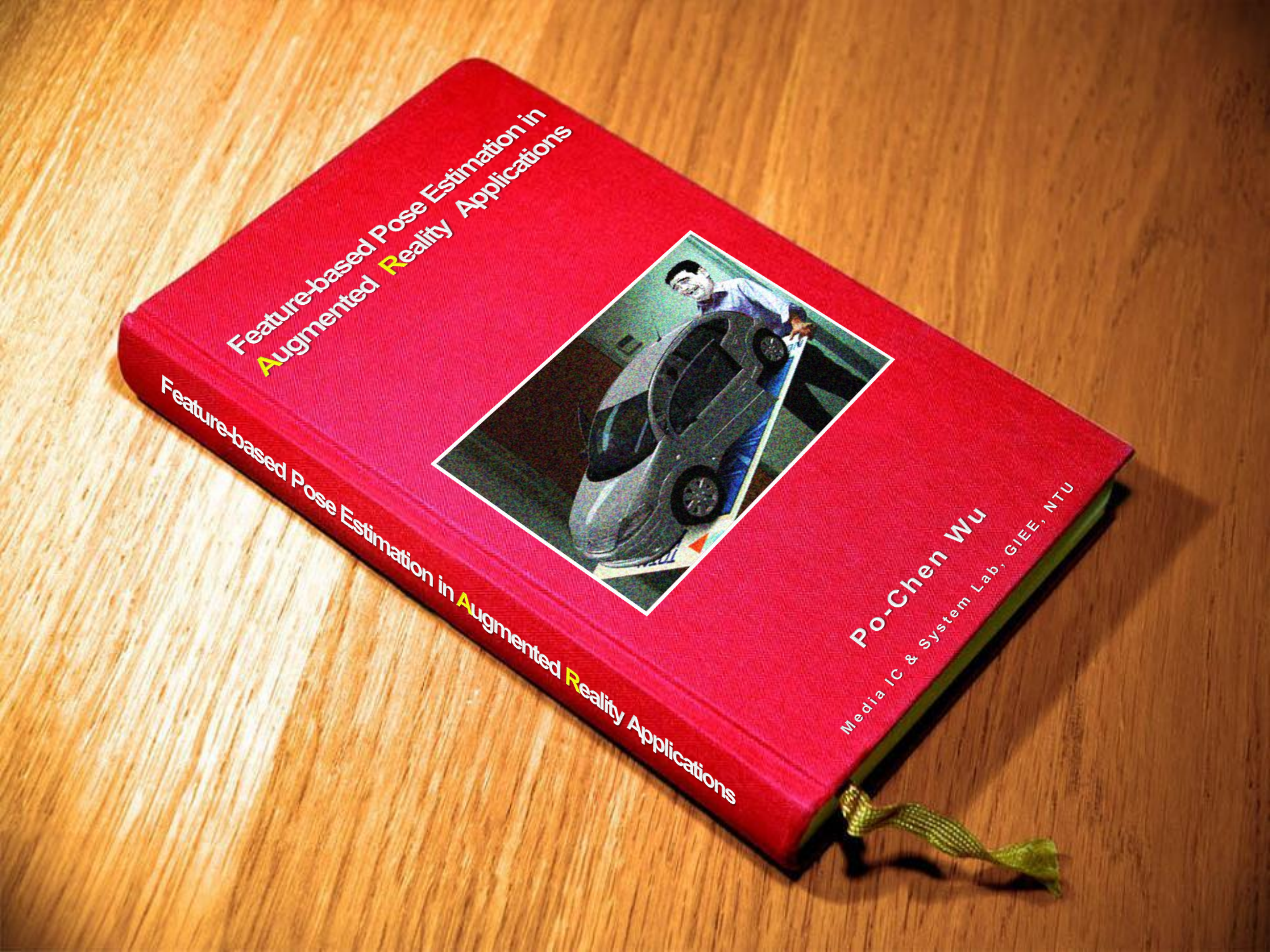

Feature-based Pose Estimation in
Augmented Reality Applications
Feature-based Pose Estimation in Augmented Reality Applications
Po-Chen Wu
Media IC & System Lab, GIEE, NTU

Outline

- Introduction
- System Overview
- Feature Detection & Matching
- Outlier Removal
- Pose Estimation
- Experimental Results
- Conclusion

Outline

- Introduction
- System Overview
- Feature Detection & Matching
- Outlier Removal
- Pose Estimation
- Experimental Results
- Conclusion

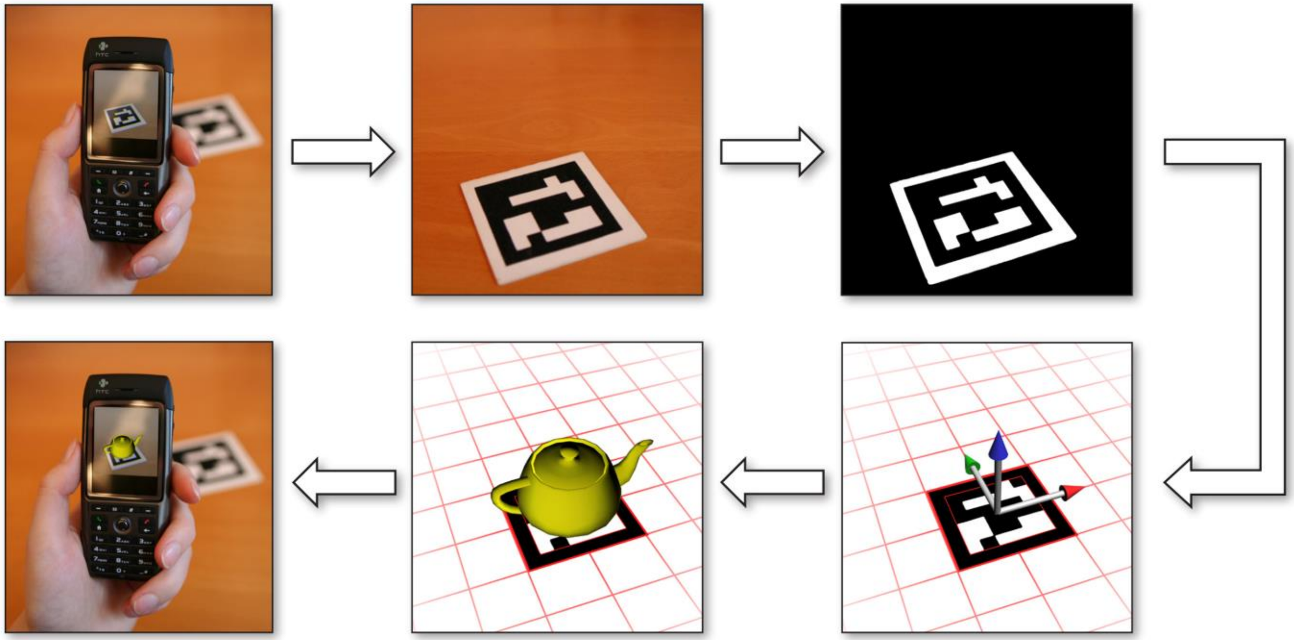
Introduction

- R. T. Azuma[1] define an Augmented Reality (AR) system to have the following properties:
  1) Combines real and virtual
  2) Interactive in real time
  3) Registered in 3-D

[1] Azuma, Ronald T. "A survey of augmented reality." Presence 6.4 (1997): 355-385.

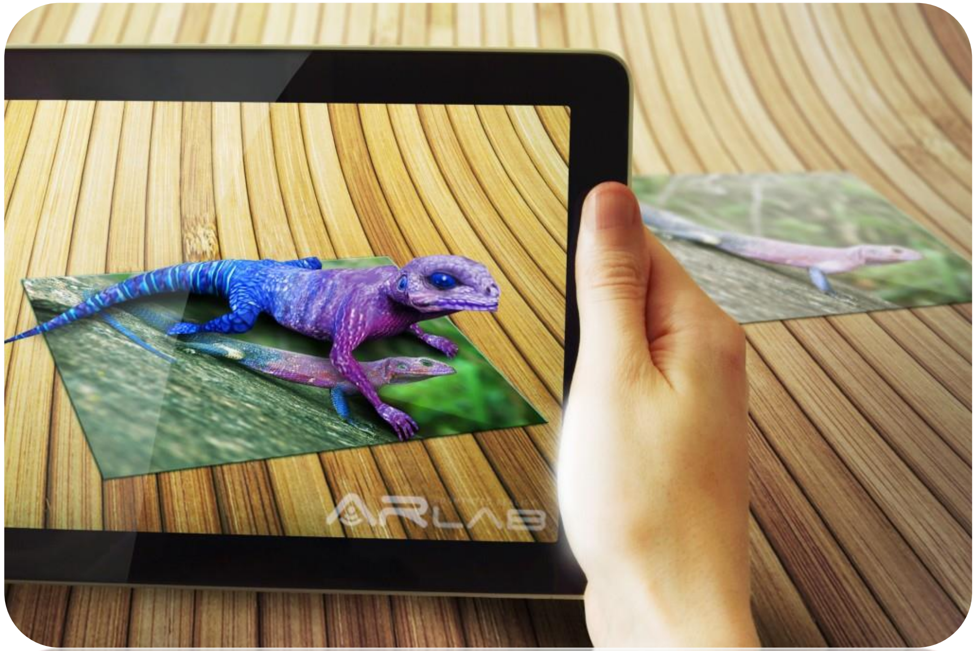
Marker-based AR

- Basic workflow of an AR application using fiducial marker tracking:



Markerless AR

- Markerless augmented reality systems rely on natural features instead of fiducial marks.



Features

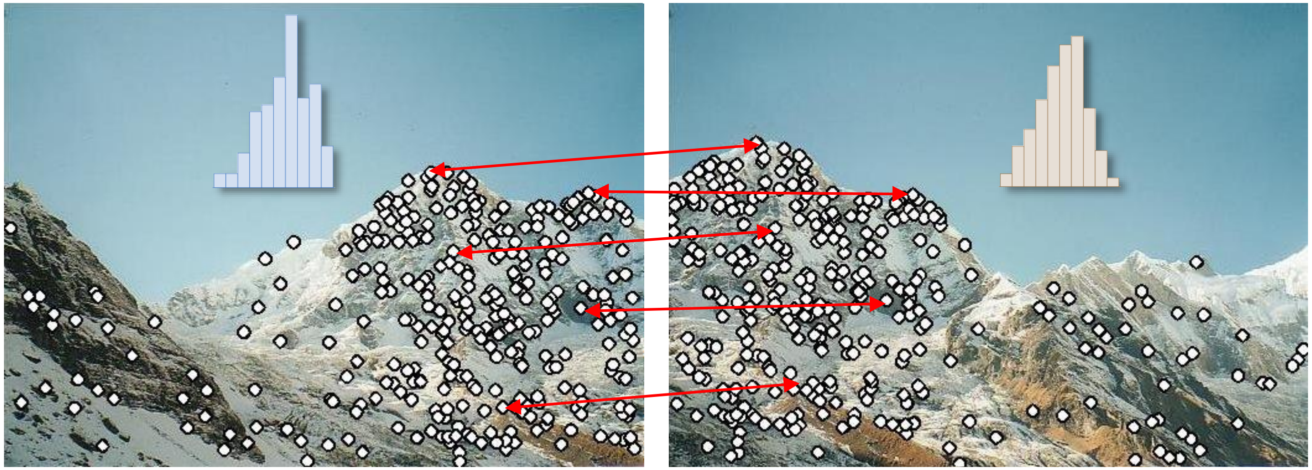
- Also known as interesting points, salient points or keypoints.
- Points that you can easily point out their correspondences in multiple images using only local information.

Desired Properties for Features

- Distinctive: a single feature can be correctly matched with high probability.
- Invariant: invariant to scale, rotation, affine, illumination and noise for robust matching across a substantial range of affine distortion, viewpoint change and so on. That is, it is <u>repeatable</u>.

Components

- Feature detection locates where they are.
- Feature description describes what they are.
- Feature matching decides whether two are the same one.



Detectors & Descriptors

- Feature detector:
  - DoG, SURF, FAST, AGAST, Multi-scale AGAST, etc.

- Feature descriptor:
  - SIFT, SURF, BRIEF, ORB, BRISK, FREAK, etc.

Another Markerless AR

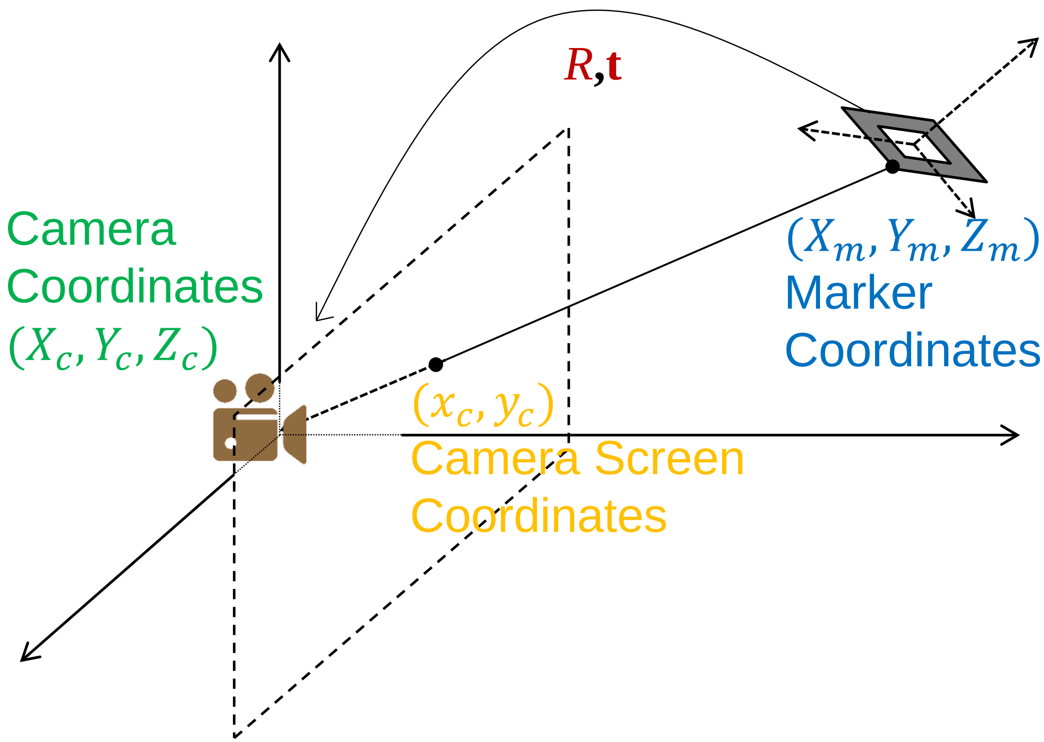
- Complete registration with GPS, inertial sensors, and magnetic sensors.

