Robust Feature Matching and Fast GMS Solution

Singapore University of Technology and Design (SUTD)

Advanced Digital Sciences Center (ADSC)

Speaker：JiaWang Bian(边佳旺)

http://jwbian.net/

June 14,2017

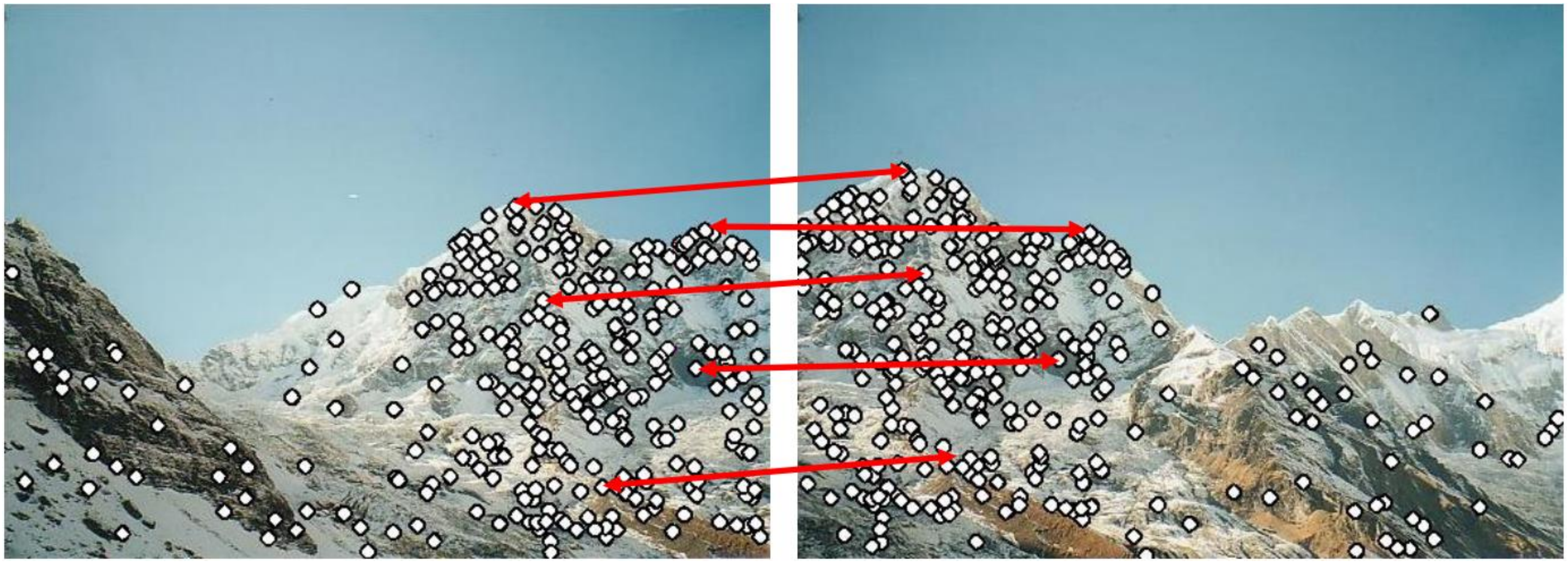
Content

- Feature Matching Introduction
  - Feature Matching
  - Feature Detector & Descriptor
  - Matching
  - RANSAC-based Geometry Estimation (or Verification)
- Recent Robust Matchers
  - CODE (PAMI,2016)
  - RepMatch (ECCV,2016)
- Fast and Robust GMS Solution(CVPR,2017)
  - Video Demo
  - Methodology
  - Algorithm
  - Share (Material Links)

