can use synchronized high speed projector and camera, but faces short illumination time
- other approaches use partially unsynchronized systems where the projector generates
  - random patterns for correspondence matching of synchronized stereo approach
  - static „single shot“ pattern that allows reconstruction from one pattern



use high speed components



use projector only to help stereo reconstruction

Kinect uses single shot approach

Challenges

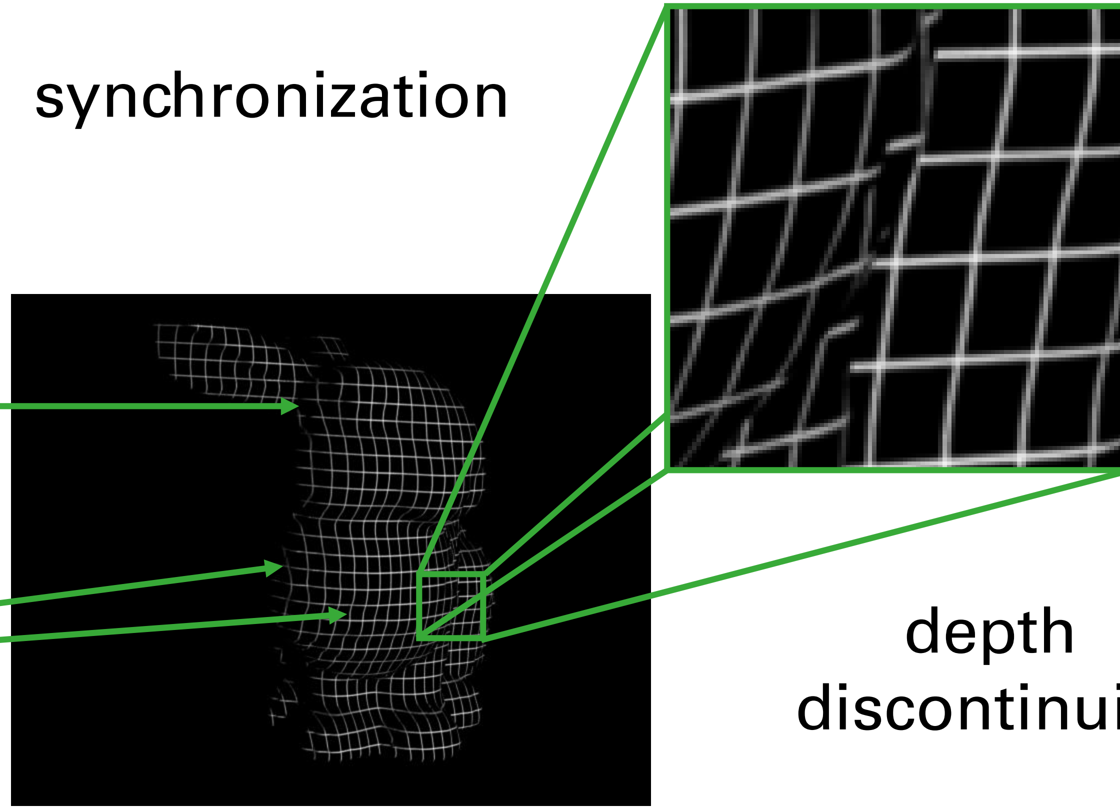
Computer Graphics and Visualization

calibration

synchronization

pattern deformation

brightness variations



depth discontinuity

modulation of pattern with object texture

brightness vs depth of field

movement of objects

STRUCTURED LIGHT APPROACHES

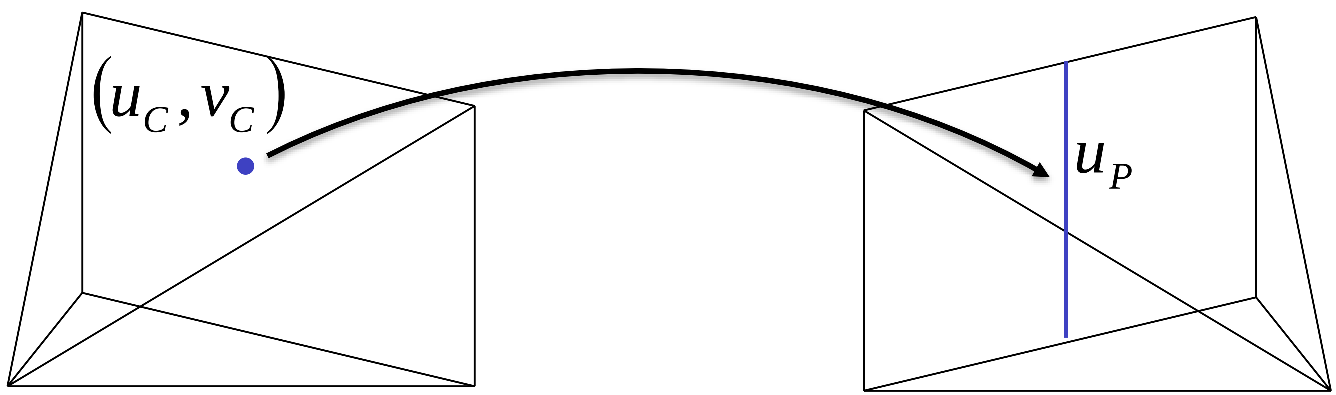
Basic Idea

- project n patterns $\Pi_i(u_P, v_P)$ that can be independent of v_P
- acquire scene with camera $\Gamma_i(u_C, v_C)$ such that correspondences $(u_C, v_C) \leftrightarrow u_P$ can be reconstructed
- assume simple model of projector-scene interaction (ignores interreflections, what will be refined later)

$$\Gamma_i(u_C, v_C) = L_d(u_C, v_C) \cdot \Pi_i(u_P, v_P) + L_{\text{amb}}(u_C, v_C)$$

pixel luminance from direct reflection

luminance due to background illumination



Line Shift Approach

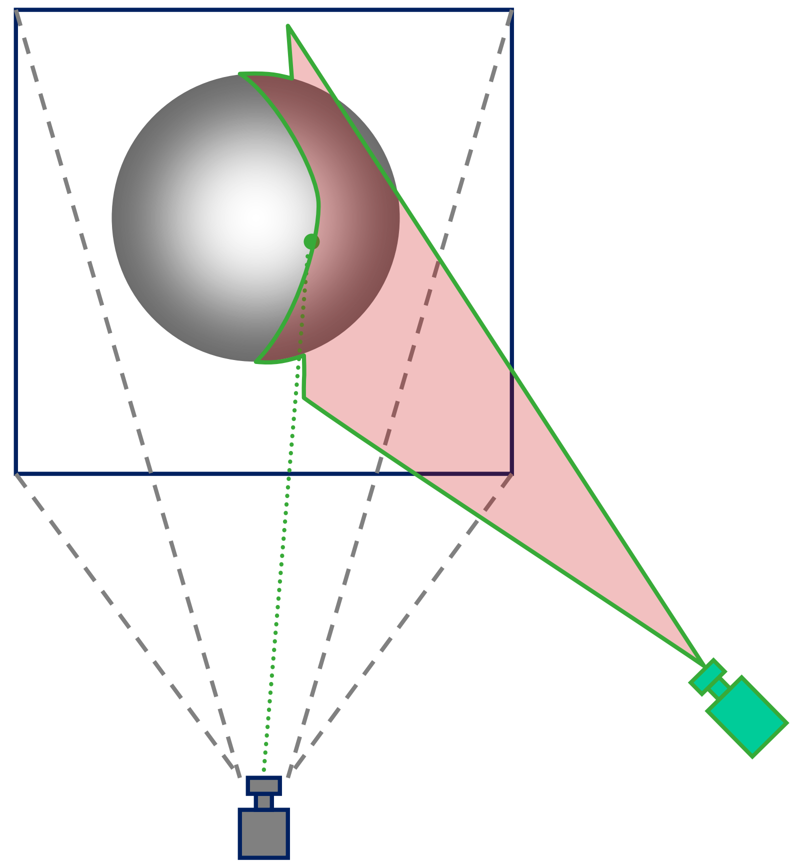
- project a one pixel wide stripe for each projector column:

$$\Pi_i(u_P, v_P) = \delta_{i,u_P} \qquad \delta_{i,u_P} = \begin{cases} 1 \text{ if } i = u_p \\ 0 \end{cases}$$

