CS 775
Advanced Computer Graphics

Tracking for Augmented Reality

Augmented Reality

- Make sense of the real world
  - Image the real world with a real calibrated camera
  - Detect features
  - Reason about world geometry with the features
  - Compute camera pose
- Augment the image of the real world
  - Construct synthetic camera
  - Contruct synthetic content
  - Render content with camera

Augmented Reality

- Make sense of the real world
  - Image the real world with a real calibrated camera
  - Detect features
  - Reason about world geometry with the features
  - Compute camera pose
- Augment the image of the real world
  - Construct synthetic camera
  - Contruct synthetic content
  - Render content with camera

Augmented Reality

- Make sense of the real world
  - Image the real world with a real calibrated camera
  - Detect features
  - Reason about world geometry with the features
  - Compute camera pose
- Augment the image of the real world
  - Construct synthetic camera
  - Contruct synthetic content
  - Render content with camera

CS475/675

Augmented Reality

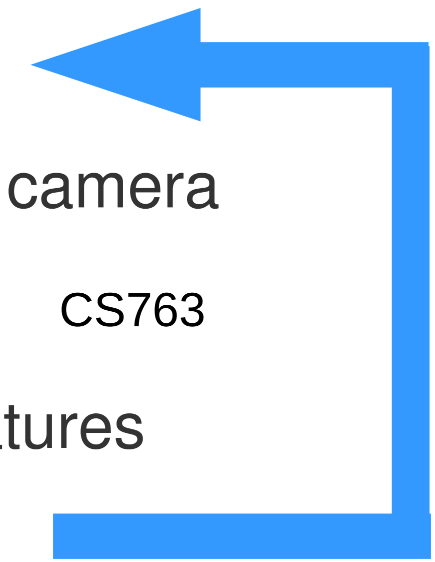
- Make sense of the real world
  - Image the real world with a real calibrated camera
  - Detect features
  - Reason about world geometry with the features
  - Compute camera pose
- Augment the image of the real world
  - Construct synthetic camera
  - Contruct synthetic content
  - Render content with camera

CS763

Augmented Reality

The Tracking Loop

- Make sense of the real world
  - Image the real world with a real calibrated camera
  - Detect features
  - Reason about world geometry with the features
  - Compute camera pose



- If we assume we know what the world is like apriori
- We know what to look for
- Easier to do, less compute intensive, restrictive
- Marker-based Tracking

- If we do not know what the world is like
- We must reconstruct (a part of) the world on the fly.
- Harder to do, more compute intensive, more versatile
- Markerless Tracking

Augmented Reality

The Tracking Loop

- Make sense of the real world
  - Image the real world with a real calibrated camera
  - Detect features
  - Reason about world geometry with the features
  - Compute camera pose

CS763

- Mostly interested in doing camera tracking in real-time for AR.
- Done offline for VFX (called camera matchmoving).
- Challenges
  - Noise, Blur, Illumination, Occlusion, Deformable Geometry