Feature Matching Introduction

Feature Matching Introduction

- Feature Matching



- Pipeline

Detection → Description → Matching → Geometry

Feature Matching Introduction

- Applications

Correct Correspondences

Geometry between 2 views

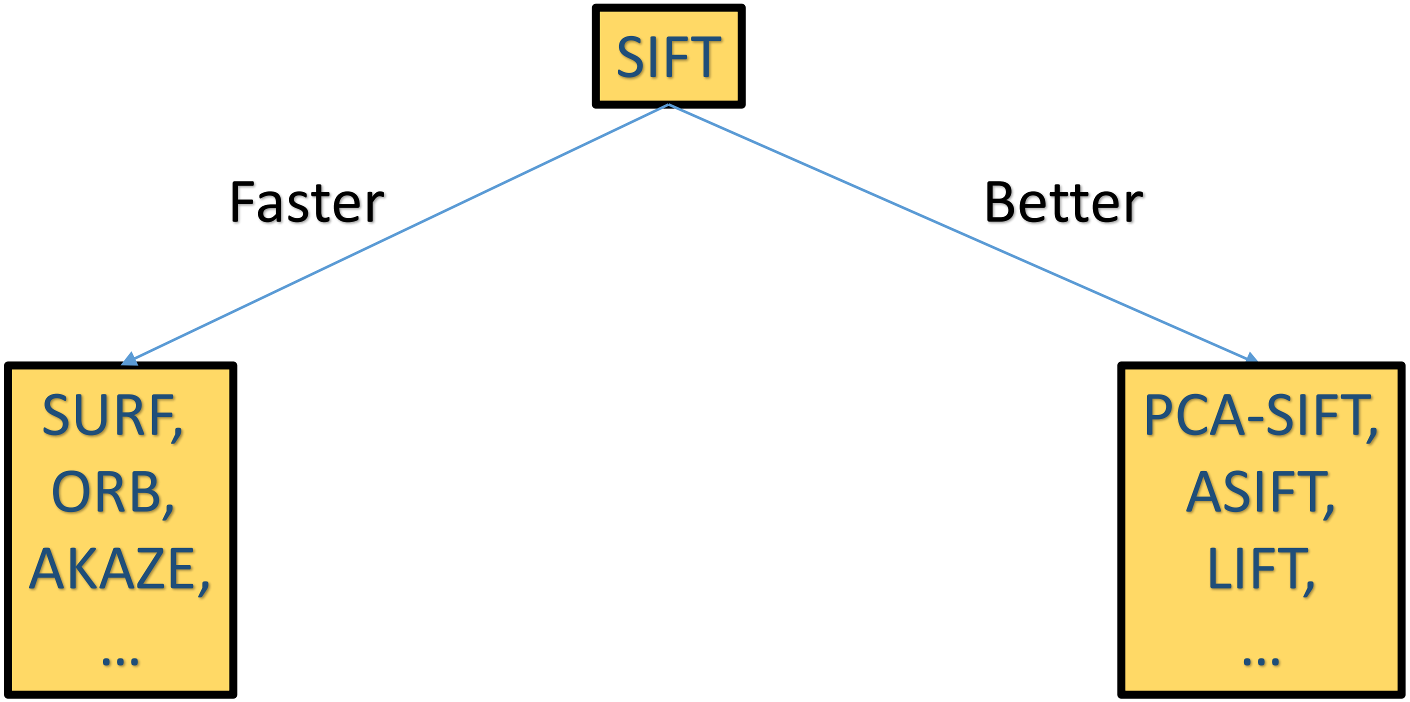
Estimate Camera Pose
Localization (SFM)
Tracking (SLAM)
…

Similarity(Number of matches)