- reconstruction:

$$u_P(u_C, v_C) = \underset{i}{\max\arg}\, \Gamma_i(u_C, v_C)$$

- fit gaussian to do subpixel accurate detection, but be careful at
  - depth discontinuities
  - texture color discontinuities
- n patterns necessary, where n is number of projector columns



Direct Coding in Intensity

- encode projector column in intensity

$$\Pi(u_P, v_P) = u_P / (n-1)$$

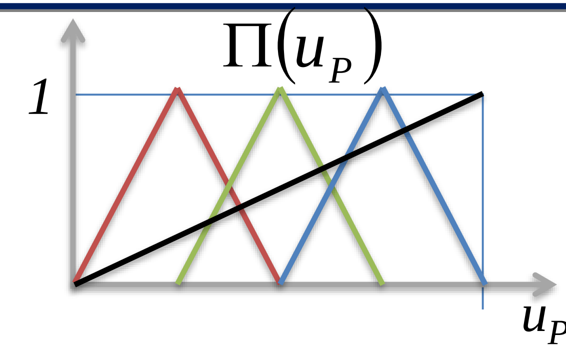
- project off and on patterns

$$\Pi_{\text{off}}(u_P, v_P) = 0$$
$$\Pi_{\text{on}}(u_P, v_P) = 1$$

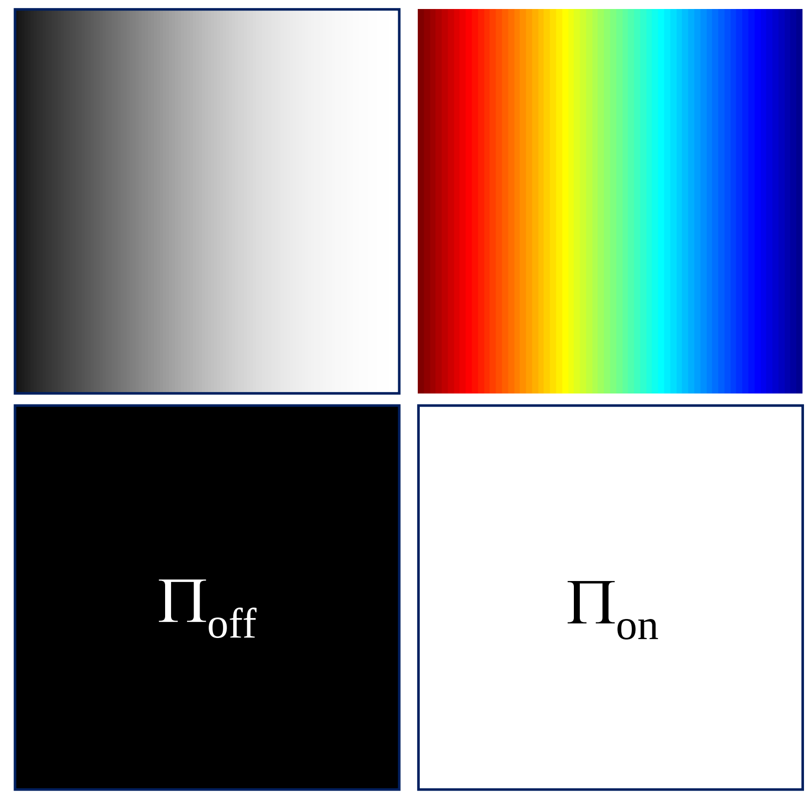
- reconstruct:

$$u_P = \frac{\Gamma(u_C, v_C) - \Gamma_{\text{off}}(u_C, v_C)}{\Gamma_{\text{on}}(u_C, v_C) - \Gamma_{\text{off}}(u_C, v_C)}$$

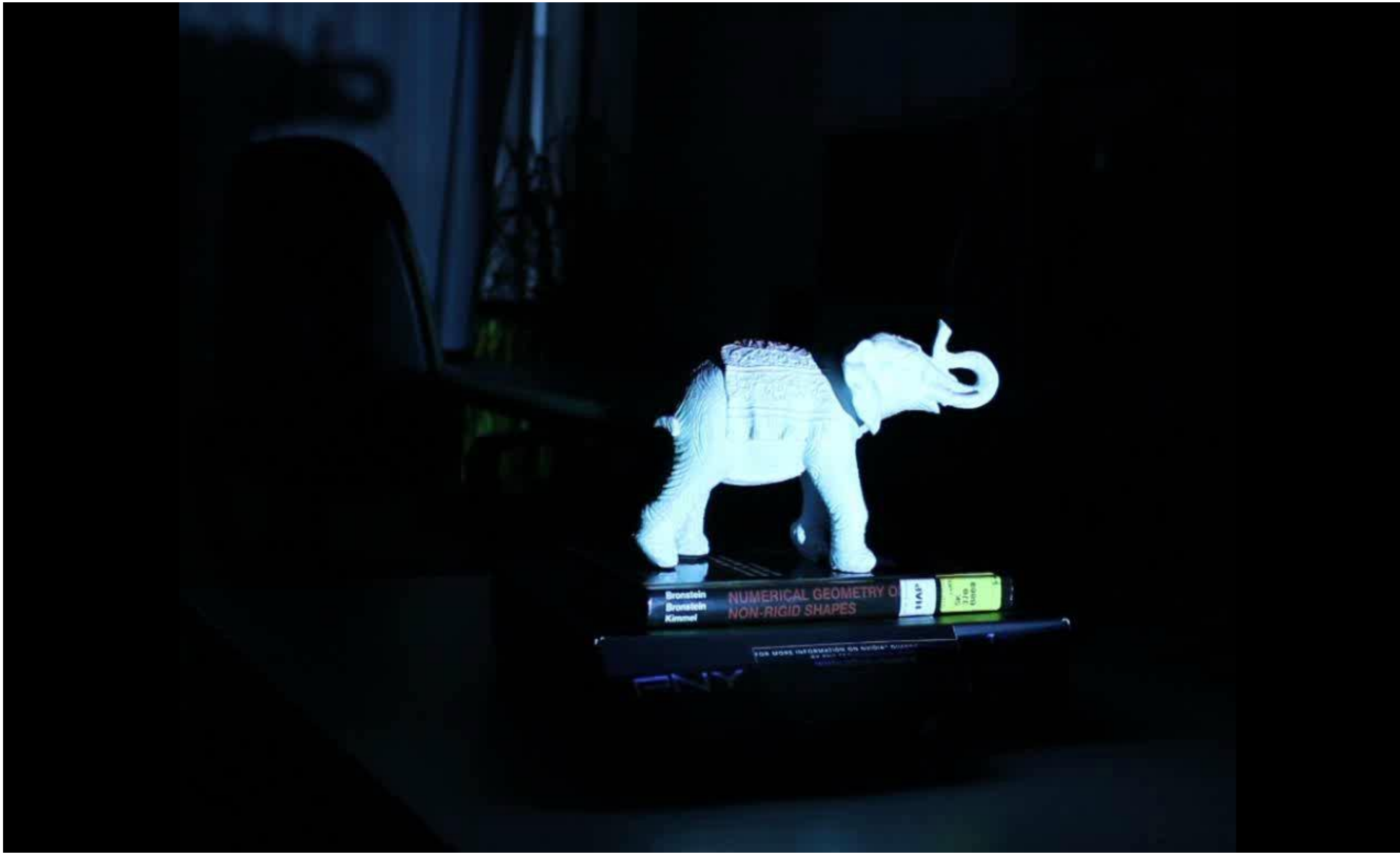
- Challenges
  - projector color resolution (8bit)
  - non linearity through gamma corrections