Marker-based Tracking

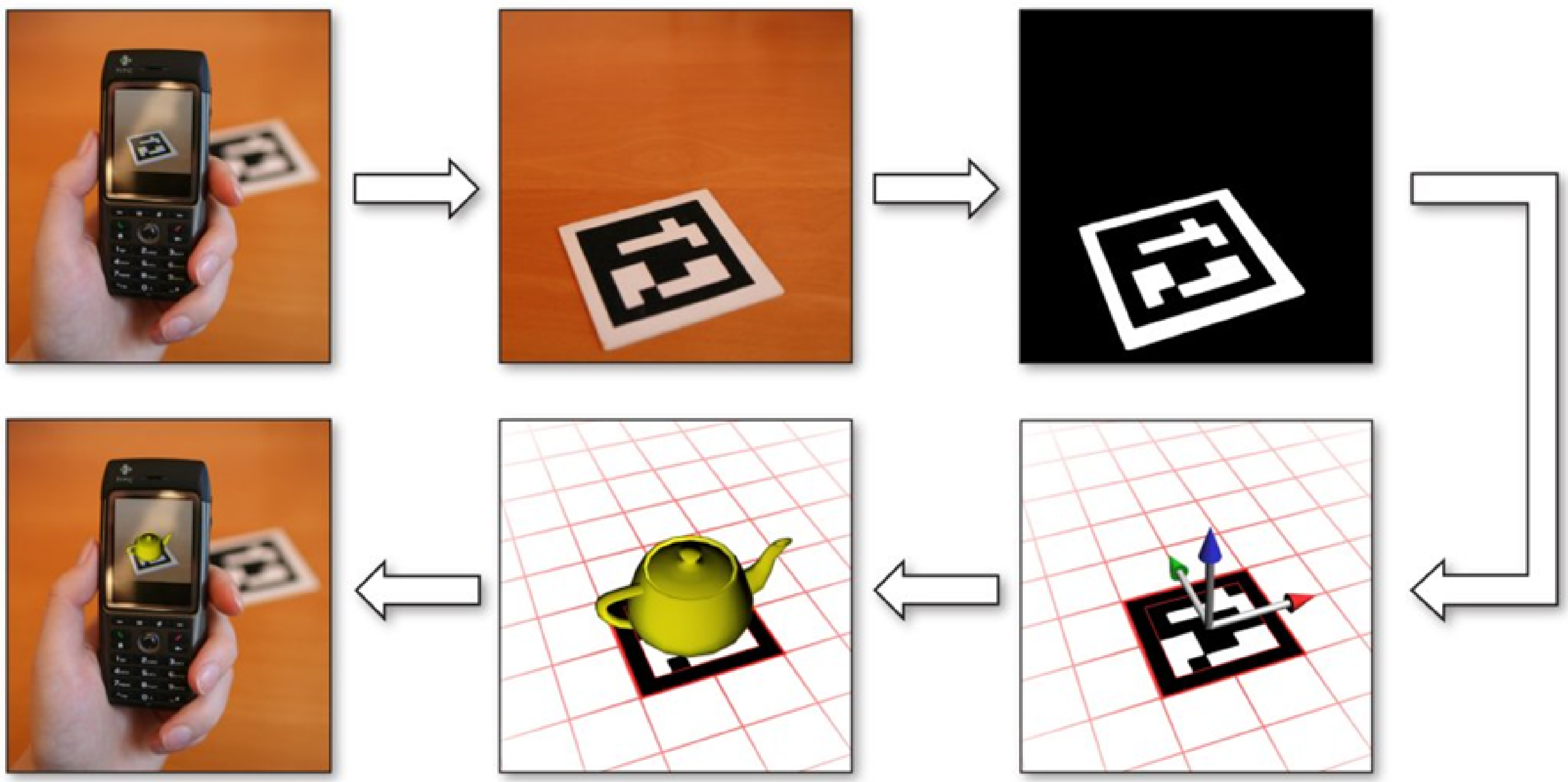


Image: Daniel Wagner
www.augmentedrealitybook.org

Marker-based Tracking

- The Computer Vision Camera

$$P = K[R|t]$$

3x4 Camera Matrix

$$K = \begin{bmatrix} f_x & s & c_u \\ 0 & f_y & c_v \\ 0 & 0 & 1 \end{bmatrix}$$