Image retrieval
Object Recognition
Loop Closing (SLAM)
Re-localization (SLAM)
…

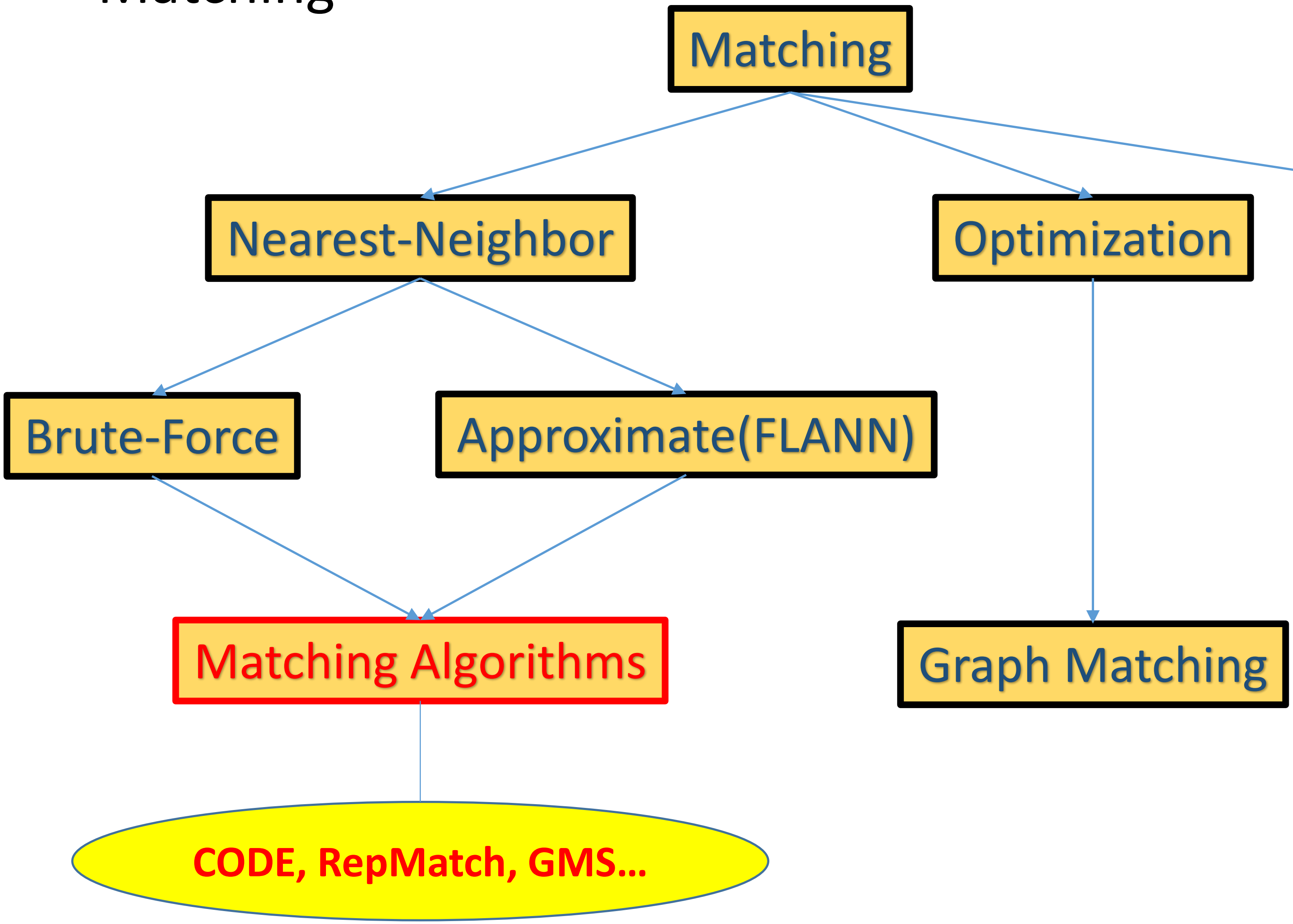
Sparse Feature Matching

- Feature detector & descriptor



Feature Matching Introduction

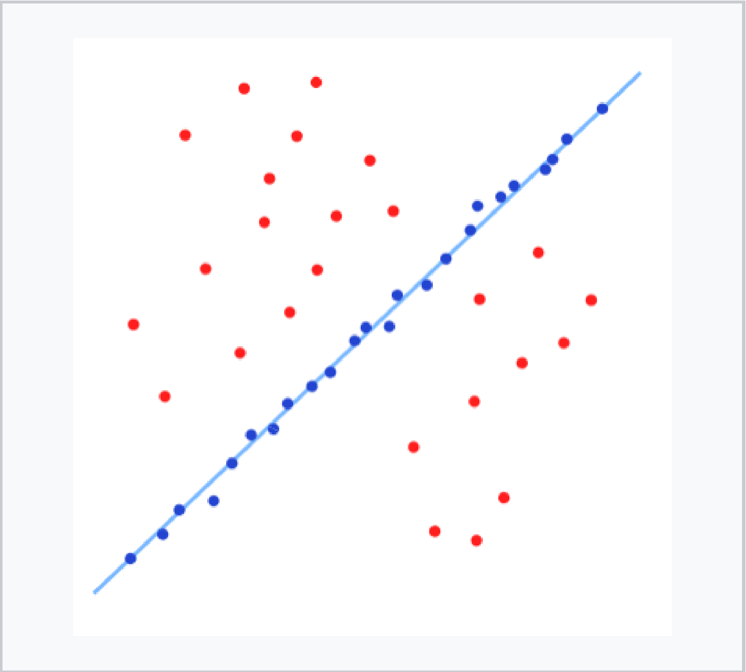
- Matching



Feature Matching Introduction