For acquisition of white surfaces, all color channels can be exploited



Example Gray Code Pattern Sequence



on and off patterns + 10 bits column code + 10 bits row code

Binary and Gray Code

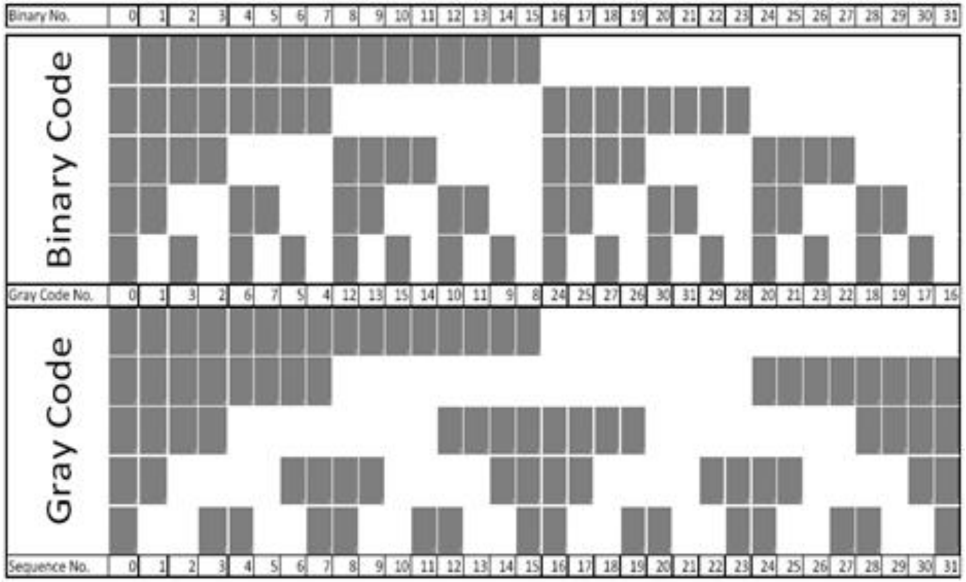
- encoding of gray code

XOR

```
gray(binary) {
  n1 = binary;
  return n1 ^ (n1/2);
}
```

(converts to reflected graycode)

- decoding of gray code:




```
binary(gray) {
  n1 = n2 = gray;
  while (n1 >>= 1) n2 ^= n1;
  return n2;
}
```

(converts to reflected binary code)

Binary and Gray Code

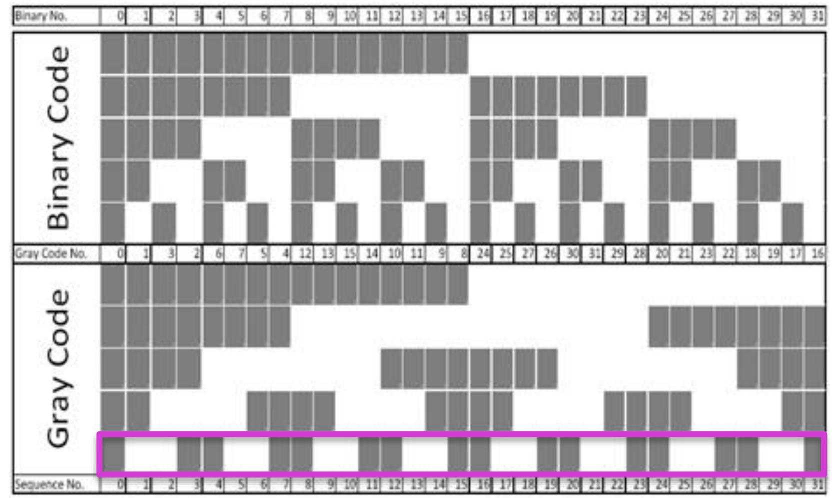
Computer Graphics and Visualization

- encode projector column with binary code

$$\Pi_i(\underline{\boldsymbol{u}}_p) = \mathrm{bit}(i, \mathrm{encode}(u_P))$$

- decode single bit

$$\mathrm{b}_i(\underline{\boldsymbol{u}}_C) = \mathrm{classify}(\Gamma_i(\underline{\boldsymbol{u}}_C))$$



lower highest frequ. pattern

- simplest classification with on and off pattern

$$\mathrm{b}_i(\underline{\boldsymbol{u}}_C) = \mathrm{classify}(\Gamma_i(\underline{\boldsymbol{u}}_C)) = \begin{cases} 1 & \Gamma_i(\underline{\boldsymbol{u}}_C) > \tau + \varepsilon \\ 0 & \Gamma_i(\underline{\boldsymbol{u}}_C) < \tau - \varepsilon \\ \boxed{\text{undef} \quad \text{otherwise}} \end{cases}$$

$$\tau = \tfrac{1}{2}\left(\Gamma_{\mathrm{off}}(\underline{\boldsymbol{u}}_C) + \Gamma_{\mathrm{on}}(\underline{\boldsymbol{u}}_C)\right)$$

- decode projector column

$$u_P(u_C, v_C) = \mathrm{decode}\{\mathrm{b}_i(u_C, v_C)\}$$

- if one bit is undef, no u_P can be decoded
- $\log n + 2$ measurements

Phase Shift Approach [Srinivasan 1985]

- project three shifted cosine patterns:

$$\Pi_i(\underline{\boldsymbol{u}}_p) = \tfrac{1}{2} + \tfrac{1}{2}\cos(\varphi(u_P) + d_i)$$

- column encoded in phase

$$\varphi(u_P) = 2\pi f u_P$$

- phase shift for N patterns:

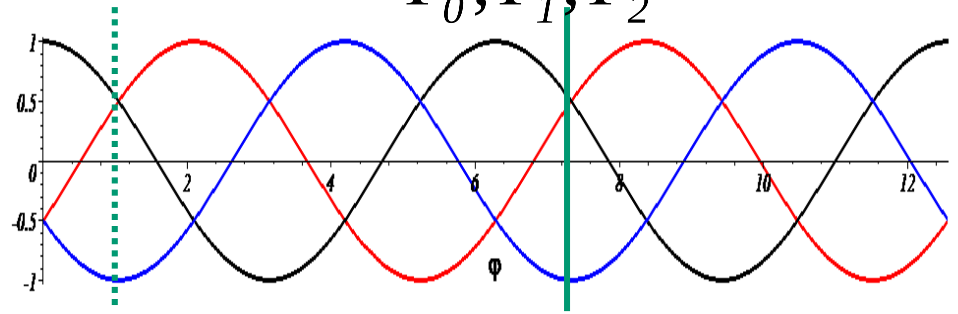
$$d_i = i\,\tfrac{2\pi}{N}, \text{i.e. } d_i = \left\{0, \tfrac{2\pi}{3}, \tfrac{4\pi}{3}\right\}$$

- measured images

$$\Gamma_i = L_d \cdot \Pi_i(\underline{\boldsymbol{u}}_P) + L_{\mathrm{amb}}$$

$$\Rightarrow \Gamma_i = A \cdot \cos(\varphi(u_P) + d_i) + B$$

$$\Gamma_0, \Gamma_1, \Gamma_2$$



- three patterns suffice:

$$\Gamma_0 = A \cdot \cos(\varphi(u_P)) + B$$
$$\Gamma_1 = A \cdot \cos\left(\varphi(u_P) + \tfrac{2\pi}{3}\right) + B$$
$$\Gamma_2 = A \cdot \cos\left(\varphi(u_P) + \tfrac{4\pi}{3}\right) + B$$