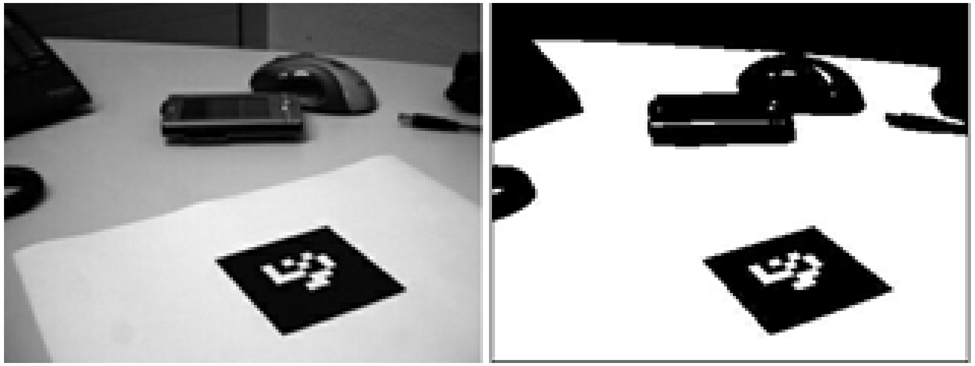
Intinsic Parameters

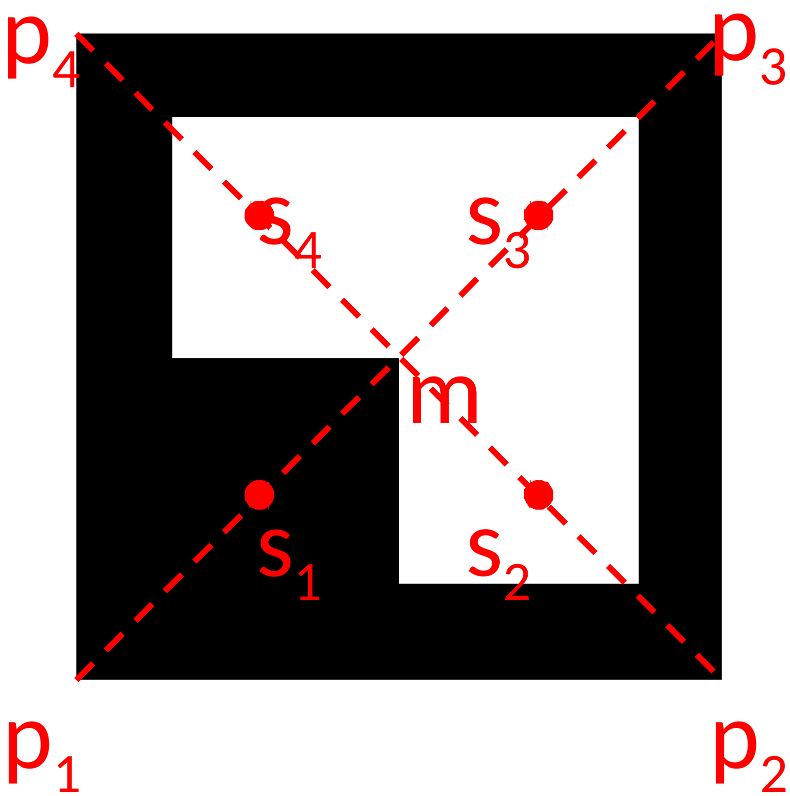
$$[R|t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$$

Extrinsic Parameters

Marker-based tracking

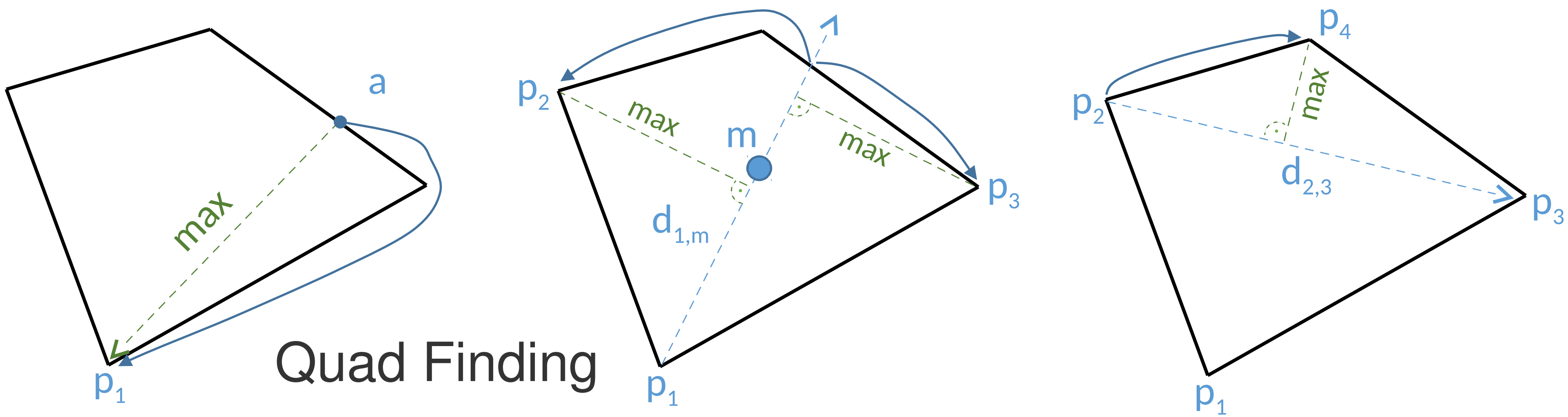
- Marker Detection
  - Grayscale image
  - Image thresholding