Outline

- Introduction
- System Overview
- Feature Detection & Matching
- Outlier Removal
- Pose Estimation
- Experimental Results
- Conclusion

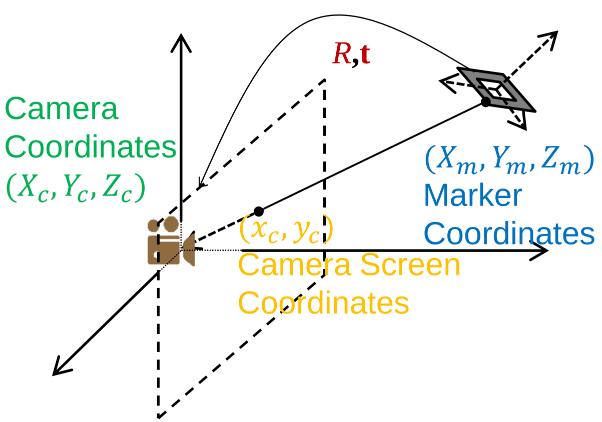
Marker-based AR

Camera Calibration → Marker Tracking → Pose Estimation → Rendering



Coordinate Systems

- The relationships among the camera screen coordinates (x_c, y_c), the camera coordinates (X_c, Y_c, Z_c) and the marker coordinates (X_m, Y_m, Z_m) can be represented as below:



$$\begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = P \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = P \cdot T_{\mathrm{cm}} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = C \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

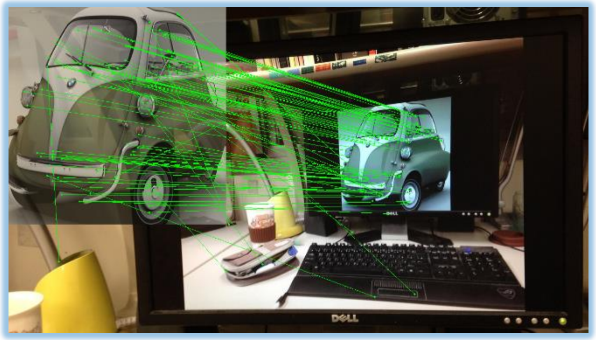
$$P = \begin{bmatrix} s_x f & 0 & x_0 & 0 \\ 0 & s_y f & y_0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{\mathrm{cm}} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

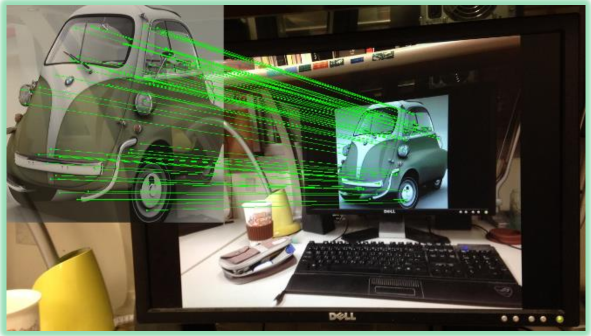
Intrinsic Matrix

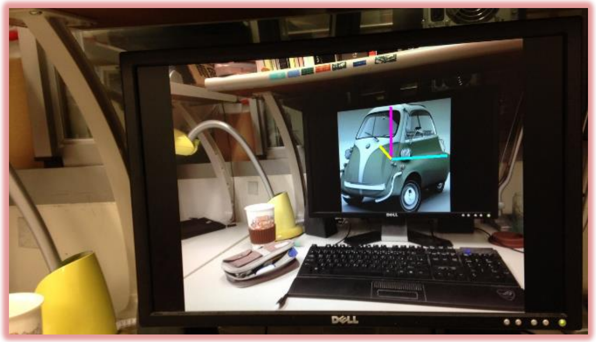
Extrinsic Matrix

Markerless AR

- Feature Detection & Matching
- Outlier Removal
- Pose Estimation
- Rendering





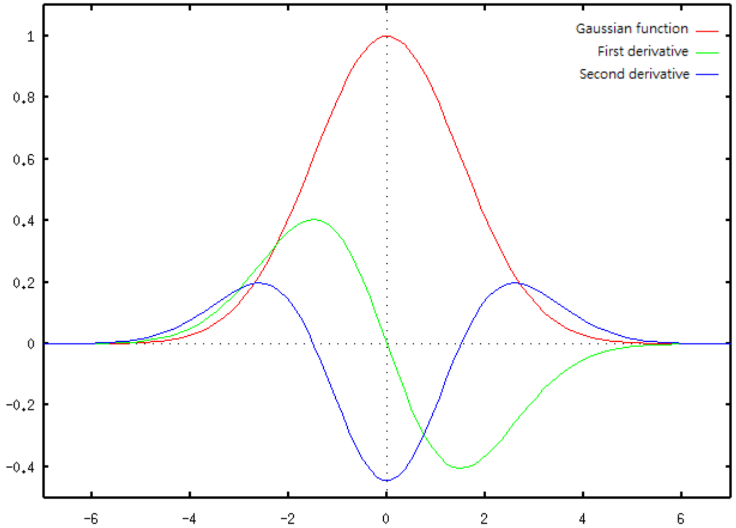


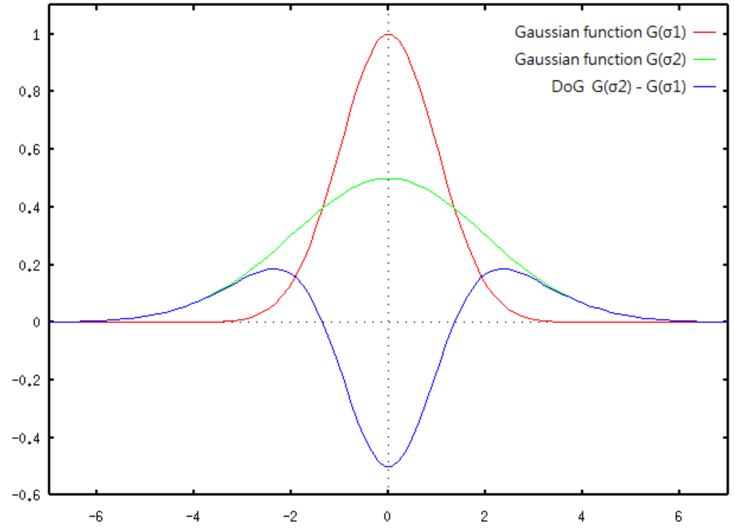
Outline

- Introduction
- System Overview
- Feature Detection & Matching
- Outlier Removal
- Pose Estimation
- Experimental Results
- Conclusion

Detector : DoG (1/2)

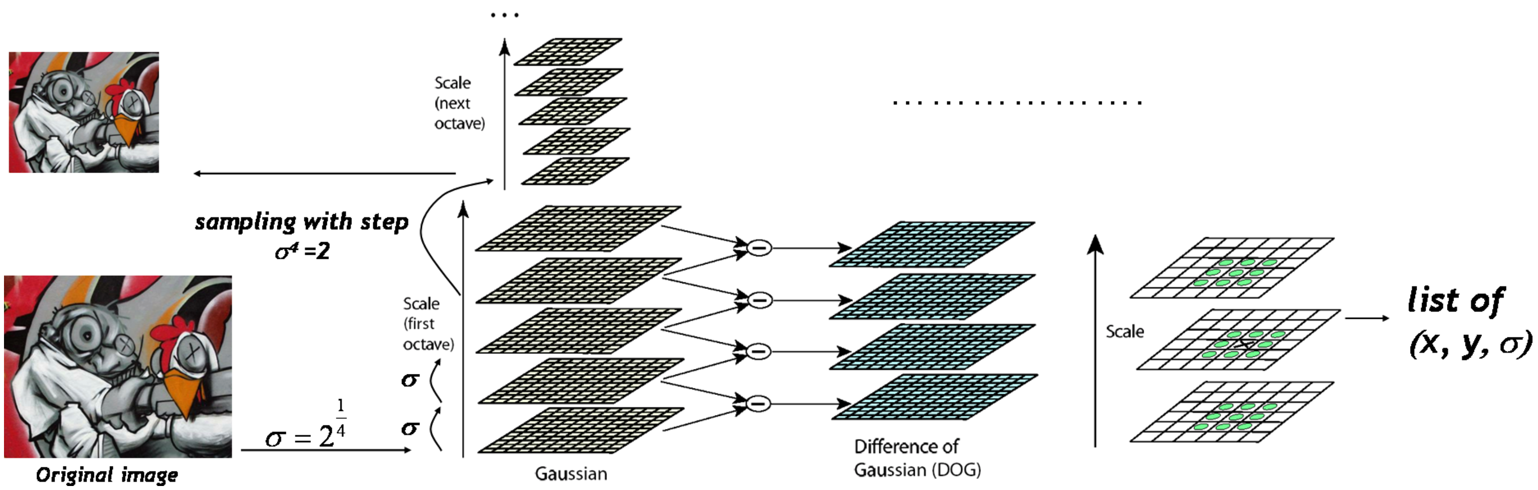
- Difference of Gaussian (DoG)
  - Used in Scale-Invariant Feature Transform (SIFT).
  - It is a scale-invariant detector which extracts blobs in the image by approximating the Laplacian of Gaussian $L_{xx}^2 + L_{yy}^2$.





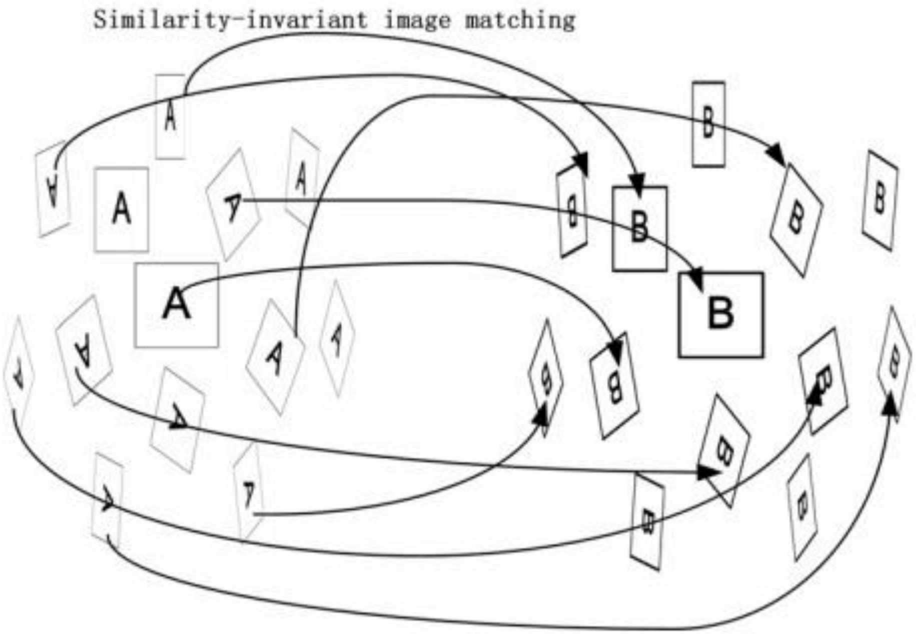
Detector : DoG (2/2)

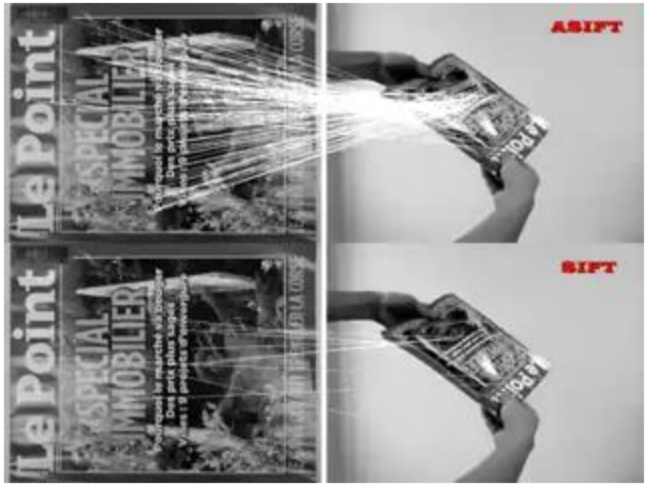
- Scale invariant



Detector: ASIFT

- Affine-SIFT (ASIFT) simulates all distortions caused by a variation of the camera optical axis direction.