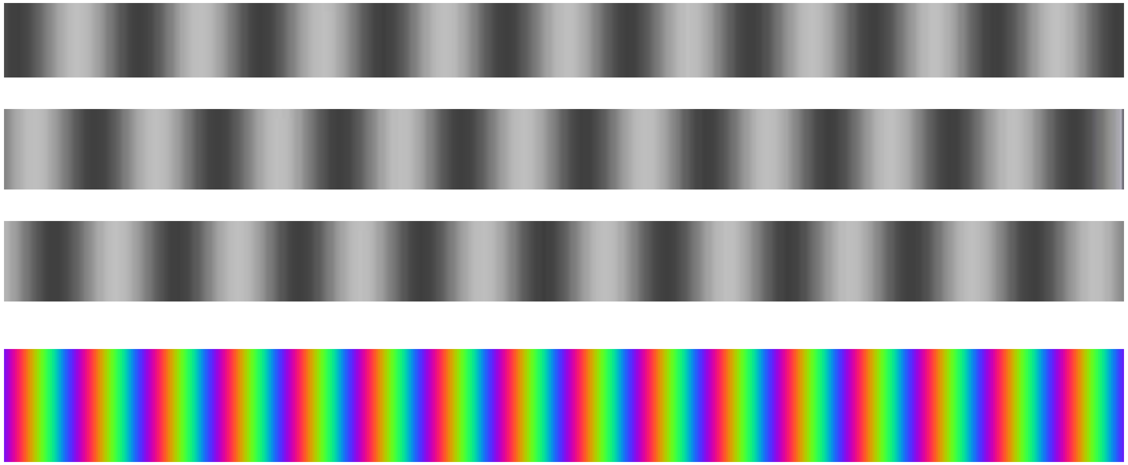
- eliminate A/B and solve for phase:

$$\tan\varphi(u_P) = \sqrt{3}\,\frac{\Gamma_2 - \Gamma_1}{2\Gamma_0 - (\Gamma_1 + \Gamma_2)}$$

- up to period ➔ unwrapping

Phase Shift Discussion

- use three or more shifted cosine patterns to encode the projector column in phase (e.g. in the 3 color channels)



- Advantage (without color coding): almost independent of the object texture and the sharpness of the projection
- Problem of ambiguous phase reconstruction can be solved by combination with Gray code or by hierarchical phase shift
- If the objects are colored, the color channels cannot be used.

Debruijn Sequences I

Computer Graphics and Visualization

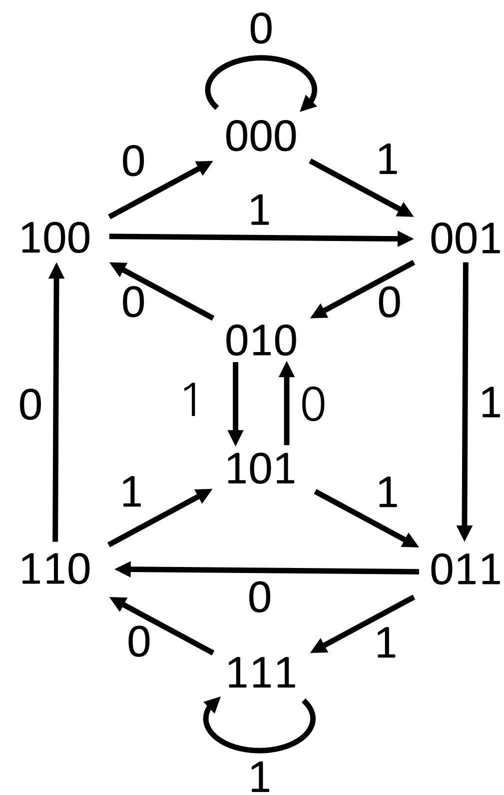
Debruijn Sequence B(n,m):
Size of Alphabet: n
all sub sequences of length m are unique

Example B(2,4):

0 0 0 0 1 1 1 1 0 1 1 0 0 1 0 1

≠ ≠ ≠ ≠ ≠ ≠ ≠ ≠ ≠ ≠ ≠ ≠

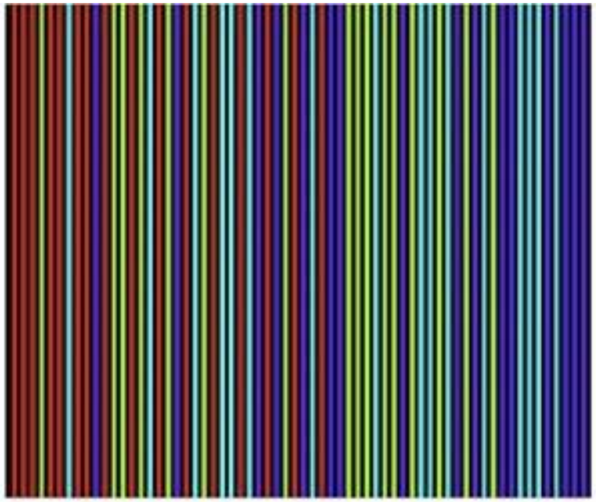
0 **0 0 0 0 0 0 0 0 0 0** 0 **0** 0 0



Debruijn Graph:

- Nodes are all possible sequences of length Länge m-1
- For each node one outgoing edge per symbol in the alphabet
- Eulerian Path of Debruijn Graph yields Debruijn Sequence

Debruijn Sequences II

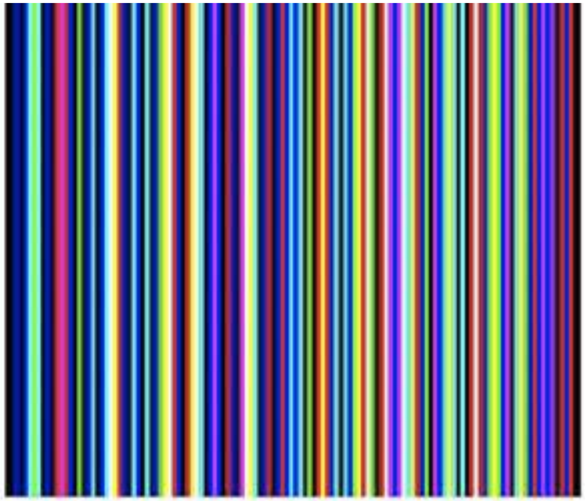


Line coding

Alphabet = Color

Color of line i is i^{th} symbol
of Debruijn Sequence

- One-Shot-Approach
- To be examined neighborhood: 2m lines
- At depth jumps erroneous decoding



Color change coding [Zhang 2002]

Alphabet = Numbers 1-7 (Binary)

„XOR“

$$p_{j+1} = p_j \text{ XOR } d_j$$

$$\begin{pmatrix}1\\0\\0\end{pmatrix} = \begin{pmatrix}0\\0\\1\end{pmatrix} \text{XOR} \begin{pmatrix}1\\0\\1\end{pmatrix}$$