www.augmentedrealitybook.org

Marker-based Tracking

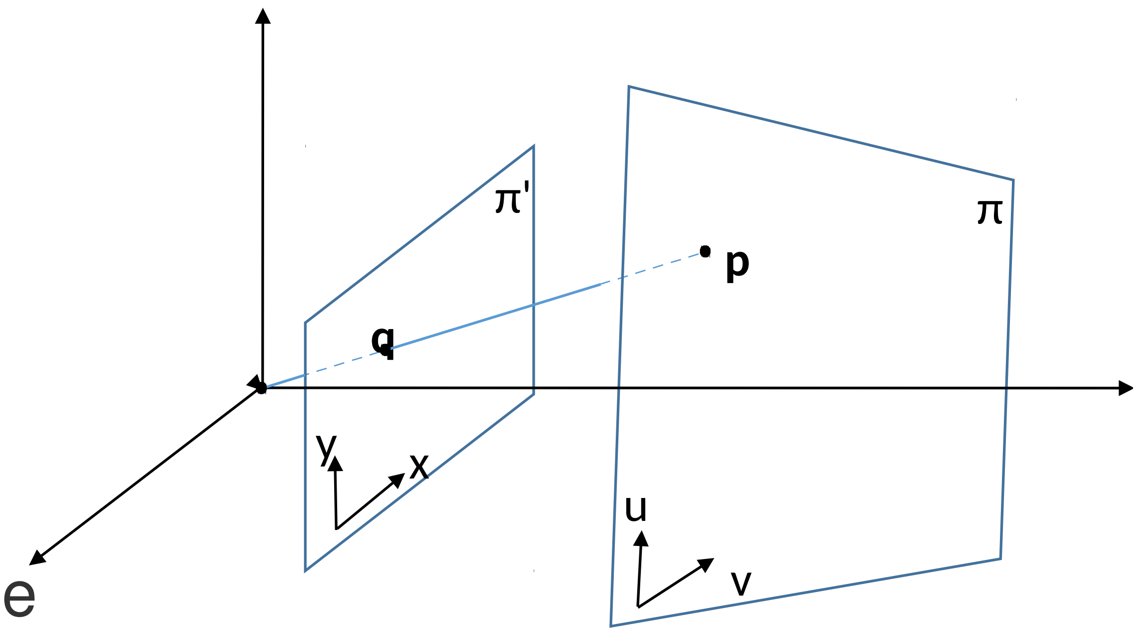


Quad Finding

- Find edges (black pixel after white) on every *n*-th line
- Follow edge in 4-connected neighborhood
- Until loop closed or hitting border
- Start at **a** and walk contour, search $\mathbf{p}_1$ at maximum distance
- Compute centroid **m**
- Find corners $\mathbf{p}_2$, $\mathbf{p}_3$ on either side of $\mathbf{d}_{1,m}=(\mathbf{p}_2,\mathbf{p}_3)$
- Find farthest point $\mathbf{p}_4$
- Determine orientation from black corner at $\mathbf{s}_i=(\mathbf{p}_i+\mathbf{m})/2$

www.augmentedrealitybook.org

Pose Estimation from Homography



www.augmentedrealitybook.org

- Marker lies in a plane.
- Homography between image plane and world plane.
- Estimate the Homography using Direct Linear Transform (DLT).
- Recover camera pose from the homography.