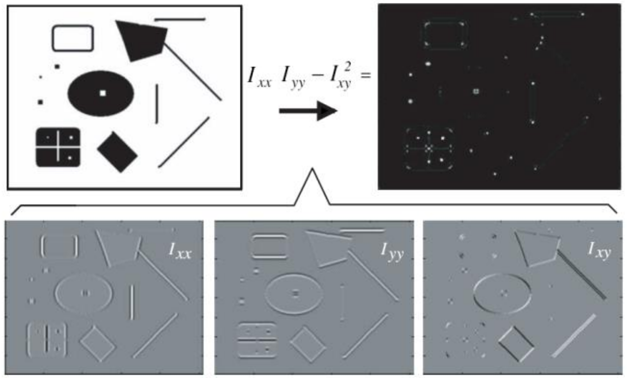




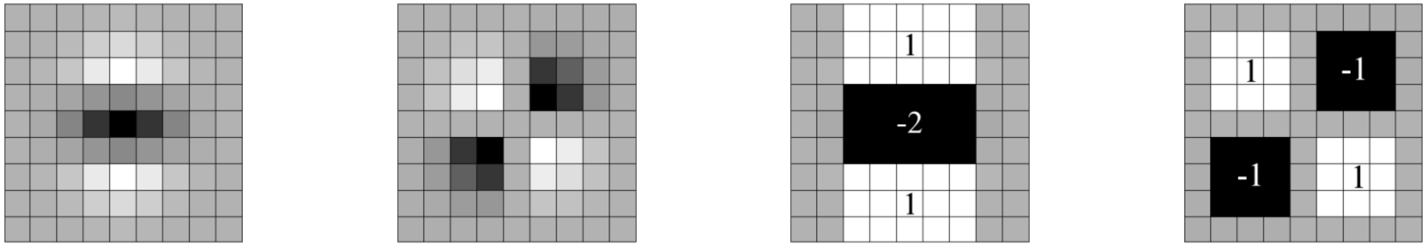
Detector : SURF

- Speeded Up Robust Features (SURF)
  - Based on the Hessian matrix (used in Hessian detector)

$$H = \begin{bmatrix} I_{xx}(\mathbf{x}, \sigma_D) & I_{xy}(\mathbf{x}, \sigma_D) \\ I_{xy}(\mathbf{x}, \sigma_D) & I_{yy}(\mathbf{x}, \sigma_D) \end{bmatrix}$$

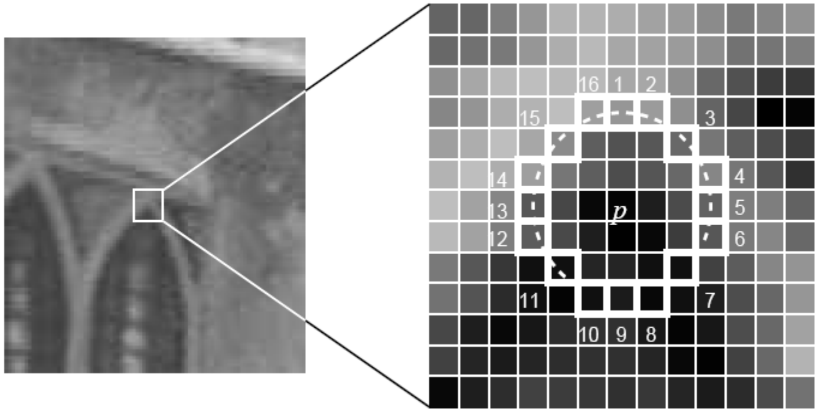


  - Can be evaluated very fast using integral images, independently of their size.



Detector : FAST (1/3)

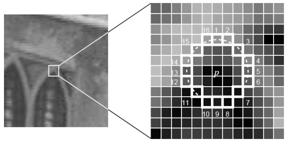
- Features from Accelerated Segment Test (FAST)
  - The original detector classifies p as a corner if there exists a set of n contiguous pixels in the circle which are all brighter than the intensity of the candidate pixel $I_p + t$, or all darker than $I_p - t$.



Detector : FAST (2/3)

- For each location on the circle x ∈ {1…16}, the pixel at that position relative to p (denoted by p→ x) can have one of three states:

$$S_{p\to x} = \begin{cases} d, & I_{p\to x} \le I_p - t \quad \text{(darker)} \\ s, & I_p - t < I_{p\to x} < I_p + t \quad \text{(similar)} \\ b, & I_p + t \le I_{p\to x} \quad \text{(brighter)} \end{cases}$$



- The entropy of the set P is:

$$H(P) = (c + \bar{c})\log_2(c + \bar{c}) - c\log_2 c - \bar{c}\log_2 \bar{c}$$

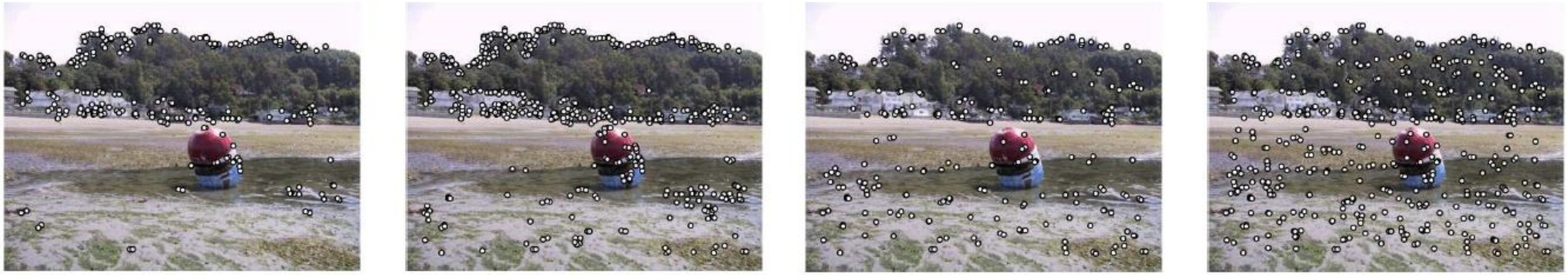
$$\text{where} \quad c = \left|\{p | K_p \text{ is true}\}\right| \quad \text{(number of corners)}$$

$$\text{and} \quad \bar{c} = \left|\{p | K_p \text{ is false}\}\right| \quad \text{(number of non corners)}$$

  – Information gain: $\boxed{H(P) - H(P_d) - H(P_s) - H(P_b)}$

Detector : FAST (3/3)

- Non-maximal suppression



(a) Strongest 250 (b) Strongest 500 (c) ANMS 250, $r = 24$ (d) ANMS 500, $r = 16$

  - Segment test does not compute a corner response function.
  - A score function, V must be computed for each detected corner.

$$V = \max\left(\sum_{x \in S_{\text{bright}}} |I_{p \to x} - I_p| - t \ , \ \sum_{x \in S_{\text{dark}}} |I_p - I_{p \to x}| - t \right)$$

Detector : AGAST

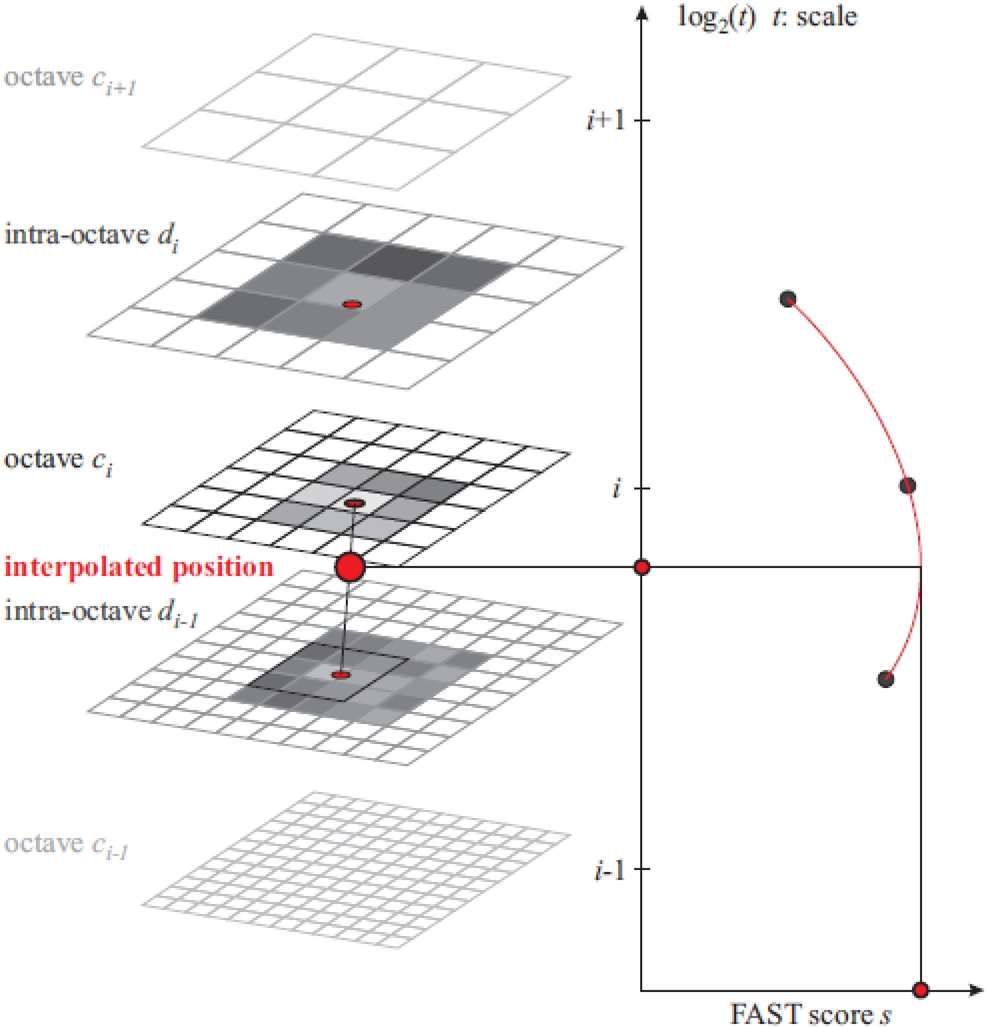
- Adaptive and Generic Corner Detection Based on the Accelerated Segment Test (AGAST)
  - binary decision tree

$$S_{n\to x} = \begin{cases} d, & I_{n\to x} < I_n - t & \text{(darker)} \\ \overline{d}, & I_{n\to x} \nless I_n - t \wedge S'_{n\to x} = u & \text{(not darker)} \\ s, & I_{n\to x} \nless I_n - t \wedge S'_{n\to x} = \overline{b} & \text{(similar)} \\ s, & I_{n\to x} \ngtr I_n + t \wedge S'_{n\to x} = \overline{d} & \text{(similar)} \\ \overline{b}, & I_{n\to x} \ngtr I_n + t \wedge S'_{n\to x} = u & \text{(not brighter)} \\ b, & I_{n\to x} > I_n + t & \text{(brighter)} \end{cases}$$

  - Using this dynamic programming technique allows us to find the decision tree for an optimal AST (OAST).

Detector : Multi-scale AGAST

- Scale-space interest point detection
  - Points of interest are identified across both the image and scale dimensions using a saliency criterion.



Descriptor: SIFT

- Scale-Invariant Feature Transform (SIFT)
  - Computed for normalized 16*16 image patches.

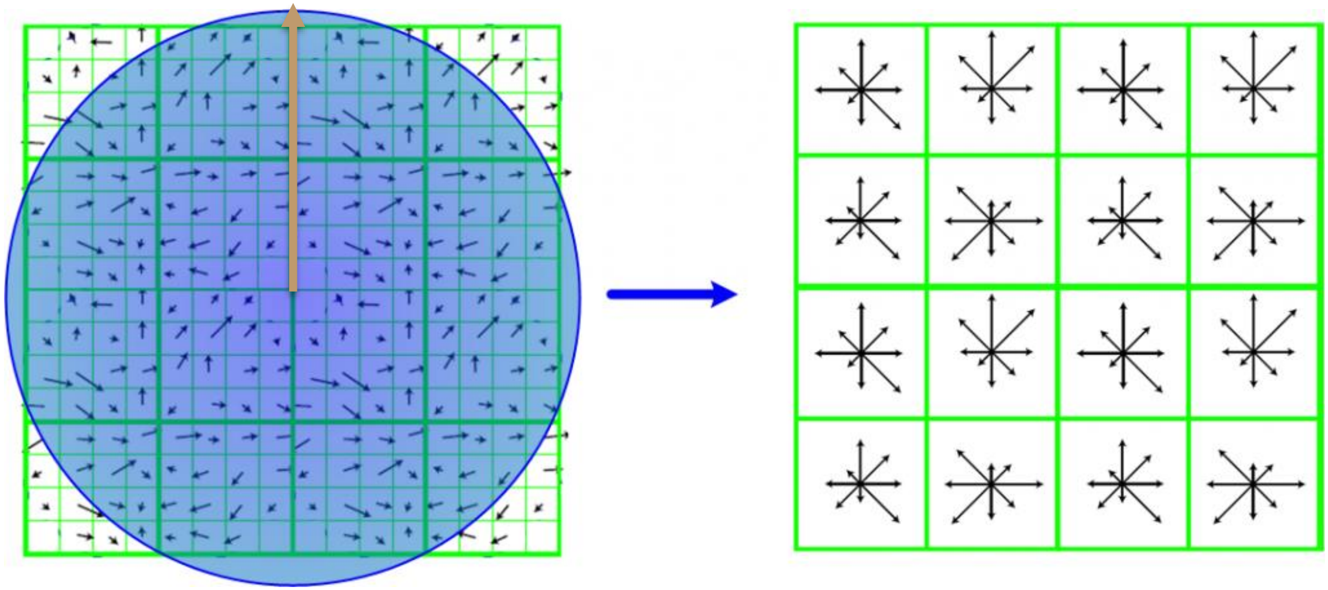
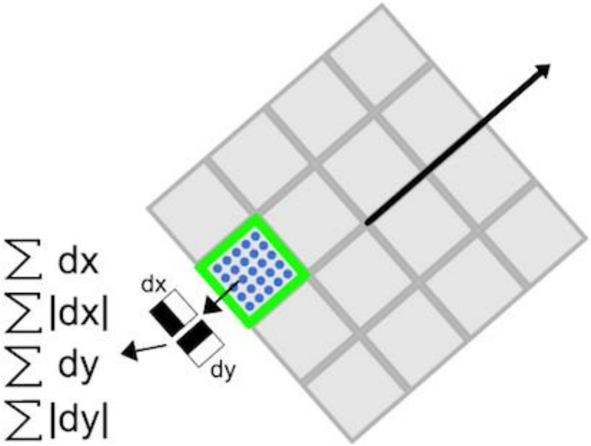


Image gradients

Keypoint descriptor

Descriptor: SURF

- Speeded Up Robust Features (SURF)
  - For each sub-region, we compute Haar wavelet responses at 5×5 regularly spaced sample points.
    - dx is the Haar wavelet response in horizontal direction.
    - dy is the Haar wavelet response in vertical direction.