- To be examined neighborhood: m lines

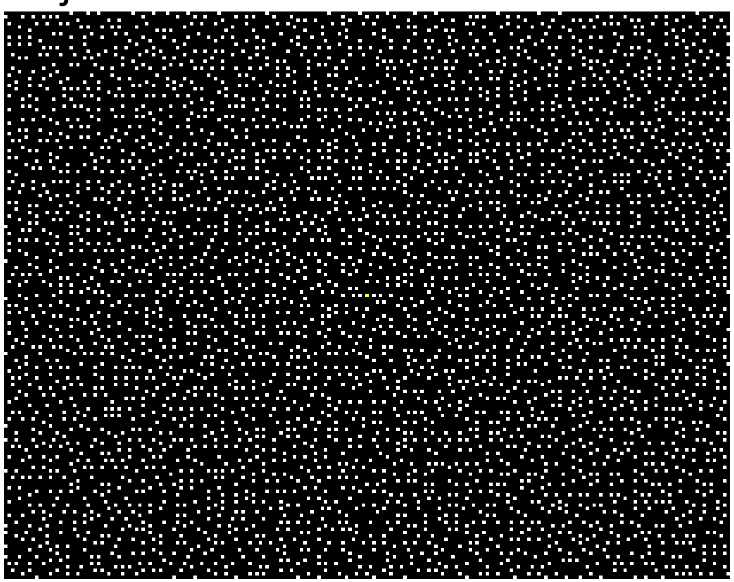
Microsoft Kinect

color camera

infrared laser projector

infrared camera

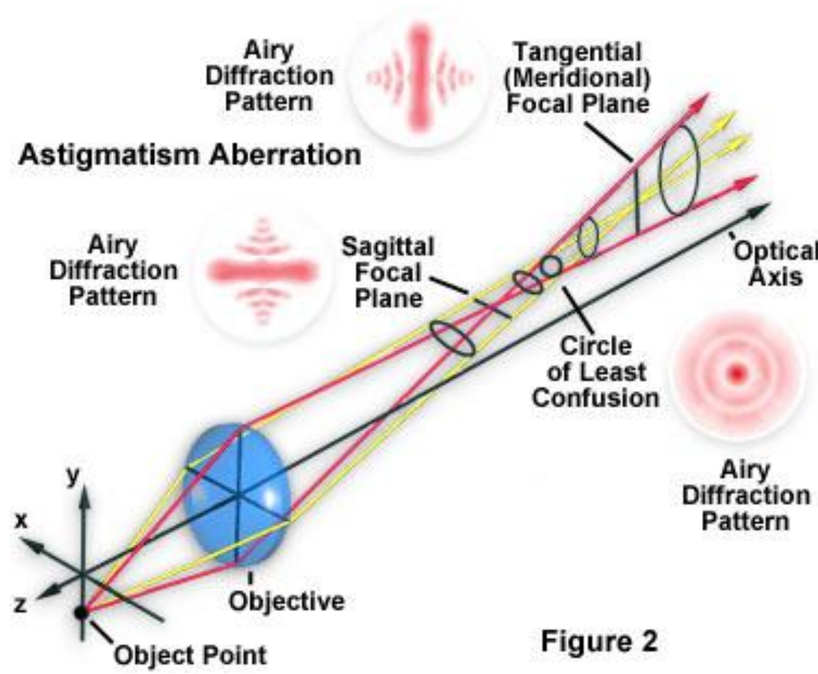
Projection Patter:



Freedman et al, PrimeSense patent application US 2010/0290698

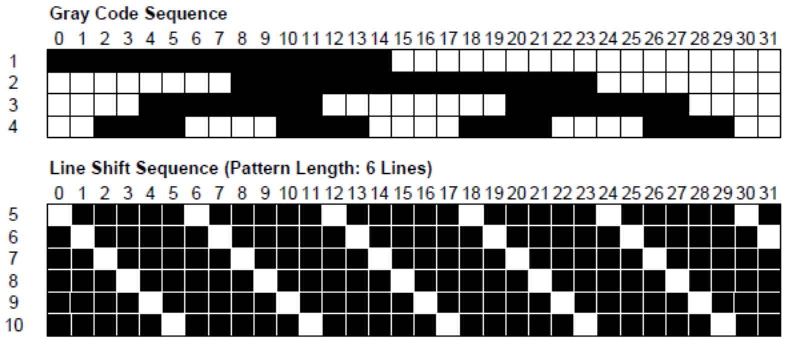
Depth Measurement Approach (speculations)

1. Stereoblockmatching with point pattern
2. Depth from anisotropic blur based on astigmatic lenses with two different focal lengths

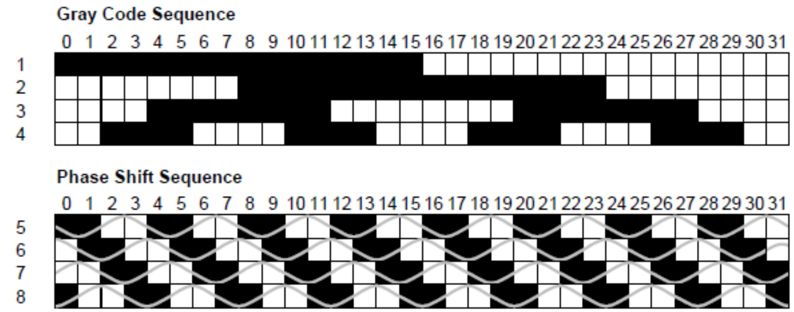


Freedman et al, PrimeSense patent application US 2010/0290698

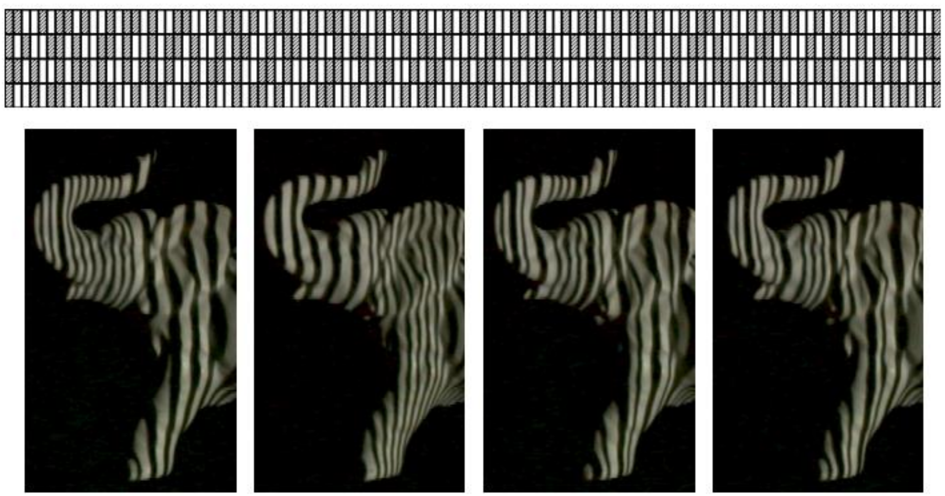
More Approaches



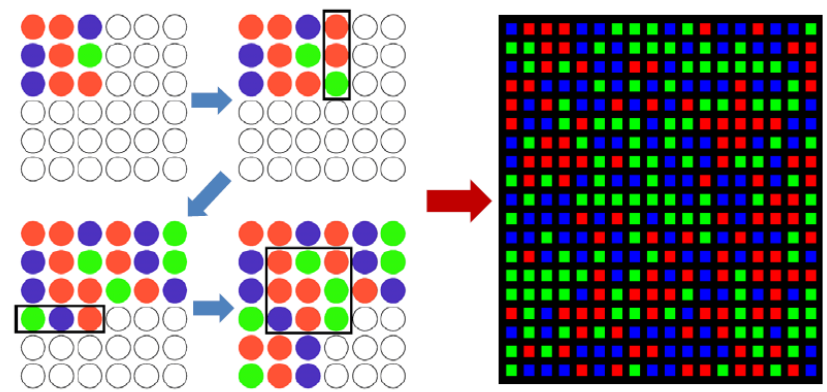
Hybrid: Graycode + Line Shift



Hybrid: Graycode + Phase Shift



Stripe Boundary Codes [Hall-Holt 2001]



Extension of Debruijn to 2D

content and images taken from

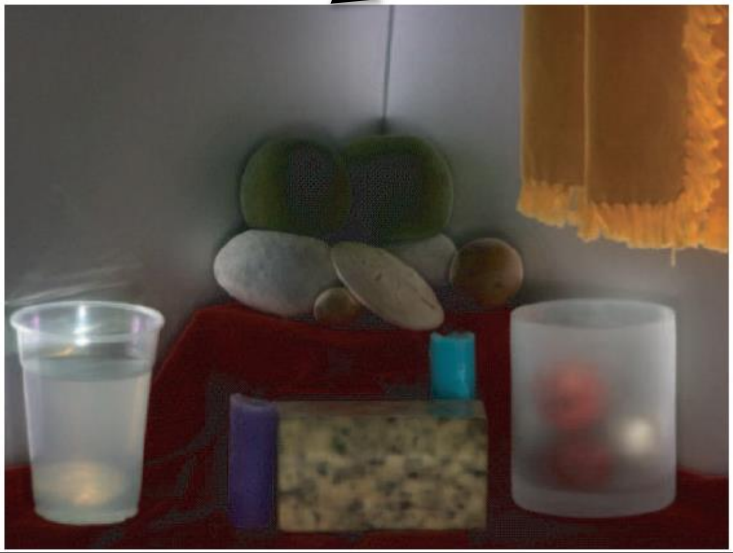
S.K. Nayar, G. Krishnan, M. D. Grossberg, R. Raskar, *Fast Separation of Direct and Global Components of a Scene using High Frequency Illumination,* ACM Trans. on Graphics (also Proc. of ACM SIGGRAPH), 2006.