Homography Estimation

- Image plane, Π
- World plane, $\Pi': q_z = 0$
- Homogenous point

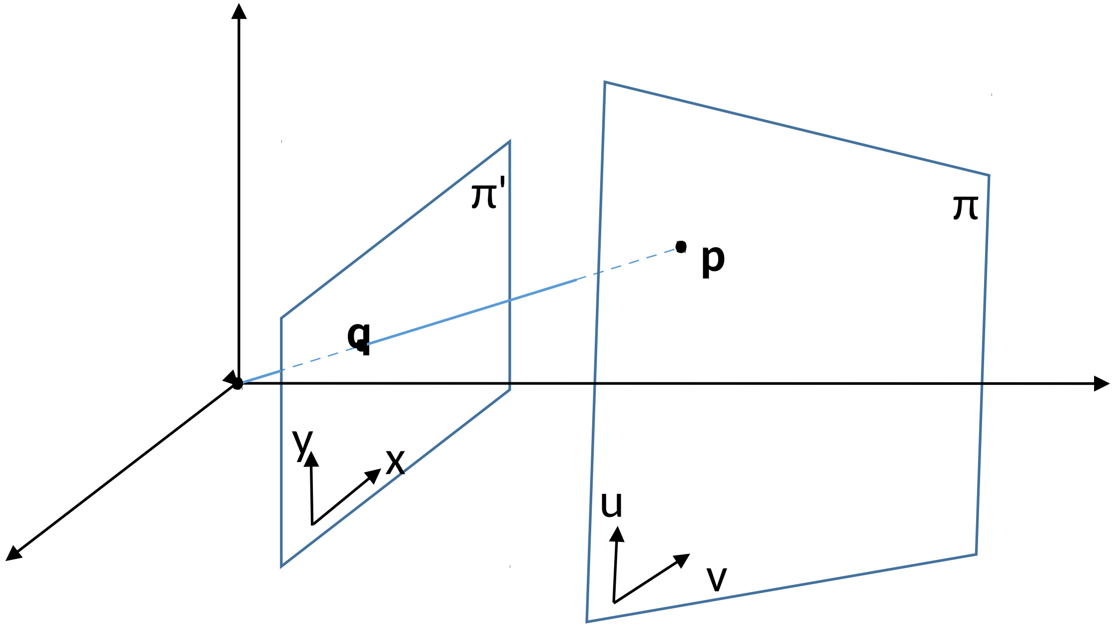
$$\boldsymbol{q}' = [q_x, q_y, 1]^T$$

- Homography

$$\boldsymbol{p} = \boldsymbol{H}\boldsymbol{q}'$$

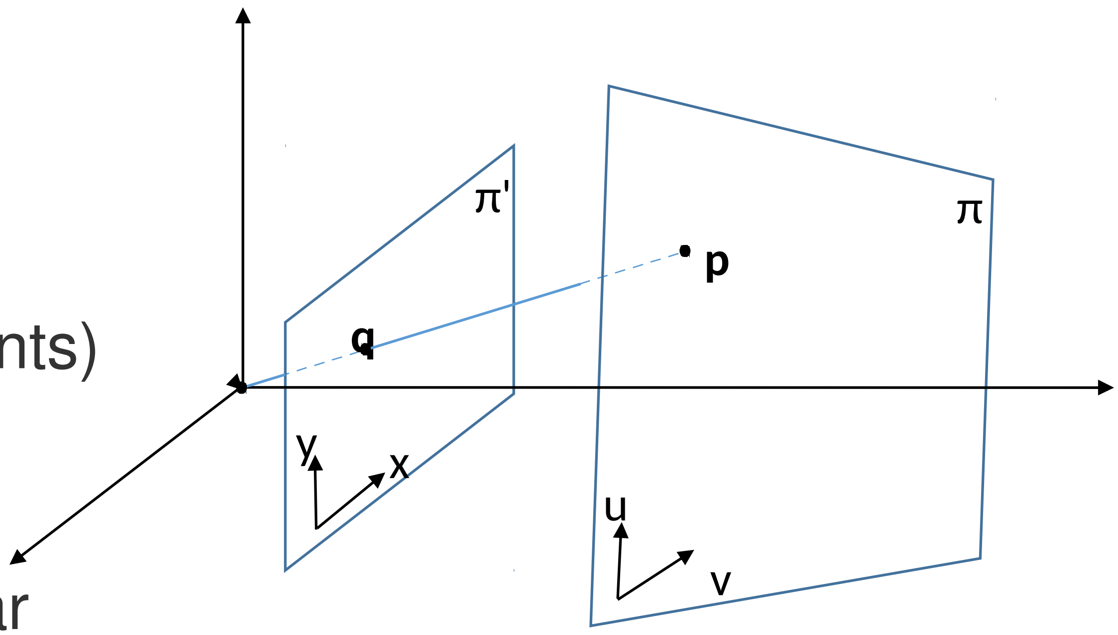
$$\boldsymbol{p} \times \boldsymbol{H}\boldsymbol{q}' = 0$$

$$\boldsymbol{p}_x = \begin{bmatrix} 0 & -p_w & p_v \\ p_w & 0 & -p_u \\ -p_v & p_u & 0 \end{bmatrix}, \boldsymbol{p}_x \begin{bmatrix} \boldsymbol{H}_{R1} \\ \boldsymbol{H}_{R2} \\ \boldsymbol{H}_{R3} \end{bmatrix} \boldsymbol{q}' = 0$$



www.augmentedrealitybook.org

Homography Estimation (DLT)



www.augmentedrealitybook.org

- $\boldsymbol{A}\,\boldsymbol{h}=0$ (A is 2N x 9 for N corresponding points)
- $\boldsymbol{A}=\boldsymbol{U}\,\boldsymbol{D}\,\boldsymbol{V}^{\boldsymbol{T}}$
- $\boldsymbol{h}$ is the smallest singular vector, i.e., the last column of $\boldsymbol{V}$
- This solution minimizes $\|\boldsymbol{A}\,\boldsymbol{h}\|$ subject to $\|\boldsymbol{h}\|=1$
- *Normalize the data before doing this.