Descriptor: BRIEF (1/2)

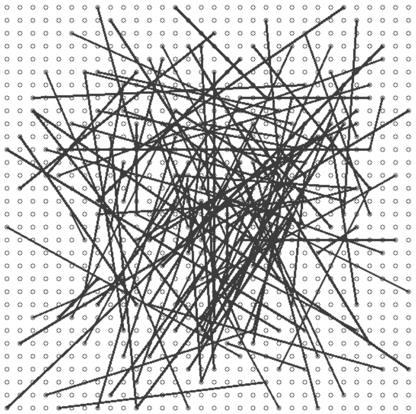
- We define test τ on patch $\mathbf{p}$ of size $S \times S$ as

$$\tau(\mathbf{p};\mathbf{x},\mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases}$$

  – where $\mathbf{p}(\mathbf{x})$ is the pixel intensity in a smoothed version of $\mathbf{p}$ at $\mathbf{x} = (u, v)$T.

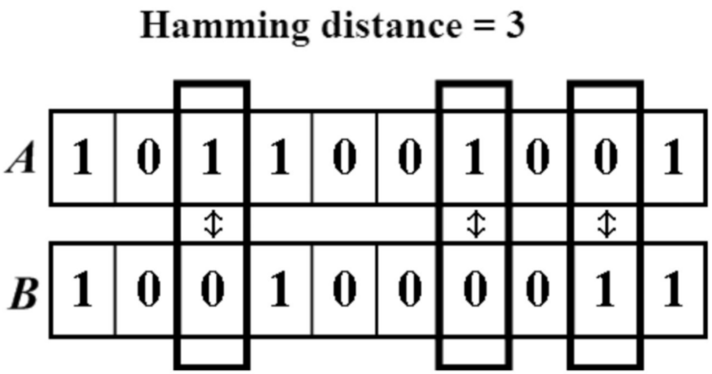
- Choosing a set of n_d (x, y)-location pairs uniquely defines a set of binary tests.
  – We take our BRIEF descriptor to be the n_d-dimensional bitstring

$$f_{n_d}(\mathbf{p}) := \sum_{1 \le i \le n_d} 2^{i-1}\, \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i)$$



Descriptor: BRIEF (2/2)

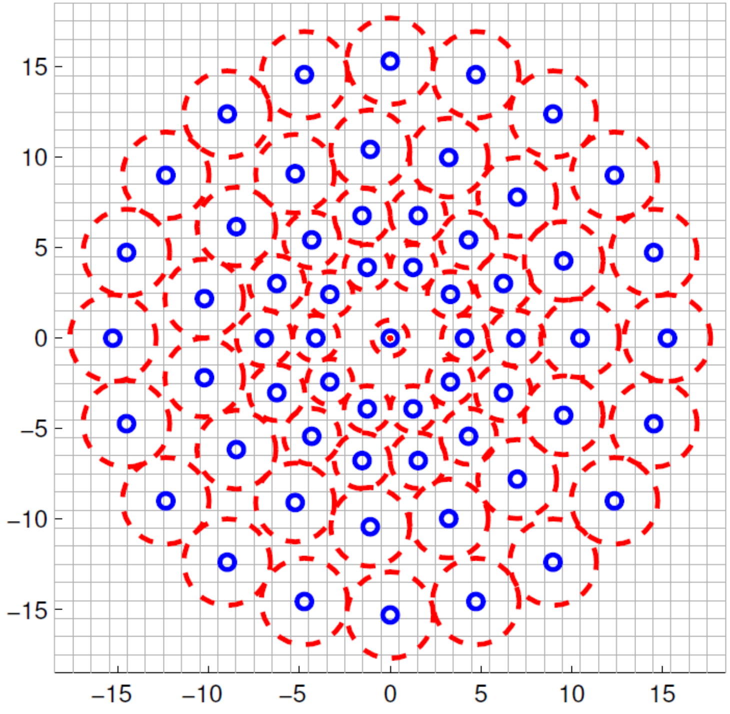
- Comparing strings can be done by computing the Hamming distance



  – It can be done extremely fast on modern CPUs that often provide a specific instruction to perform a XOR or bit count operation, as is the case in the latest SSE instruction set.

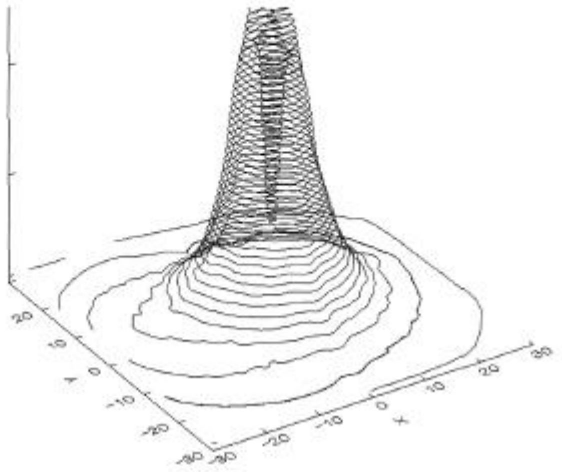
Descriptor: BRISK

- Binary Robust Invariant Scalable Keypoints (BRISK)
  - The BRISK sampling pattern with N = 60 points.

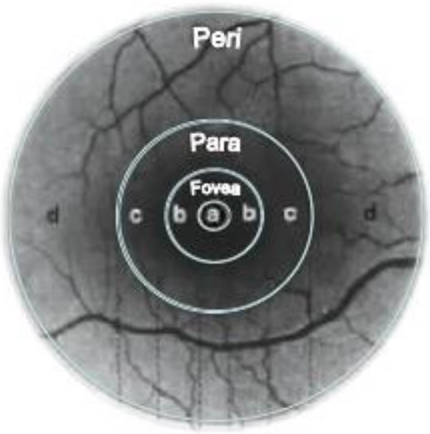


Descriptor: FREAK

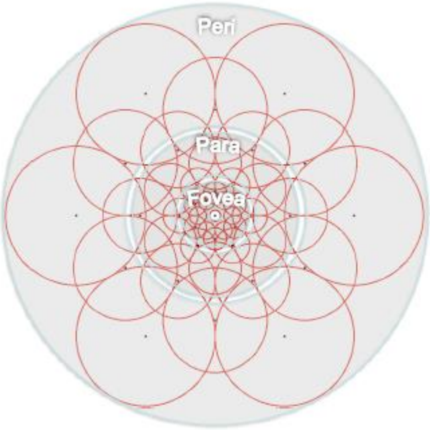
- Fast Retina Keypoint (FREAK)
  - A cascade of binary strings is computed by efficiently comparing image intensities over a retinal sampling pattern.



(a) Density of ganglion cells over the retina



(b) Retina areas

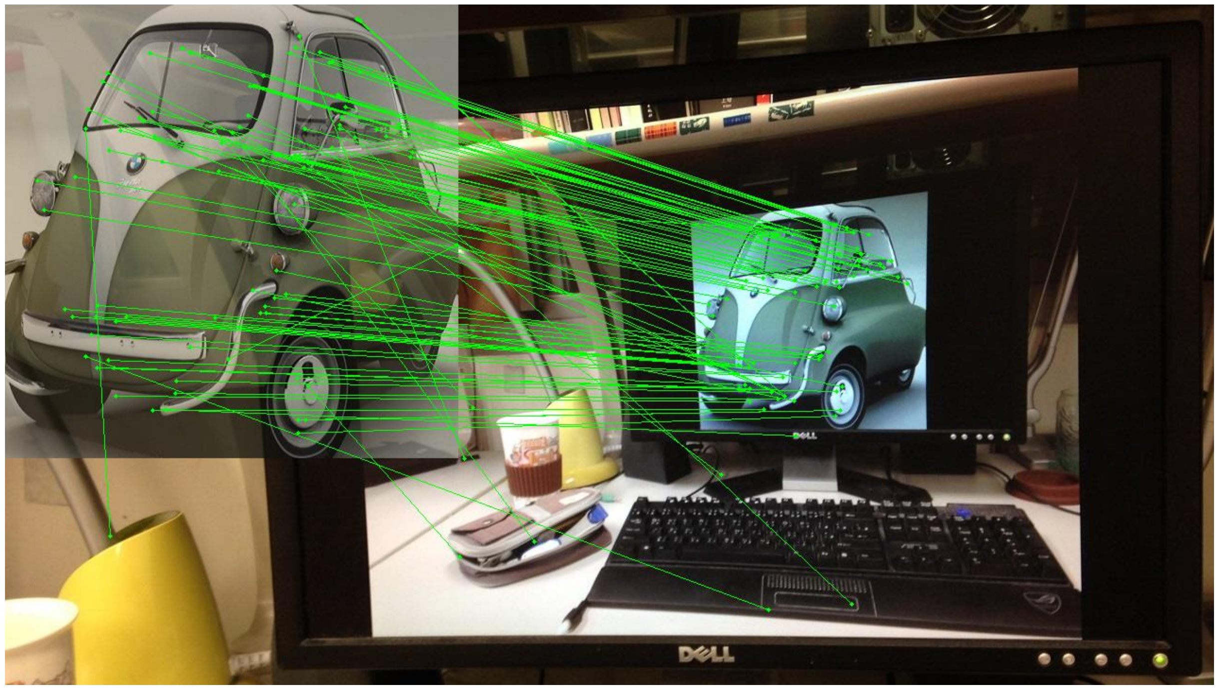


- retinal sampling pattern

Outline

- Introduction
- System Overview
- Feature Detection & Matching
- Outlier Removal
- Pose Estimation
- Experimental Results
- Conclusion

Outlier Removal



Tentative Correspondences
(inliers + outliers)

Reliable Correspondences
(inliers)

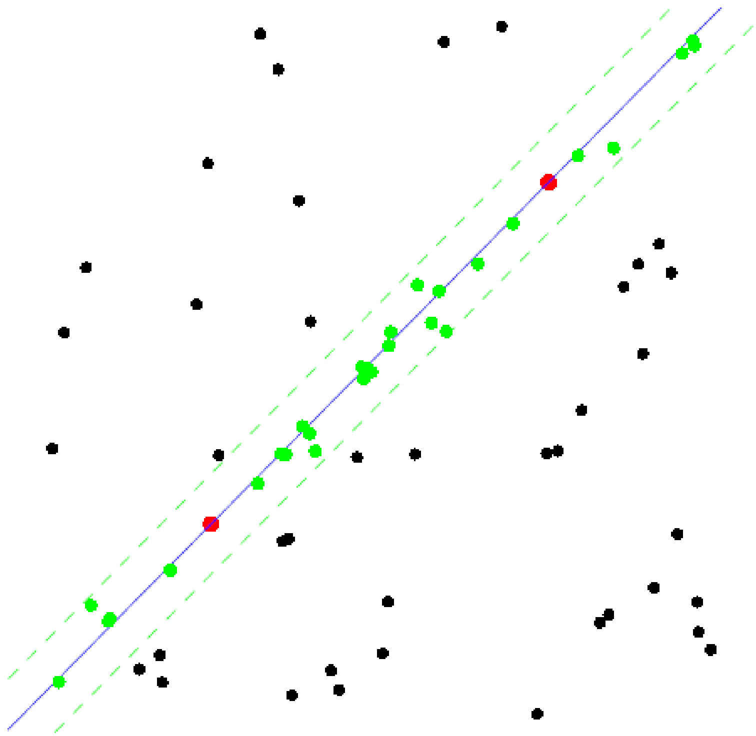
Related Papers

- [ACM Comm. 1981] Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. RANSAC
- [CVIU 2000] MLESAC: A new robust estimator with application to estimating image geometry. MLESAC
- [ECCV 2002] Guided sampling and consensus for motion estimation.
- [BMVC 2002 Napsac: High noise, high dimensional robust estimation - it’s in the bag.
- [ICCV 2003] Preemptive RANSAC for live structure and motion estimation. NAPSAC
- [CVPR 2005] Matching with PROSAC – Progressive Sample Consensus. PROSAC
- [ECCV 2008] A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. ARRSAC
- [ICCV 2013] EVSAC: Accelerating Hypotheses Generation by Modeling Matching Scores with Extreme Value Theory. EVSAC

[ACM Comm. 1981] Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography

RANSAC (1/3)

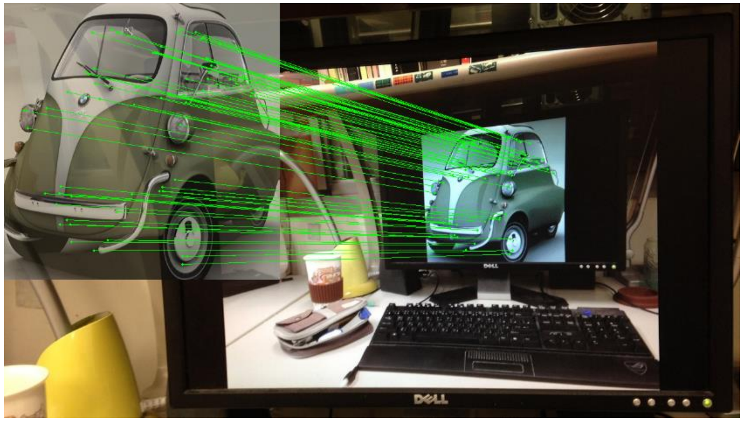
- Example: line fitting



- Pose estimation

$$\begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = \begin{bmatrix} s_x f & 0 & x_0 & 0 \\ 0 & s_y f & y_0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boxed{\begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



RANSAC (2/3)

- Random Sample Consensus (RANSAC)

Run k times: ← How many times?