- RANSAC-based Geometry Estimation (or Verification)
  - An example for RANSAC framework (fitting a line)

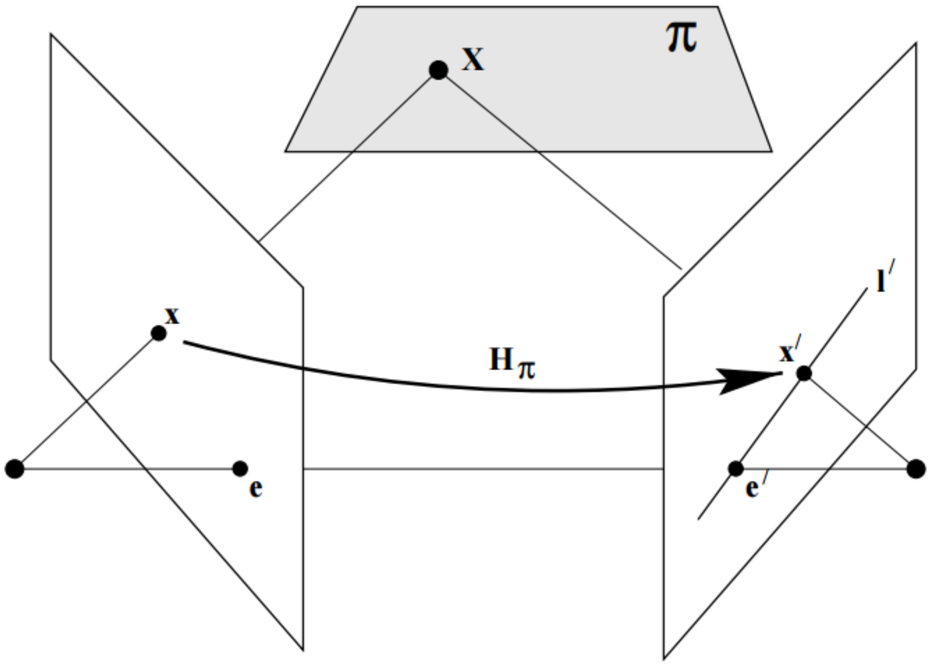
A data set with many outliers for which a line has to be fitted.



Fitted line with RANSAC; outliers have no influence on the result.