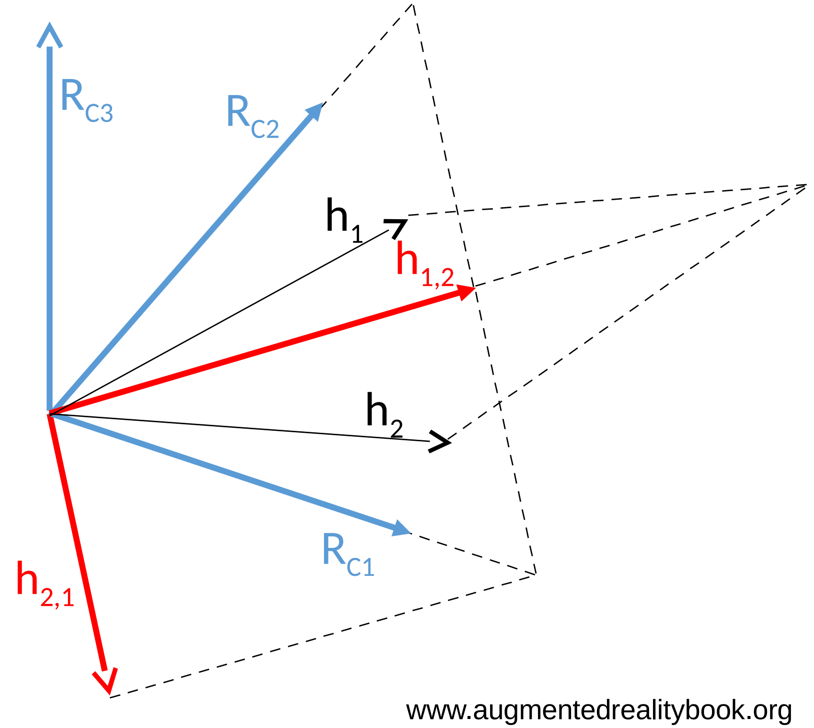
Camera Pose Estimation

$$\boldsymbol{H}=\boldsymbol{K}[\boldsymbol{R}_{C1}|\boldsymbol{R}_{C2}|\boldsymbol{t}]$$

$$\boldsymbol{H}^{K}=\boldsymbol{K}^{-1}\boldsymbol{H}$$

$$d=1/\sqrt{|\boldsymbol{H}_{C1}^{K}|.|\boldsymbol{H}_{C2}^{K}|}$$

$$\boldsymbol{t}=d\,\boldsymbol{H}_{C3}^{K}$$



$$\boldsymbol{h}_1=d\,\boldsymbol{H}_{C1}^{k} \qquad \boldsymbol{h}_2=d\,\boldsymbol{H}_{C2}^{k}$$

$$\boldsymbol{h}_{1,2}=\underline{N}(\boldsymbol{h}_1+\boldsymbol{h}_2) \qquad \boldsymbol{h}_{2,1}=\underline{N}(\boldsymbol{h}_{1,2}\times(\boldsymbol{h}_1\times\boldsymbol{h}_2))$$

$$\boldsymbol{R}_{C1}=(\boldsymbol{h}_{1,2}+\boldsymbol{h}_{2,1})/\sqrt{2} \qquad \boldsymbol{R}_{C2}=(\boldsymbol{h}_{1,2}-\boldsymbol{h}_{2,1})/\sqrt{2}$$