(1) draw n samples randomly

(2) fit parameters Θ with these n samples

(3) for each of other N-n points, calculate its distance to the fitted model, count the number of inlier points, c

Output Θ with the largest c

RANSAC (3/3)

- How to determine k
  - P: probability of success after k trials
  - p: probability of real inliers

$$P = 1 - \underbrace{\underbrace{(1 - \underbrace{p^{n}}_{\text{n samples are all inliers}})}_{\text{a failure}}{}^{k}}_{\text{failure after k trials}}$$

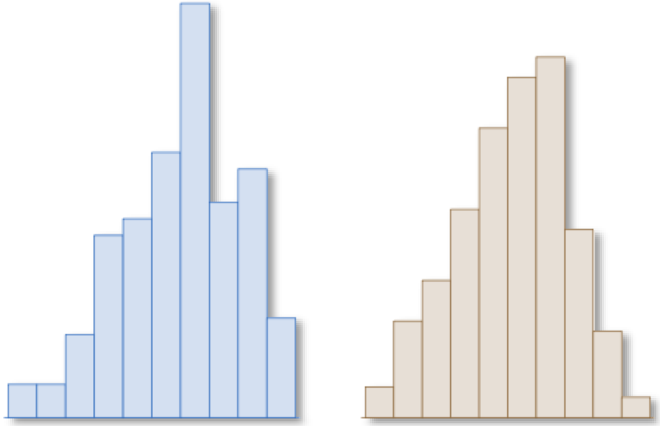
$$k = \frac{\log(1 - P)}{\log(1 - p^{n})}$$

for $P=0.99$

n	p	k
3	0.5	35
6	0.6	97
6	0.5	293

PROSAC (1/2)

- Progressive Sample Consensus (PROSAC)
  - The improvement in efficiency rests on the mild assumption that tentative correspondences with high similarity are more likely to be inliers.



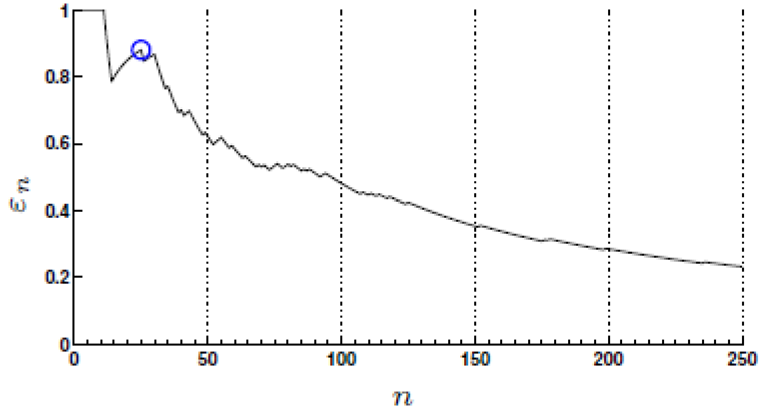
Euclidean distance

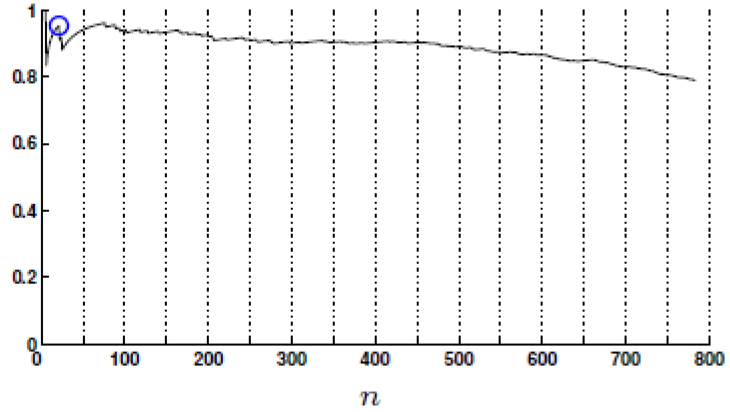
A	1	0	1	0	1	0	1	0	1
B	1	0	1	0	1	1	0	0	1

Hamming distance

PROSAC (2/2)

- The fraction of inliers ε among top n correspondences sorted by quality.





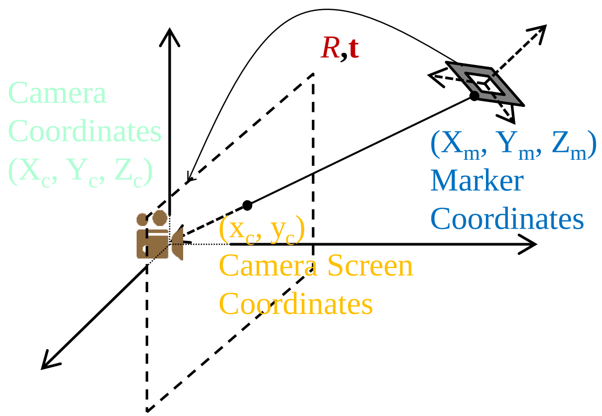
	k	min k	max k	time [sec]
RANSAC	106,534	97,702	126,069	10.76
PROSAC	9	5	29	0.06

Outline

- Introduction
- System Overview
- Feature Detection & Matching
- Outlier Removal
- Pose Estimation
- Experimental Results
- Conclusion

Introduction to PnP Problem

- The aim of the Perspective-n-Point (PnP) problem is to determine the position and orientation of a camera given its:
  1) intrinsic parameters
  2) a set of n correspondences between 3D points and their 2D projections.



$$\begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = \begin{bmatrix} s_x f & 0 & x_0 & 0 \\ 0 & s_y f & y_0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \underbrace{\boxed{\begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}}_{\text{Pose}} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

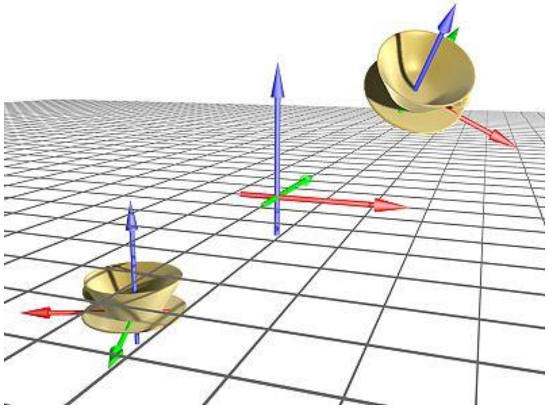
Related Papers

- [TPAMI 2000] Fast and Globally Convergent Pose Estimation from Video Images. OI
- [TPAMI 2006] Robust Pose Estimation from a Planar Target. RPP
- [BMVC 2008] Globally Optimal O (n) Solution to the PnP Problem for General Camera Models GOP
- [IJCV 2009] EPnP An Accurate O(n) Solution to the PnP Problem. EPnP
- [ICCV 2011] A direct least-squares (DLS) method for PnP. DLS
- [TPAMI 2012] A Robust O(n) Solution to the Perspective-n-Point Problem. RPnP
- [ICCV 2013] Revisiting the PnP Problem: A Fast, General and Optimal Solution. OPnP
- [ECCV 2014] UPnP: An Optimal O (n) Solution to the Absolute Pose Problem with Universal Applicability. UPnP
- [BMVC 2014] Leveraging Feature Uncertainty in the PnP Problem. CEPPnP
- [CVPR 2014] A General and Simple Method for Camera Pose and Focal Length Determination. GPnPf
- [CVPR 2014] Very Fast Solution to the PnP Problem with Algebraic Outlier Rejection. REPPnP

Camera Model (1/2)

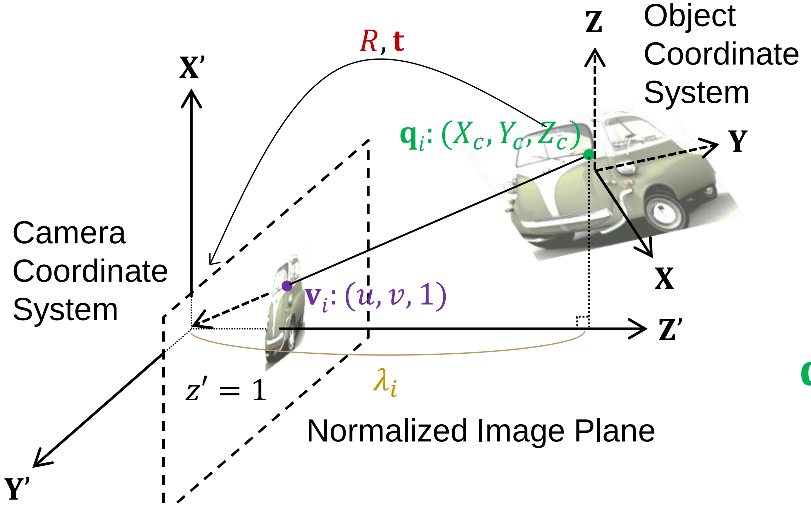
- Given a set of 3D coordinates of reference points $\mathbf{p}_i = (X_m, Y_m, Z_m)^t, i = 1, \dots, n, n \geq 3$, expressed in an object-centered reference frame, the corresponding camera-space coordinates $\mathbf{q}_i = (X_c, Y_c, Z_c)^t$, are related by a rigid transformation as $\mathbf{q}_i = R\mathbf{p}_i + \mathbf{t}$, where

$$R = \begin{pmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \\ \mathbf{r}_3^t \end{pmatrix} \in \mathcal{SO}(3), \quad \mathbf{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \in R^3$$



Camera Model (2/2)

- The reference points $\mathbf{p}_i$ are projected to the plane with $z' = 1$, referred to as the normalized image plane, in the camera reference frame.
  - Let the image point $\mathbf{v}_i = (u, v, 1)^t$ be the projection of $\mathbf{p}_i$ (or $\mathbf{q}_i$) on the normalized image plane.



$$\mathbf{q}_i = R\mathbf{p}_i + \mathbf{t} = \lambda_i \mathbf{v}_i$$

OI (1/5)

- Optimal Absolute Orientation Solution
  - If $\mathbf{q}_i$ could be reconstructed, then R and $\mathbf{t}$ in $\mathbf{q}_i = R\mathbf{p}_i + \mathbf{t}$ can be obtained as a solution to the following least-squares problem:

$$(R^*, \mathbf{t}^*) = \arg\min_{R,\mathbf{t}} \sum_{i=1}^{n} \|R\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|^2, \text{subject to } R^t R = I$$

  - The closed form solution of R^* and $\mathbf{t}^*$ would be:

$$\bar{\mathbf{p}} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_i, \bar{\mathbf{q}} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \mathbf{q}_i, \mathbf{p}_i' = \mathbf{p}_i - \bar{\mathbf{p}}, \mathbf{q}_i' = \mathbf{q}_i - \bar{\mathbf{q}}, M = \sum_{i=1}^{n} \mathbf{q}_i' \mathbf{p}_i'^t$$

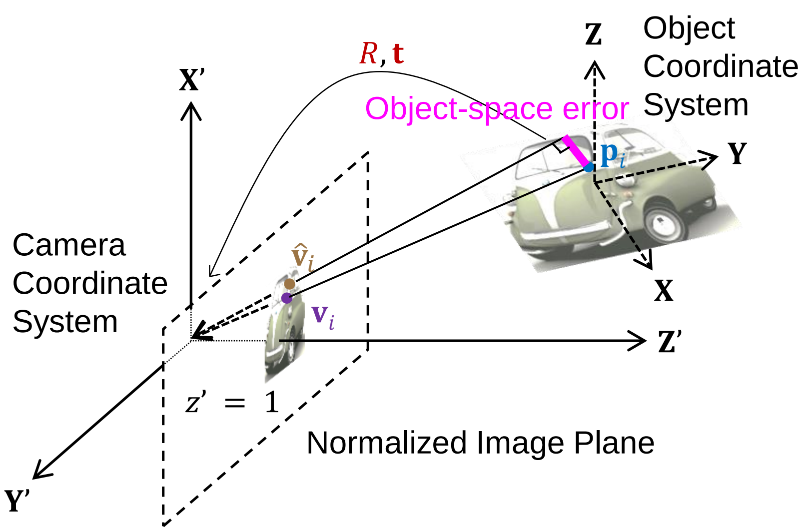
Use the SVD of M in the form $M = U\Sigma V^t \rightarrow R^* = VU^t, \mathbf{t}^* = \bar{\mathbf{q}} - R^*\bar{\mathbf{p}}$

OI (2/5)

- We then seek to minimize the sum of the squared error over R and $\mathbf{t}$:

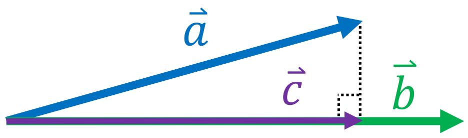
$$E_{os}(R,\mathbf{t}) = \sum_{i=1}^{n} \|\mathbf{e}_i\|^2 = \sum_{i=1}^{n} \left\|(I - \hat{V}_i)(R\mathbf{p}_i + \mathbf{t})\right\|^2, \hat{V}_i = \frac{\hat{\mathbf{v}}_i \hat{\mathbf{v}}_i^t}{\hat{\mathbf{v}}_i^t \hat{\mathbf{v}}_i}$$

Projection matrix



Orthogonal Projection

$$\vec{c} = \left(\frac{\vec{a} \cdot \vec{b}}{\left|\vec{b}\right|^2}\right)\vec{b}$$



OI (3/5)

- Since this objective function is quadratic in **t**, given a fixed rotation R, the optimal value for **t** can be computed in closed form as

$$\mathbf{t}(R) = \frac{1}{n}\left(I - \frac{1}{n}\sum_{i=1}^{n}\hat{V}_i\right)^{-1}\sum_{i=1}^{n}\left(\hat{V}_i - I\right)R\mathbf{p}_i$$

- Given the optimal translation as a function of R and defining $\mathbf{q}_i(R) \stackrel{\text{def}}{=} \hat{V}_i\left(R\mathbf{p}_i + \mathbf{t}\,(R)\right)$, then we reformulate the problem

$$(R^*, \mathbf{t}^*) = \arg\min_{R,\mathbf{t}} \sum_{i=1}^{n}\left\|(I - \hat{V}_i)(R\mathbf{p}_i + \mathbf{t})\right\|^2$$

$$R^* = \arg\min_{R} \sum_{i=1}^{n}\left\|(I - \hat{V}_i)\left(R\mathbf{p}_i + \mathbf{t}(R)\right)\right\|^2 = \arg\min_{R} \sum_{i=1}^{n}\left\|R\mathbf{p}_i + \mathbf{t}(R) - \mathbf{q}_i(R)\right\|^2$$

OI (4/5)

- R can be computed iteratively as follows:
  - Assume the k-th estimate of R is $R^{(k)}$, $\mathbf{t}^{(k)} = \mathbf{t}\left(R^{(k)}\right)$, and $\mathbf{q}_i^{(k)} = \hat{V}_i\left(R^{(k)}\mathbf{p}_i + \mathbf{t}^{(k)}\right)$.
  - Then
$$R^{(k+1)} = \arg\min_{R} \sum_{i=1}^{n} \left\| R\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i^{(k)} \right\|^2$$
  - We then compute the next estimate of $\mathbf{t}$ as $\mathbf{t}^{(k+1)} = \mathbf{t}\left(R^{(k+1)}\right)$
  - A solution R^*:
$$R^* = \arg\min_{R} \sum_{i=1}^{n} \left\| R\mathbf{p}_i + \mathbf{t} - \hat{V}_i\left(R^*\mathbf{p}_i + \mathbf{t}(R^*)\right) \right\|^2$$

It can be proved that $E\left(R^{(k+1)}\right) < E\left(R^{(k)}\right)$ until converging to an optimum.