SEPARATION OF DIRECT AND INDIRECT ILLUMINATION

Motivation

scene with global illumination effects illuminated from a point light source

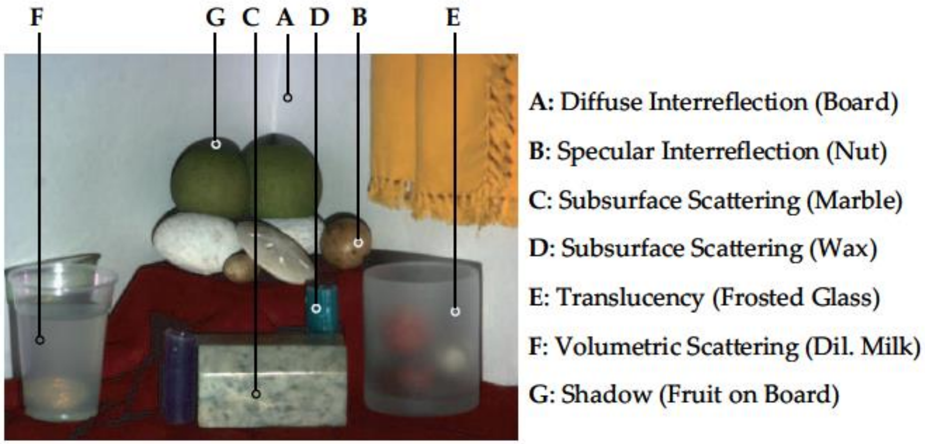
directly reflected light

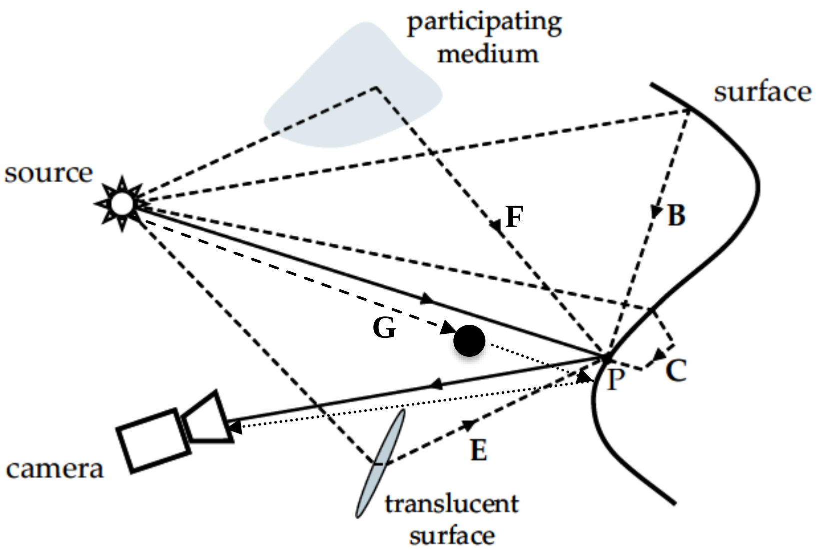


indirectly reflected light

Indirect Illumination

- in structured light scanning only the direct illumination is of interest
- scene points in shadow (G) should be ignored
- at other points the luminance due to
  - diffuse or specular interreflections (B)
  - subsurface scattering (C)
  - transluceny (E), or
  - volumetric scattering (F)
- should be determined and filtered out





Idea of Separation

- per pixel we want to split incoming luminance:

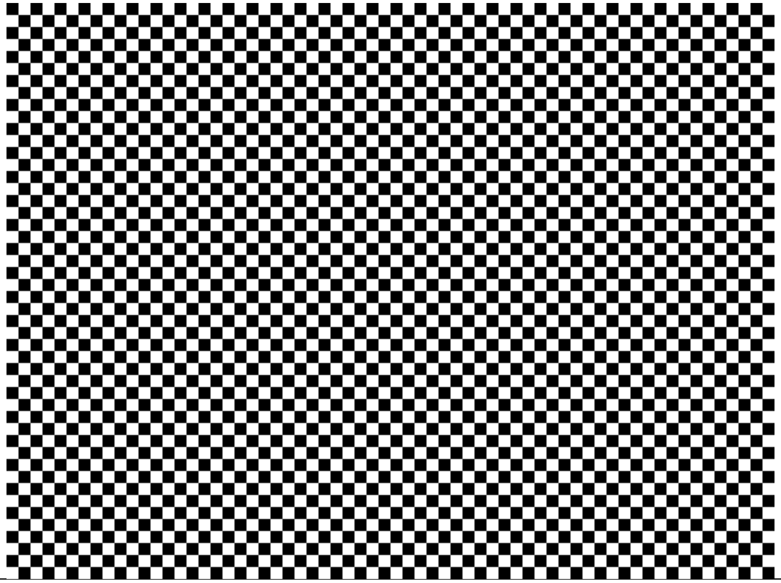
  $$L = L_d + L_g$$

  in direct component L_d and indirect or global comp. L_g
- **assumtion:** L_g is a smooth function of projected direct light pattern (violated for mirror reflection)
- **idea:** project high frequency pattern with 50% pixels on and its negative, such that each scene point is once illuminated and once not illuminated
- measure minimum and maximum luminance

  $$L_{\min} = \tfrac{1}{2} L_g$$

  $$L_{\max} = L_d + \tfrac{1}{2} L_g$$
- and reconstruct comp.:

  $$L_d = L_{\max} - L_{\min}, \quad L_g = 2L_{\min}$$



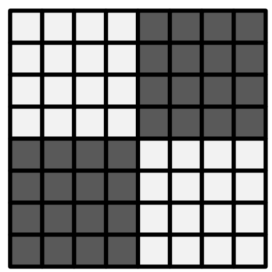
Practical Separation

problem 1:

- camera pixels can overlap parts of both black and white projector pixels
- projector sharpness is not perfect and varies over acquisition volume

solution 1:

- do not use maximum frequency (i.e. 4x4 up to 6x6 squares)



- project several shifted versions of pattern (i.e. 16 for 4x4 with offsets +0, +2, +4, +6 in x- and y-direction) and compute $L_{\min}/L_{\max}$ per pixel over all acquired images

problem 2:

- black projector pixels still emits some fraction b of brightness

solution 2:

- calibrate projector for b and extend formulae:

$$L_{\min} = bL_d + (1+b)\tfrac{1}{2} L_g$$

$$L_{\max} = L_d + (1+b)\tfrac{1}{2} L_g$$

$$L_d = \frac{L_{\max} - L_{\min}}{1-b}$$

$$L_g = 2\frac{L_{\min} - bL_{\max}}{1-b^2}$$

Extension to Phase Shift

- each cosine pattern distributes light uniformly, such that all generate the same global component $\frac{1}{2}L_g$:

$$\Gamma_i = L_d \cdot \left(\tfrac{1}{2} + \tfrac{1}{2}\cos\left(\varphi\left(u_P\right) + d_i\right)\right) + \tfrac{1}{2} L_g + L_{\text{amb}}$$

- luminance from ambient illumination and global component add up per pixel, such that standard phase shift is based on direct component only
- in order to reconstruct the indirect component one can project an off pattern to determine L_{amb} directly