$$\boldsymbol{R}_{C3}=R_{C1}\times R_{C2}$$

Camera Pose Estimation

$$H = K[R_{C1}|R_{C2}|t]$$

$$H^K = K^{-1} H$$

$$d = 1/\sqrt{|H^K_{C1}|.|H^K_{C2}|}$$

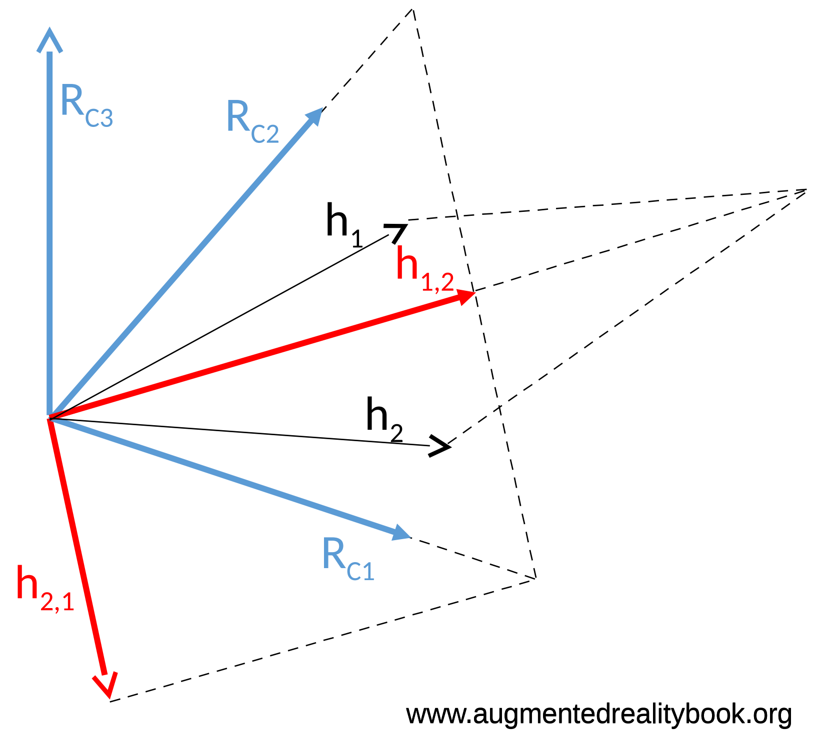
$$t = d\, H^K_{C3}$$

$$h_1 = d\, H^k_{C1} \qquad h_2 = d\, H^k_{C2}$$

$$h_{1,2} = \underline{N}(h_1 + h_2) \qquad h_{2,1} = \underline{N}(h_{1,2} \times (h_1 \times h_2))$$

$$R_{C1} = (h_{1,2} + h_{2,1})/\sqrt{2} \qquad R_{C2} = (h_{1,2} - h_{2,1})/\sqrt{2}$$

$$R_{C3} = R_{C1} \times R_{C2}$$



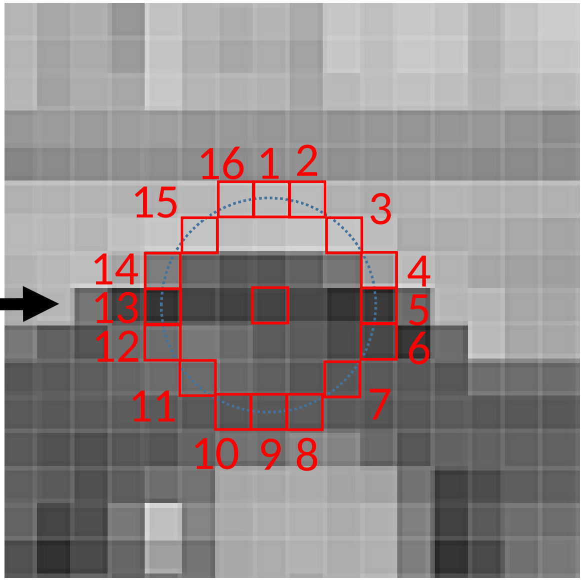
Is this solution unique?

Marker-less Tracking

- Make a model of the world before tracking and fit the model to detected feature points.
- Make a model of the world during tracking
  - SLAM – Simultaneous Localization and Mapping
    - PTAM – Parallel Tracking and Mapping (ISMAR2007)
    - KinectFusion (UIST2011)

PTAM

- FAST Corners



FAST searches for a contiguous sequence of pixels on a circle, which are consistently lighter or darker than the center.

Early exit can be found by first testing the pixels at the top, bottom, left, and right (right image).

Often, an improved detection method based on machine learning and a precompiled decision tree algorithm is used, allowing better generalization for arc lengths smaller than 12 pixels.