OI (5/5)

- Initialization and Weak Perspective Approximation

$$\mathbf{q}_i \approx s\mathbf{v}_i = \left(sx_c, sx_y, s\right)^t$$

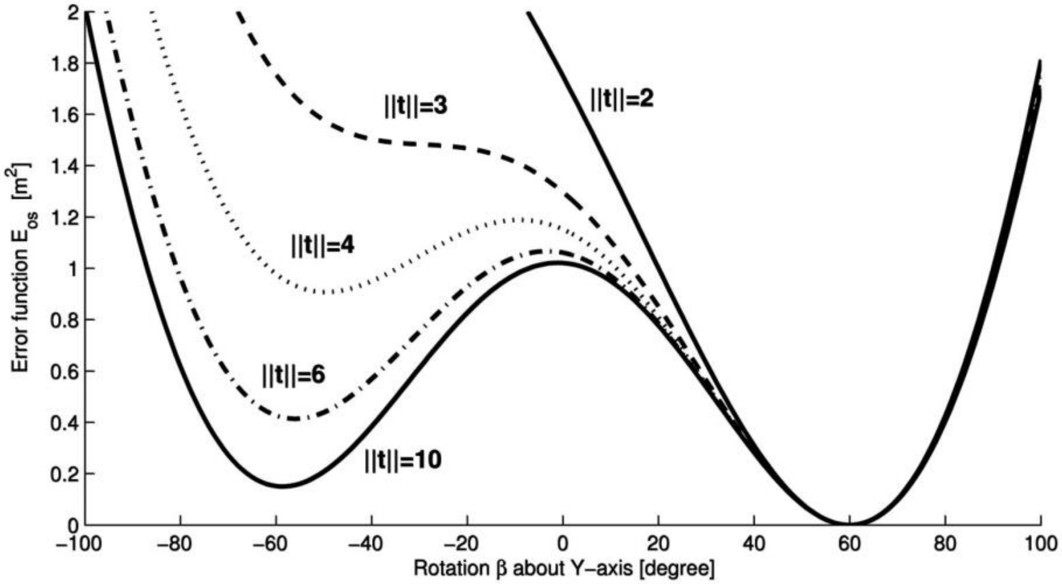
  - Where $s = \sqrt{\frac{\sum_{i=1}^{n}\|\mathbf{p}_i'\|^2}{\sum_{i=1}^{n}\|\mathbf{v}_i'\|^2}}$

$$\boxed{\bar{\mathbf{v}} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\mathbf{v}_i \ , \mathbf{v}_i' = \mathbf{v}_i - \bar{\mathbf{v}}}$$

- It’s called Orthogonal Iteration (OI) algorithm.

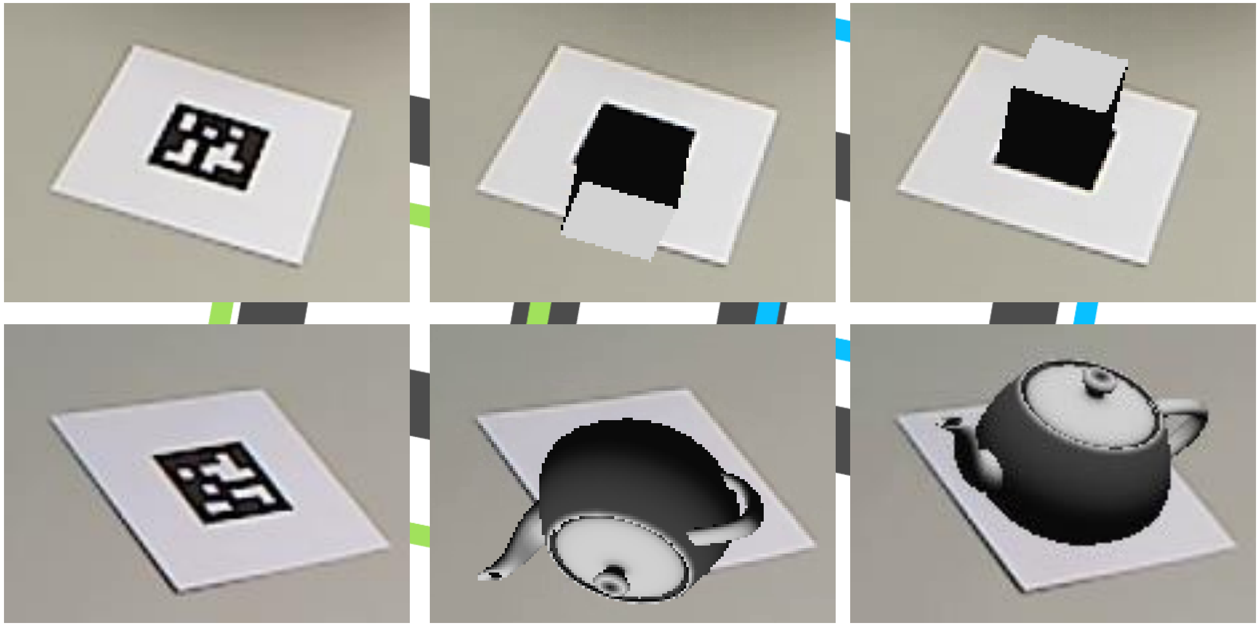
RPP (1/3)

- Definition of Pose Ambiguity:
  - Two distinct local minima of the according error function (e.g., E_{os}).



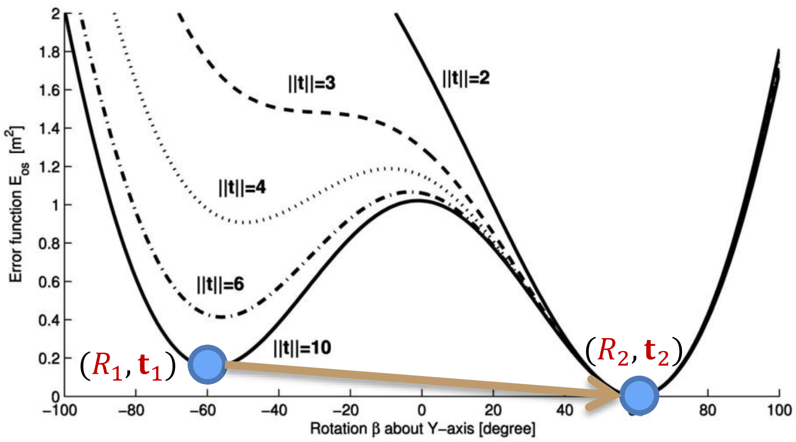
RPP (2/3)

- The cause of geometric illusions of computer vision is the pose ambiguity.



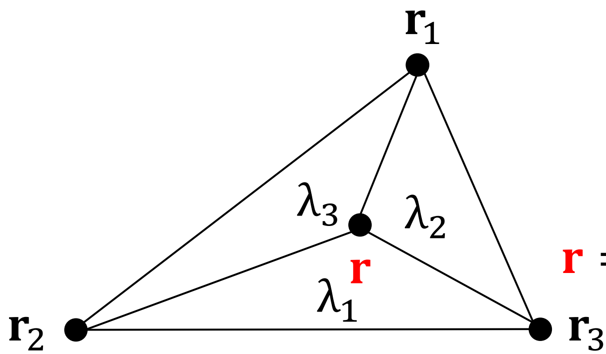
RPP (3/3)

- Begin with a known pose $(R_1, \mathbf{t}_1)$ got from any pose estimation algorithm (e.g., OI).
- Use this first guess of pose to estimate a second pose, which also has a minima of E_{os}.



EPnP (1/3)

- Homogeneous Barycentric Coordinates
  - Also known as area coordinates.



$$\mathbf{r} = \lambda_1 \mathbf{r}_1 + \lambda_2 \mathbf{r}_2 + \lambda_3 \mathbf{r}_3$$

- We can write the coordinates of the n 3D points $\mathbf{p}_i$ and $\mathbf{q}_i$ as a weighted sum of four virtual control points $\mathbf{c}_i^{\mathrm{p}}$ and $\mathbf{c}_i^{\mathrm{q}}$:

$$\mathbf{p}_i = \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_i^{\mathrm{p}}, \text{ with} \sum_{j=1}^{4} \alpha_{ij} = 1 \Rightarrow \mathbf{q}_i = \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_i^{\mathrm{q}}$$

$$\boxed{\begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_{i1} \\ \alpha_{i2} \\ \alpha_{i3} \\ \alpha_{i4} \end{bmatrix}}$$

$$\boxed{(R^*, \mathbf{t}^*) = \arg\min_{R,\mathbf{t}} \sum_{i=1}^{n} \left\| R\mathbf{c}_i^{\mathrm{p}} + \mathbf{t} - \mathbf{c}_i^{q} \right\|^2}$$

EPnP (2/3)

$$\begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = P \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = P \cdot T_{\mathrm{cm}} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \begin{bmatrix} s_x f & 0 & x_0 & 0 \\ 0 & s_y f & y_0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

- Let A be the intrinsic matrix.

$$\forall i, \qquad h_i \mathbf{v}_i = h_i \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = A\mathbf{q}_i = A \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_i^q = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \sum_{j=1}^{4} \alpha_{ij} \begin{bmatrix} x_j^c \\ y_j^c \\ z_j^c \end{bmatrix}$$

  - The unknown parameters of this linear system are the 12 control point coordinates and the n projective parameters.
  - From equation above, we know that $h_i = \sum_{j=1}^{4} \alpha_{ij} z_j^c$, and we can get:

$$\sum_{j=1}^{4} \left(\alpha_{ij} f_x x_j^c + \alpha_{ij} (x_0 - x_c) z_j^c \right) = 0 \text{ and } \sum_{j=1}^{4} \left(\alpha_{ij} f_y y_j^c + \alpha_{ij} (y_0 - y_c) z_j^c \right) = 0$$

EPnP (3/3)

- Finally, we generate a linear system of the form

$$M\mathbf{x} = 0$$

  - where $\mathbf{x} = \left[{\mathbf{c}_1^q}^t, {\mathbf{c}_2^q}^t, {\mathbf{c}_3^q}^t, {\mathbf{c}_4^q}^t\right]^t$ is a 12-vector made of the unknowns, and M is a $2n \times 12$ matrix.
  - The solution therefore belongs to the null space, or kernel, of M, and can found efficiently as the null eigenvectors of matrix $M^t M$, which is of small constant (12×12) size.
  - Computing the product $M^t M$ has $O(n)$ complexity, and is the most time consuming step in this method.

It's first closed-form solution to the problem with $O(n)$ complexity

OPnP (1/7)

- To facilitate global optimization via polynomial system solving, we advocate the non-unit quaternion parameterization, which is free of any trigonometric function.
- Letting $s = a^2 + b^2 + c^2 + d^2$, the rotation matrix can be expressed as

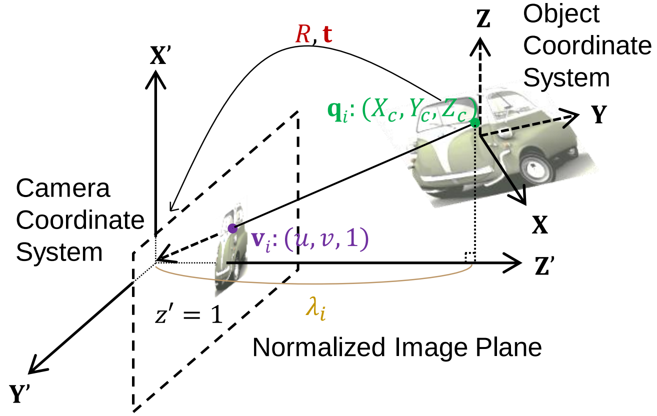
$$R = \frac{1}{s}\begin{bmatrix} a^2+b^2+c^2+d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2-d^2 \end{bmatrix}$$

$$\boxed{R(a,b,c,d) = R(ka,kb,kc,kd)}$$

OPnP (2/7)

- We've known that

$$\mathbf{q}_i = \lambda_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = R\mathbf{p}_i + \mathbf{t}, \mathrm{i} = 1,2,\ldots,\mathrm{n}$$



- Since the absolute scale of (a, b, c, d) is arbitrary, we can fix it by using the reciprocal of the average depth

$$s \equiv \frac{1}{\frac{1}{n}\sum_{i=1}^{n} \lambda_\mathrm{i}} = \frac{1}{\bar{\lambda}}$$

OPnP (3/7)