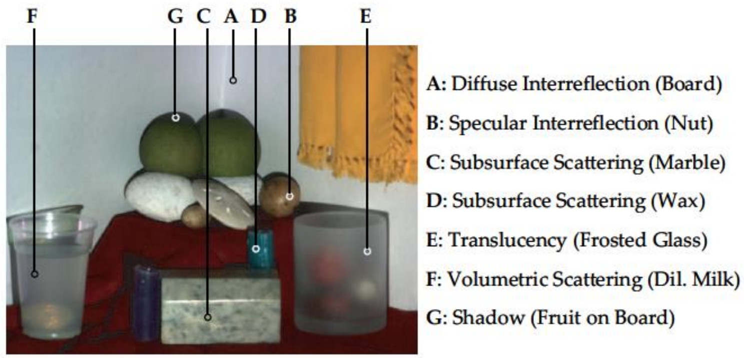
Verification

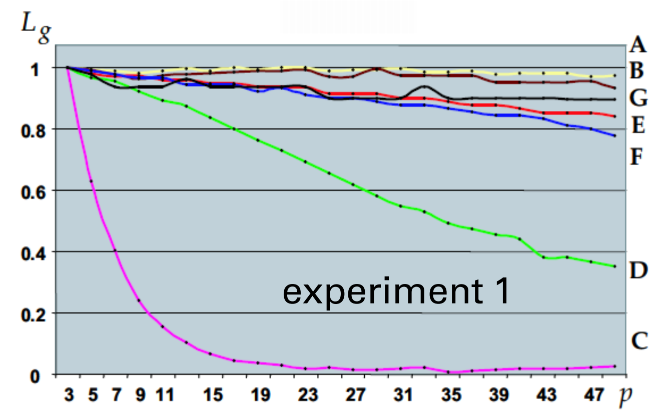
experiment 1

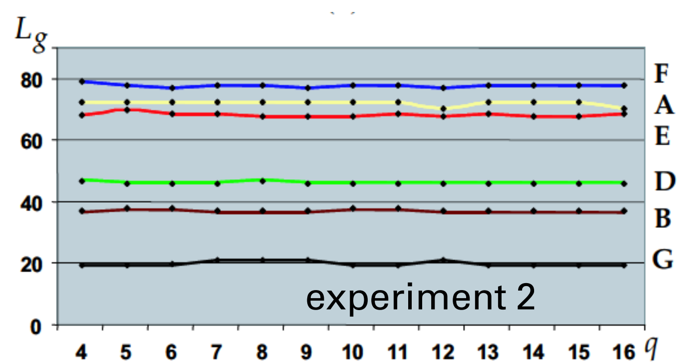
- illuminate all but square of increasing size p around point of interest
- this yields the global component only
- in case of subsurface scattering square needs to be small

experiment 2

- use checkerboard with squares of increasing pixel size q for separation (point C excluded)
- shows invariance to q