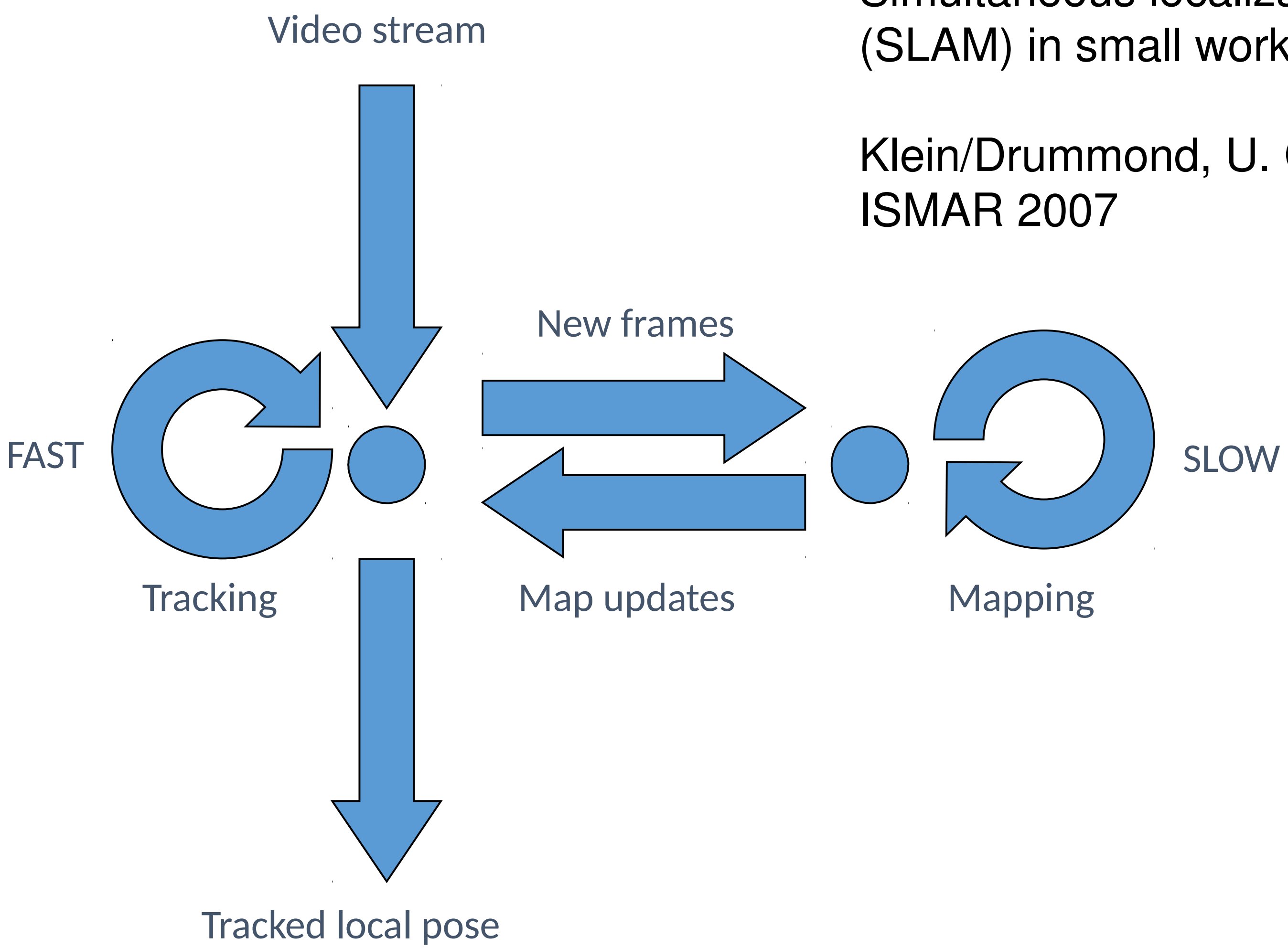
Image: Gerhard Reitmayr

www.augmentedrealitybook.org

PTAM

Simultaneous localization and mapping (SLAM) in small workspaces

Klein/Drummond, U. Cambridge
ISMAR 2007



www.augmentedrealitybook.org

PTAM

- Standard SLAM: Repeat until tracking is lost
  - Extract features from live image (or track features in image)
  - Match features to existing map
  - Determine camera pose from matched 3D points
  - Try to triangulate new features to get new 3D points
  - Insert any new 3D points into map (or update existing map points)
- Keyframe SLAM
  - Build map only from selected keyframes
  - Split tracking and mapping into two threads
  - Tracking at framerate, mapping at slower rate

KinectFusion

- Kinect for Windows SDK, Microsoft