- So

$$\hat{\lambda}_i = \frac{\lambda_i}{\bar{\lambda}} \qquad \begin{bmatrix} \hat{t}_1 \\ \hat{t}_2 \\ \hat{t}_3 \end{bmatrix} = \frac{\mathbf{t}}{\bar{\lambda}}$$

$$(s\lambda_i)\mathbf{v}_i = (sR)\mathbf{p}_i + (s\mathbf{t}) \Longrightarrow \hat{\lambda}_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \\ \mathbf{r}_3^t \end{bmatrix} \mathbf{p}_i + \begin{bmatrix} \hat{t}_1 \\ \hat{t}_2 \\ \hat{t}_3 \end{bmatrix}, i = 1,2,\ldots,n$$

$$\begin{bmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \\ \mathbf{r}_3^t \end{bmatrix} = \begin{bmatrix} a^2+b^2+c^2+d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2-d^2 \end{bmatrix}$$

- It is straightforward to recognize that

$$\sum_{i=1}^{n} \hat{\lambda}_i = \frac{\sum_{i=1}^{n} \lambda_\mathrm{i}}{\bar{\lambda}} = \frac{\sum_{i=1}^{n} \lambda_\mathrm{i}}{\frac{1}{n}\sum_{i=1}^{n} \lambda_\mathrm{i}} = n\frac{\sum_{i=1}^{n} \lambda_\mathrm{i}}{\sum_{i=1}^{n} \lambda_\mathrm{i}} = n \quad \text{and} \quad \hat{\lambda}_i = \mathbf{r}_3^t\mathbf{p}_i + \hat{t}_3, i = 1,2,\ldots,n$$

OPnP (4/7)

- Because $\sum_{i=1}^{n} \hat{\lambda}_i = n$ and $\hat{\lambda}_i = \mathbf{r}_3^t \mathbf{p}_i + \hat{t}_3$

$$\Rightarrow \quad \hat{t}_3 = 1 - \mathbf{r}_3^t \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{p}_i \right) = 1 - \mathbf{r}_3^t \overline{\mathbf{p}}$$

$$\Rightarrow \quad \hat{\lambda}_i = \mathbf{r}_3^t \mathbf{p}_i + (1 - \mathbf{r}_3^t \overline{\mathbf{p}}) = 1 + \mathbf{r}_3^t (\mathbf{p}_i - \overline{\mathbf{p}}) = 1 + \mathbf{r}_3^t \mathbf{p}_i'$$

$$\hat{\lambda}_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \\ \mathbf{r}_3^t \end{bmatrix} \mathbf{p}_i + \begin{bmatrix} \hat{t}_1 \\ \hat{t}_2 \\ \hat{t}_3 \end{bmatrix} \quad \Rightarrow \quad (1 + \mathbf{r}_3^t \mathbf{p}_i') \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \\ \mathbf{r}_3^t \end{bmatrix} \mathbf{p}_i + \begin{bmatrix} \hat{t}_1 \\ \hat{t}_2 \\ 1 - \mathbf{r}_3^t \overline{\mathbf{p}} \end{bmatrix}$$

$$\Rightarrow \quad (1 + \mathbf{r}_3^t \mathbf{p}_i') \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \end{bmatrix} \mathbf{p}_i + \begin{bmatrix} \hat{t}_1 \\ \hat{t}_2 \end{bmatrix}$$

OPnP (5/7)

- Due to noise, the equation below could not be completely satisfied in general.

$$(1+\mathbf{r}_3^t\mathbf{p}_i')\begin{bmatrix}u_i\\ v_i\end{bmatrix}=\begin{bmatrix}\mathbf{r}_1^t\\ \mathbf{r}_2^t\end{bmatrix}\mathbf{p}_i+\begin{bmatrix}\hat{t}_1\\ \hat{t}_2\end{bmatrix}, i=1,2,\dots,n$$

- Therefore, we directly minimize this cost function:

$$\min_{a,b,c,d,\hat{t}_1,\hat{t}_2}\sum_{i=1}^{n}[(1+\mathbf{r}_3^t\mathbf{p}_i')u_i-\mathbf{r}_1^t\mathbf{p}_i-\hat{t}_1]^2+\sum_{i=1}^{n}[(1+\mathbf{r}_3^t\mathbf{p}_i')v_i-\mathbf{r}_2^t\mathbf{p}_i-\hat{t}_2]^2$$

  – Before really solving this problem, we can easily project out $\hat{t}_1$ and $\hat{t}_2$ in closed-form as follows

$$\hat{t}_1=\bar{u}+\mathbf{r}_3^t\left(\frac{1}{n}\sum_{i=1}^{n}u_i\,\mathbf{p}_i'\right)-\mathbf{r}_1^t\bar{\mathbf{p}}\quad\text{and}\quad\hat{t}_2=\bar{v}+\mathbf{r}_3^t\left(\frac{1}{n}\sum_{i=1}^{n}v_i\,\mathbf{p}_i'\right)-\mathbf{r}_2^t\bar{\mathbf{p}}$$

$$\boxed{\bar{u}\stackrel{\text{def}}{=}\frac{1}{n}\sum_{i=1}^{n}u_i}\qquad\boxed{\bar{v}\stackrel{\text{def}}{=}\frac{1}{n}\sum_{i=1}^{n}v_i}$$

OPnP (6/7)

- Now letting $\boldsymbol{\alpha} = [1, a^2, ab, ac, ad, b^2, bc, bd, c^2, cd, d^2]^t$, we rewrite the cost function into the matrix form

$$\min_{a,b,c,d} f(a,b,c,d) = \|M\boldsymbol{\alpha}\|_2^2 = \boldsymbol{\alpha}^t M^t M \boldsymbol{\alpha}$$

  - M is a $2n \times 11$ matrix

- By calculating the derivative of cost function with respect to a, b, c, d, the first-order optimality condition reads

$$\frac{\partial f}{\partial a} = 0, \frac{\partial f}{\partial b} = 0, \frac{\partial f}{\partial c} = 0, \frac{\partial f}{\partial d} = 0$$

  - which is composed of four three-degree polynomials with respect to a, b, c, d.

Gröbner Basis Solver

OPnP (7/7)

- Although solving multivariate polynomial systems is challenging in general, the multiview geometry community has achieved much progress by means of the Gröbner basis (GB) technique.
- Kukelova et al. [2] even developed an automatic generator of GB solvers, which facilitates the solving of polynomial systems arising from geometric computer vision problems.

[2] "*Automatic generator of minimal problem solvers*," ECCV 2008.

Outline

- Introduction
- System Overview
- Feature Detection & Matching
- Outlier Removal
- Pose Estimation
- Experimental Results
- Conclusion

Experimental Setup

- Our test images are a combination of template images and background images.

Detectors and Descriptors (1/3)

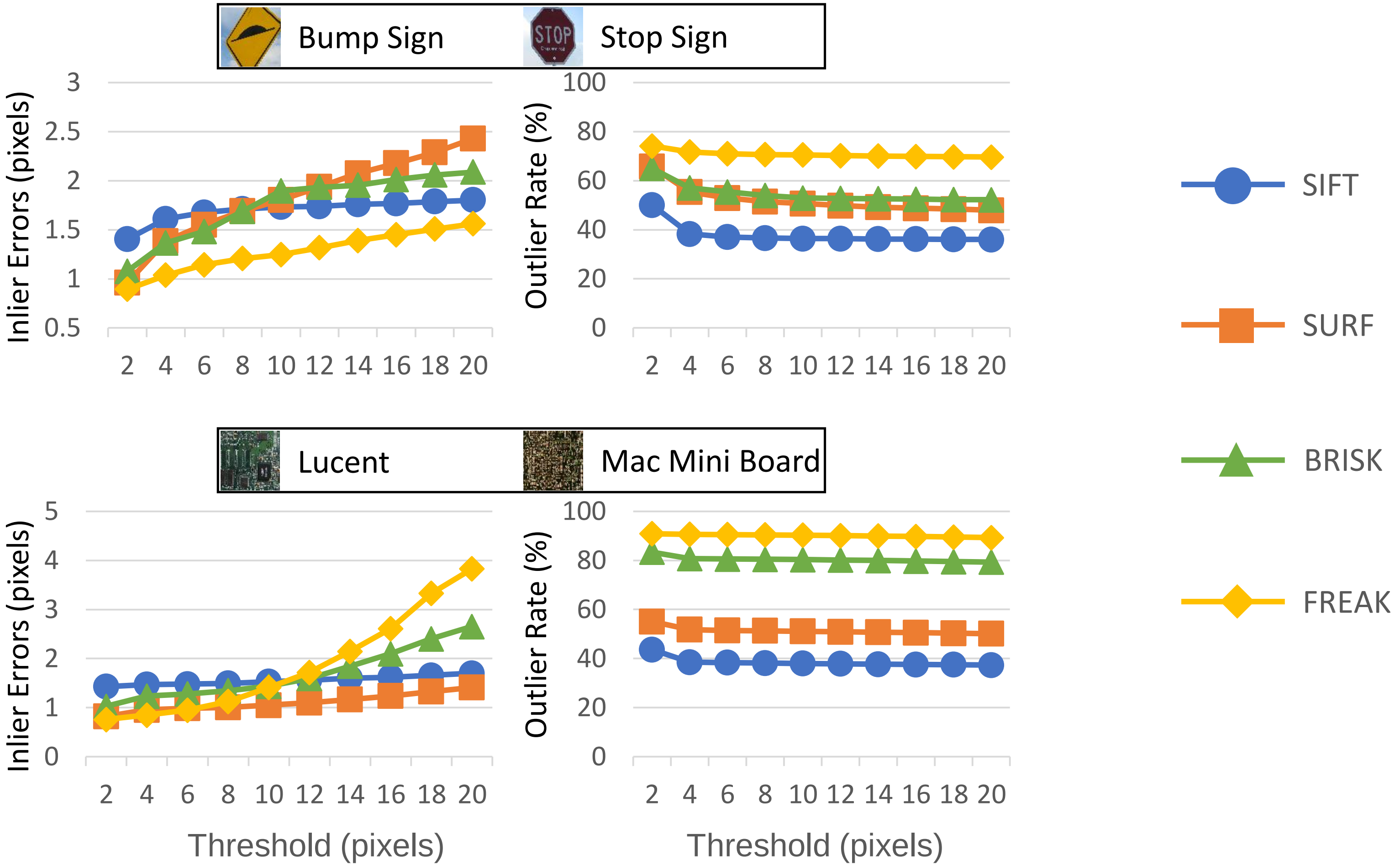
- One match will be classified into outliers if the distance from the location of feature detected on the test image to its nominal location is above a specific threshold.

$$outlier\ rate = \frac{\#outlier}{\#attempted\ matches}$$

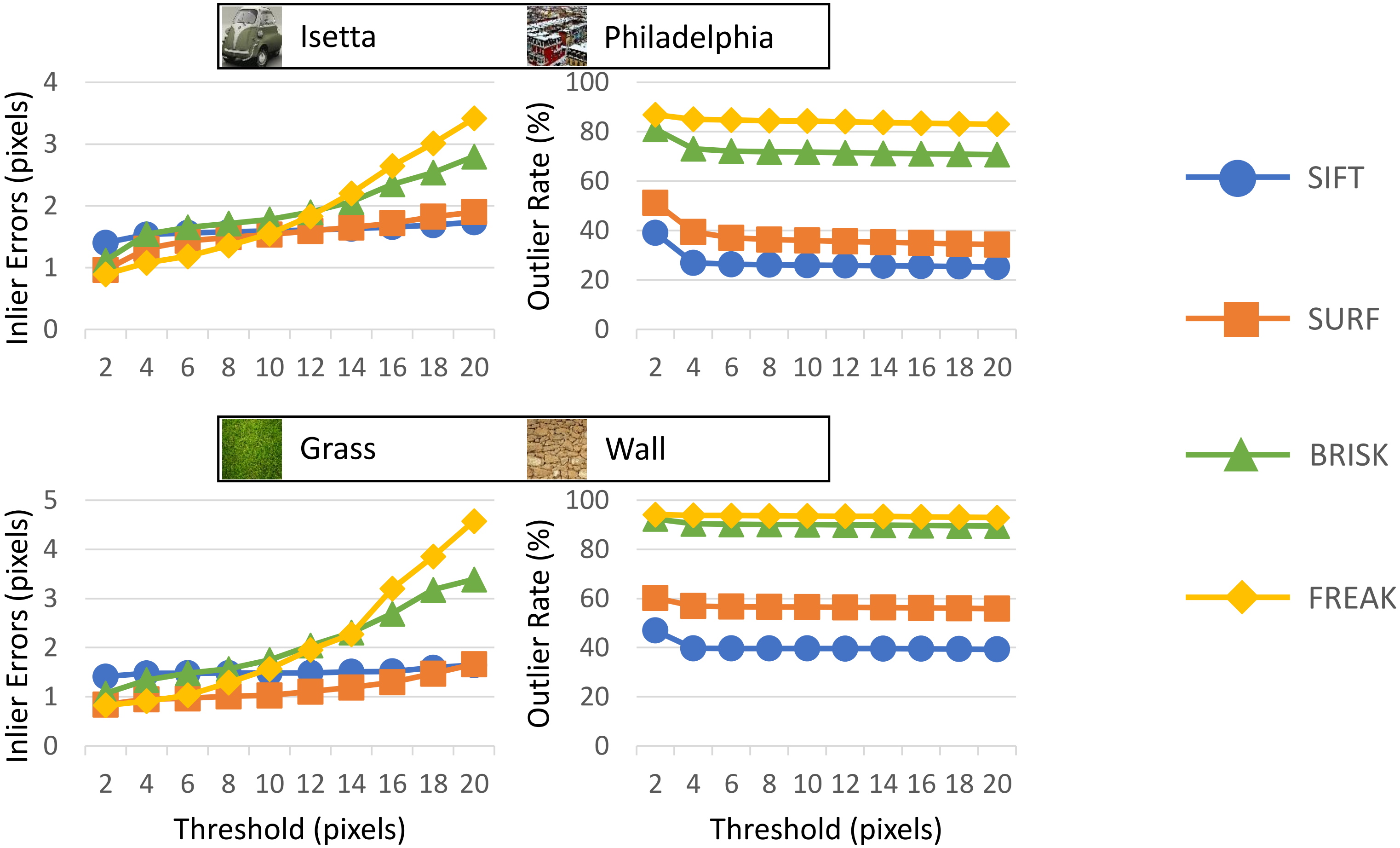
- The performance of PnP algorithms depends on the localization accuracy of the remaining feature correspondences after removing outliers, so we identify the inlier error as an accuracy metric:

$$inlier\ error = \frac{\sum inlier\ distances\ on\ test\ image}{\#inliers}$$

Detectors and Descriptors (2/3)