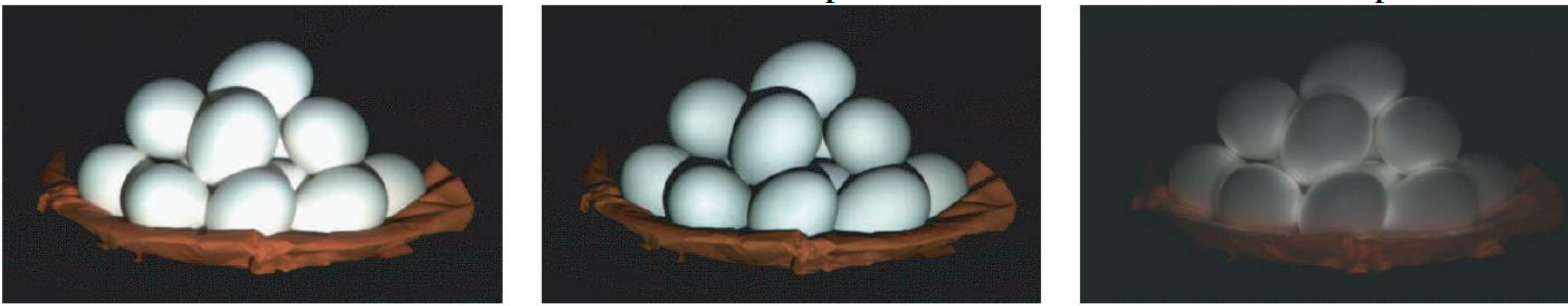






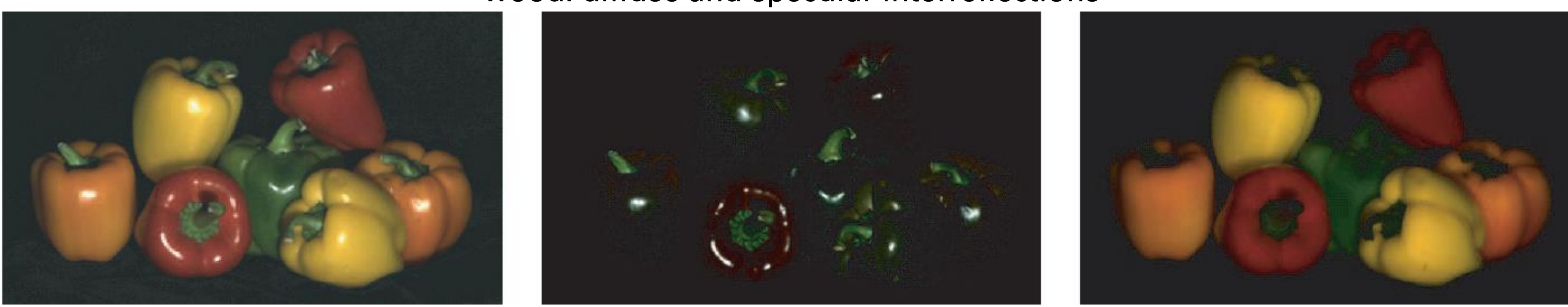
Results

Scene	Direct Component	Global Component



eggs: diffuse interreflections

wood: diffuse and specular interreflections

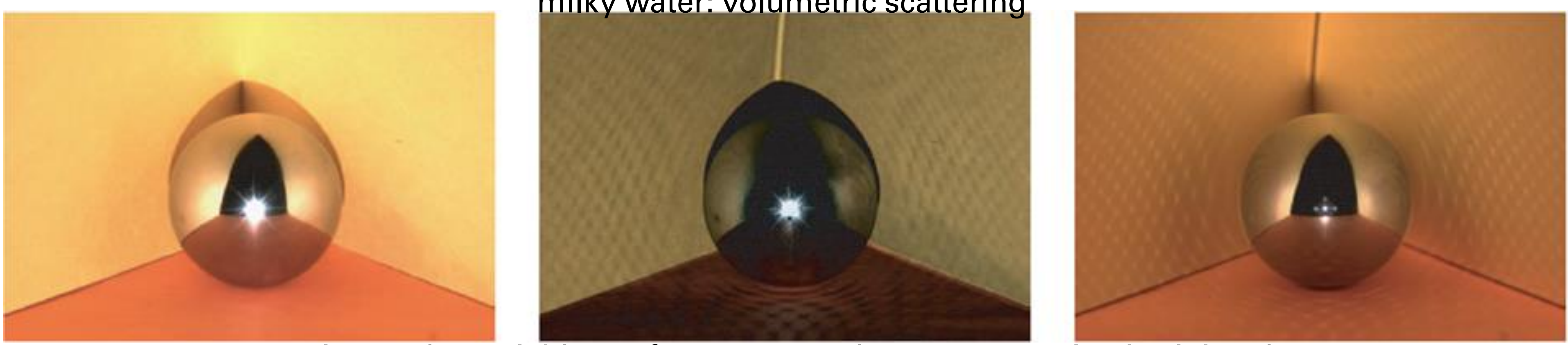


peppers: subsurface scattering

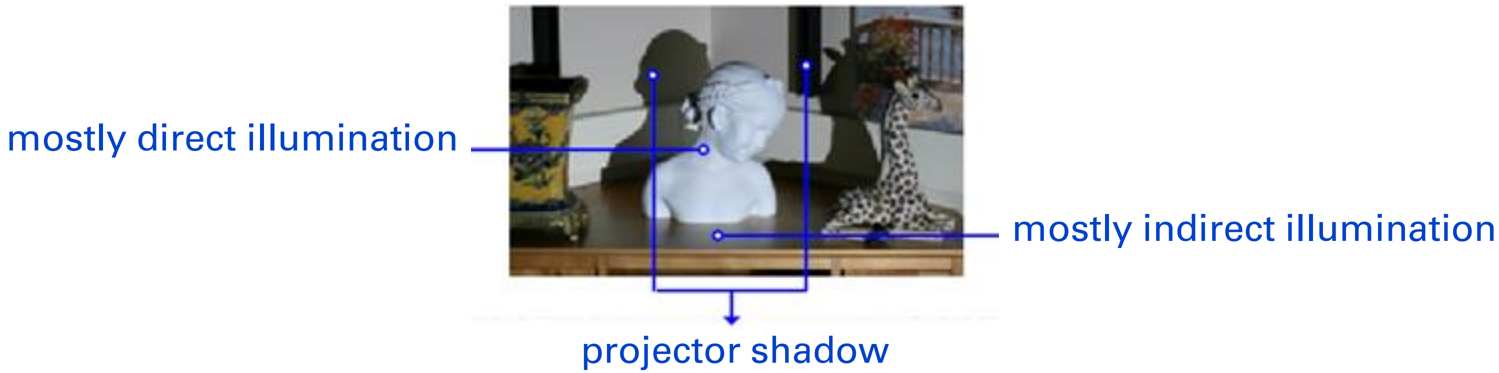
Results

grapes and cheese: subsurface scattering

milky water: volumetric scattering



mirror sphere yields artefacts as smoothness assumption is violated



content and images taken from

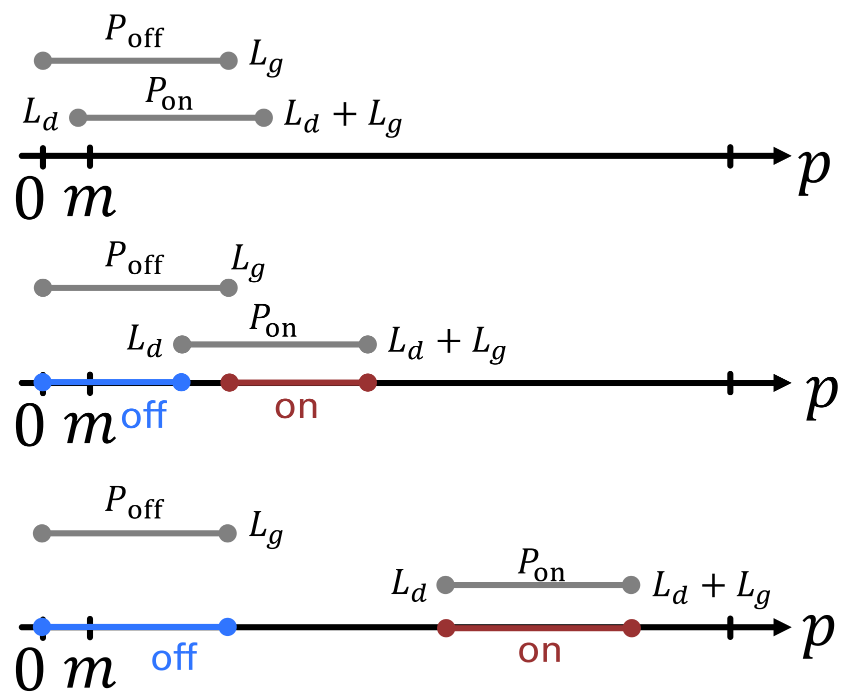
Y. Xu, D. Aliaga: *Robust pixel classification for 3D modeling with structured light*. Graphics Interface 2007: 233-240

ROBUST PIXEL CLASSIFICATION

Robust Pixel Classification

- first direct-indirect light separation is done as in previous paper yielding per pixel direct and global component: $L = L_d + L_g$
- if L_d is less than threshold m, scene point is in projector shadow
- otherwise pixel values p in images Γ_i are classified to decode projector column
- L_g is an upper bound on indirect light component for illumination with any pattern Π_i

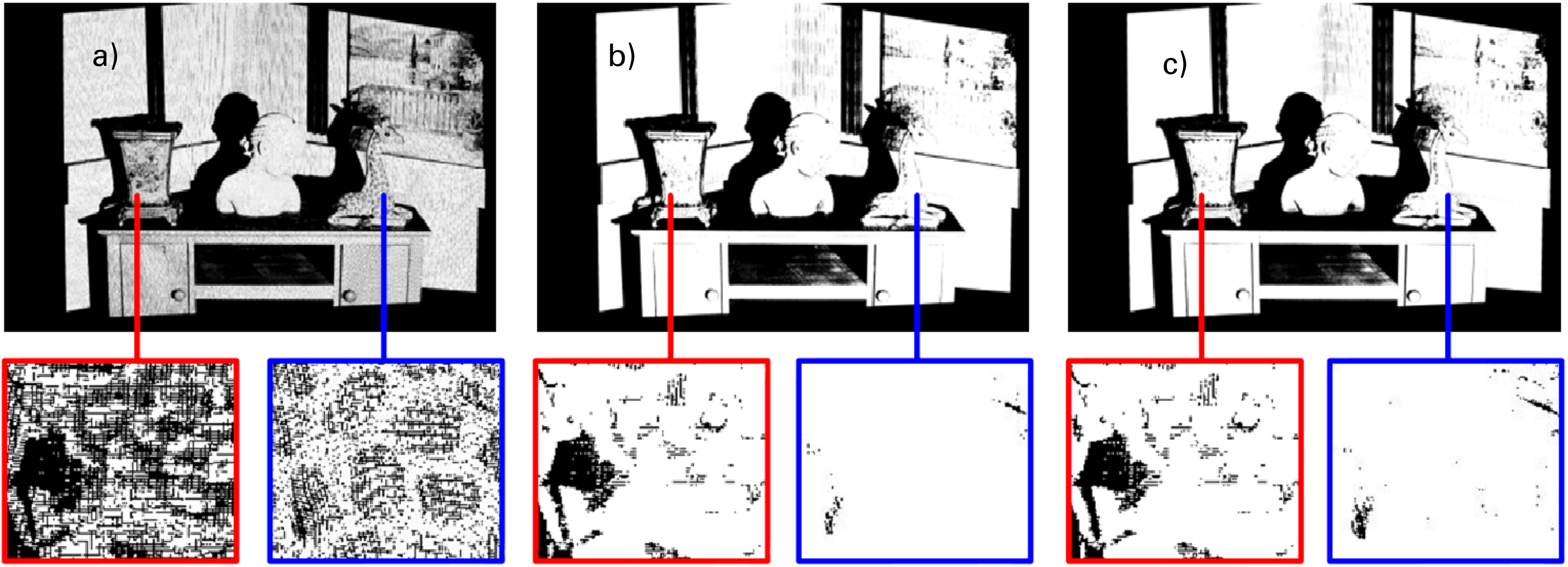
- Conservative estimate of luminance intervals for on and off pixel classification:
  $P_{\text{off}} = [0, L_g], \qquad P_{\text{on}} = [L_d, L_d + L_g]$



$L_d < m \rightarrow$ pixel uncertain
$p < \min(L_d, L_g) \rightarrow$ pixel off
$p > \max(L_d, L_g) \rightarrow$ pixel on
otherwise $\rightarrow$ pixel uncertain

Dual Pattern Rules and Comparison

- If two complementary patterns Γ_i and $\bar{\Gamma}_i$ are available, one can add the constraint that a pixel must classify oppositely in the two patterns
- comparison of approaches:



white pixels are classified correctly for a) standard method b) single and c) dual pattern rules

CONCLUSION

Conclusion

- camera-projector setup has problems with highlights which can be eliminated by adding a second camera
- gray codes, phase shift and their combinations are most prominent methods
- one needs to project in order of $\log n$ patterns
- to reduce the number of patterns for fast scanning, one needs to encode projector column in spatial neighborhood
- direct illumination component can be separated from indirect one with two complementary high frequency patterns
- robust binary classification uses global indirect light component to derive classification intervals
- we did not cover brightness and color calibration. Both projector and camera do not map them linearly!!!

References

- V. Srinivasan, H.-C. Liu, M. Halioua. *Automated phase-measuring profilometry: a phase mapping approach*. Applied Optics **24**(2), 1985, pp 185-188.
- O. Hall-Holt, S. Rusinkiewicz. *Stripe boundary codes for real-time structured-light range scanning of moving objects*. ICCV 2001
- L. Zhang, B. Curless, S. M. Seitz. *Rapid shape acquisition using color structured light and multi-pass dynamic programming.* 3D Data Processing Visualization and Transmission (3DPVT), 2002
- S.K. Nayar, G. Krishnan, M. D. Grossberg, R. Raskar, *Fast Separation of Direct and Global Components of a Scene using High Frequency Illumination,* ACM Trans. on Graphics (also Proc. of ACM SIGGRAPH), 2006.
- Y. Xu, D. Aliaga: *Robust pixel classification for 3D modeling with structured light*. Graphics Interface 2007: 233-240