Detectors and Descriptors (3/3)

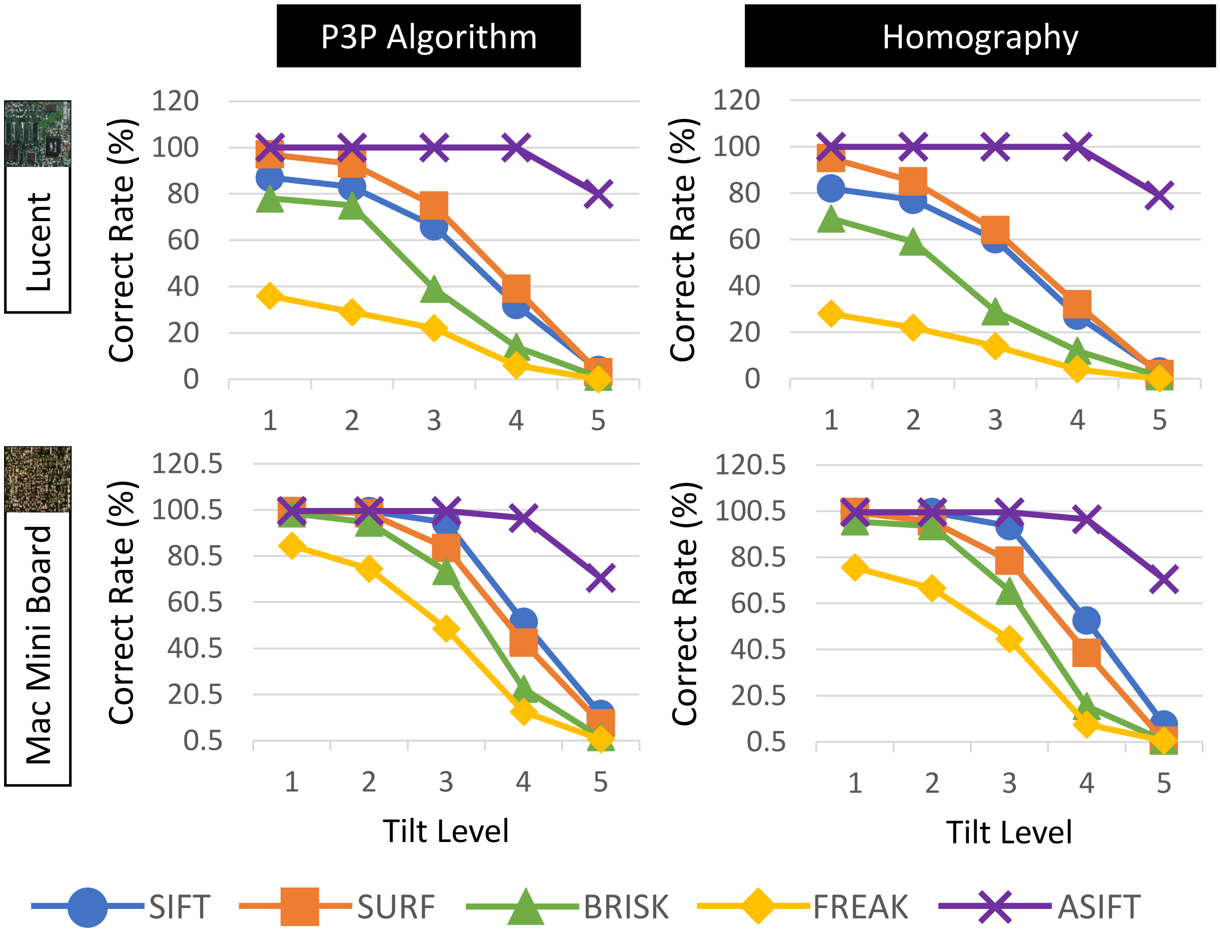


Outlier Removal (1/2)

- Here we identify the correct rate of the RANSAC-based schemes as

$$\text{correct rate} = \frac{\#\text{tests with inlier error} < \text{threshold}}{\#\text{all tests}}$$

Outlier Removal (2/2)

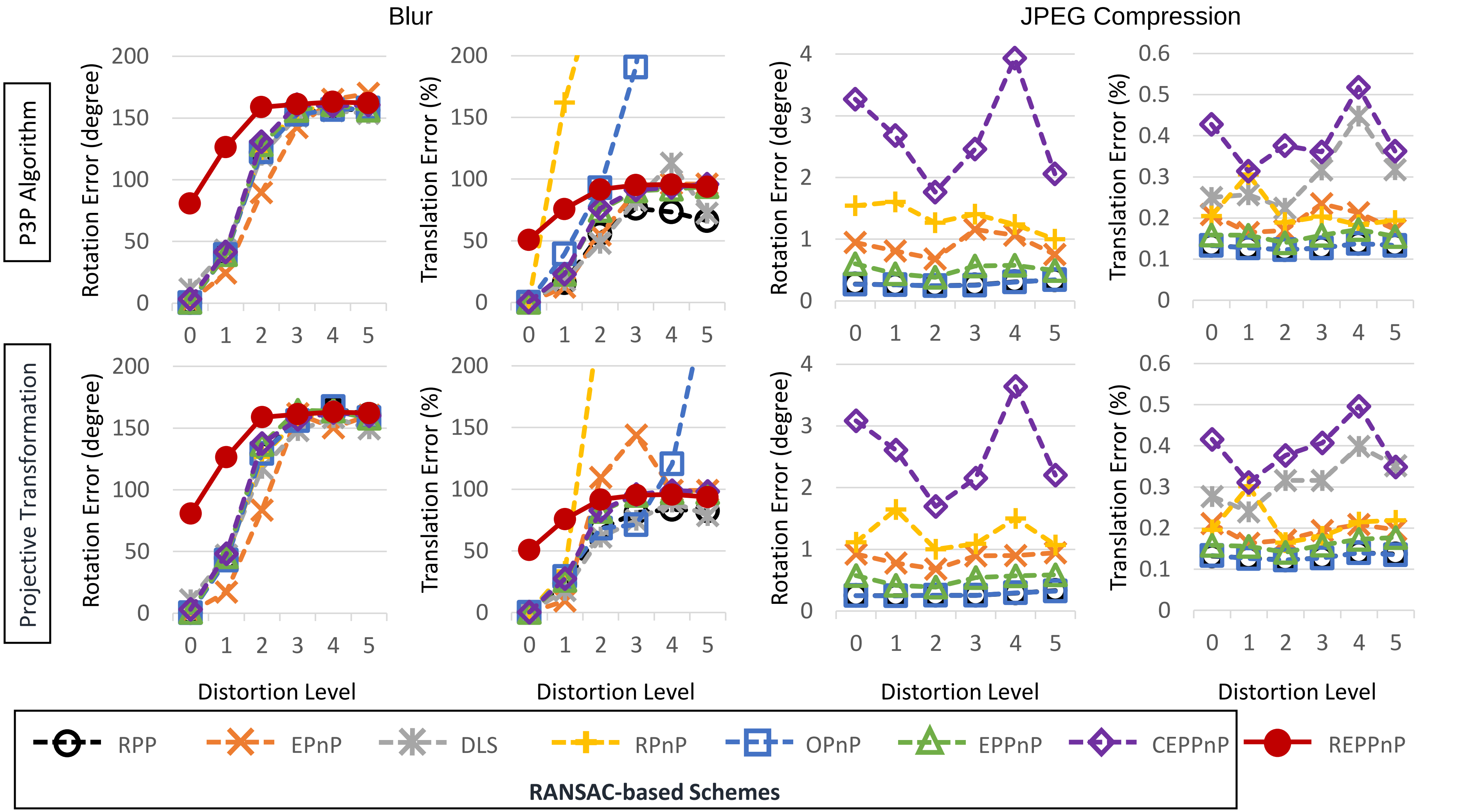


Mean Cost (ms)	ASIFT	Others
P3P	1.52	184.83
Homo.	0.59	54.37

PnP Algorithm (1/3)

- Given the true rotation matrix R_{true} and translation vector $\mathbf{t}_{true}$
  - $E_{rot}(degree) = \sqrt{\sum_{i=1}^{3}\left[acosd\left(\mathbf{r}_{true}^{i} \cdot \mathbf{r}^{i}\right)\right]^2}$
    - $\mathbf{r}_{true}^{i}$ and $\mathbf{r}^{i}$ are the k-th column of R_{true} and R
    - $acosd(\cdot)$represent the arc-cosine operation in degrees
  - $E_{trans}(\%) = \frac{\|\mathbf{t}_{true} - \mathbf{t}\|}{\|\mathbf{t}_{true}\|} \times 100$

PnP Algorithm (2/3)

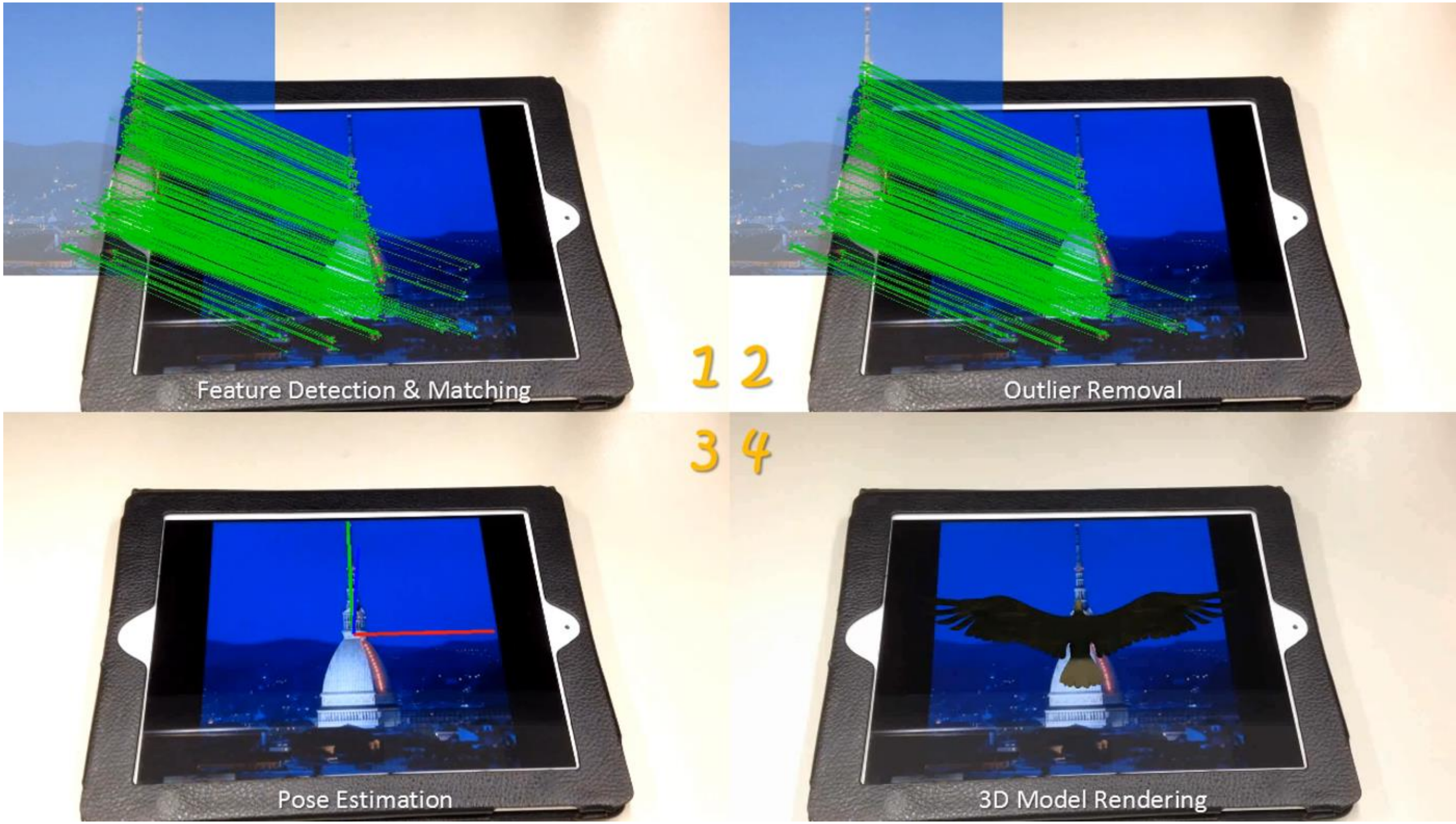


PnP Algorithm (3/3)

- Computation time of RANSAC-based schemes and state-of-the-art PnP algorithms.

Mean Cost (ms)	RANSAC	RPP	EPnP	DLS	RPnP	OPnP	EPPnP	CEPPnP	REPPnP
RANSAC P3P	141.59	869.45	23.55	44.45	2.36	17.64	2.00	3.45	3.00
RANSAC Proj.	42.52	873.55	31.73	37.55	2.18	19.18	2.00	3.36	

Demo Video
(ASIFT + RANSAC-P3P + OPnP)



Outline

- Introduction
- System Overview
- Feature Detection & Matching
- Outlier Removal
- Pose Estimation
- Experimental Results
- Conclusion

Conclusion

- The patch-based descriptors perform consistently better then the binary descriptors on the correct match rate.
- RANSAC-P3P is more reliable than -Homography.
- OPnP is the most prominent with considering both the accuracy of the estimated pose and the computation time.
- The optimal solution to address the feature-based pose estimation problem at the current stage is the combination of ASIFT, RANSAC-P3P, and OPnP.