Feature Matching Introduction

- RANSAC-based Geometry Estimation (or Verification)
  - Fundamental Matrix (for 3D scenes)
    - Point to Line (weak, general)
  - Homography (for 2D scenes)
    - Point to Point (strong, narrow range)



Recent Robust Matchers

Recent Robust Matchers

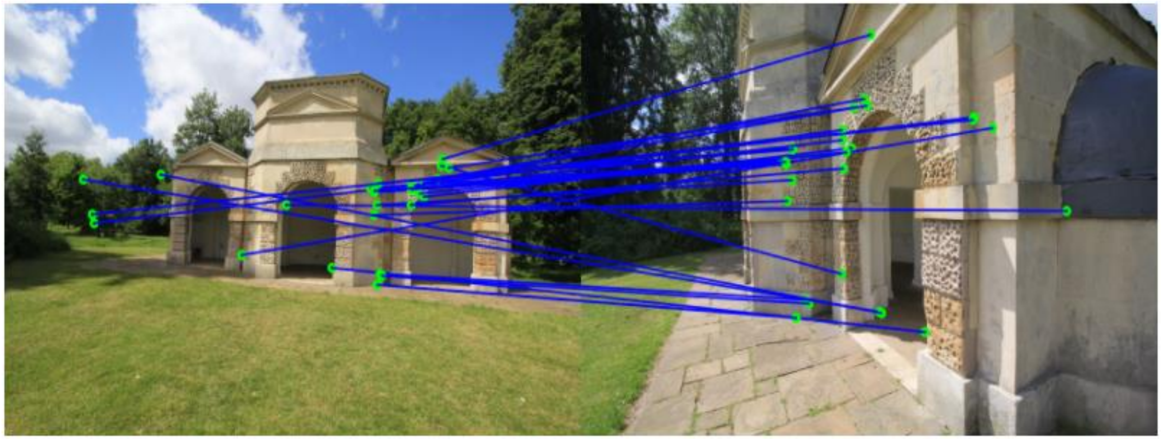
- CODE[1]
  - For wide-baseline matching.

- RepMatch[2]
  - Based on CODE[1].
  - Solve the repeated structure problem.

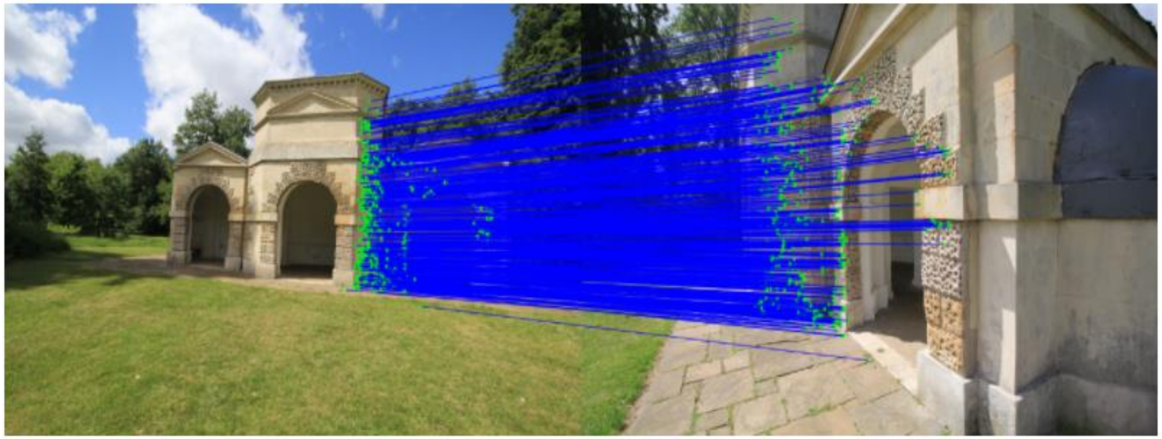
[1] CODE: Coherence Based Decision Boundaries for Feature Correspondence, IEEE TPAMI,2016, Lin et. al.
[2] RepMatch: Robust Feature Matching and Pose for Reconstructing Modern Cities, ECCV, 2016,, Lin et. al.

Recent Robust Matchers (CODE)

- Wide-baseline matching



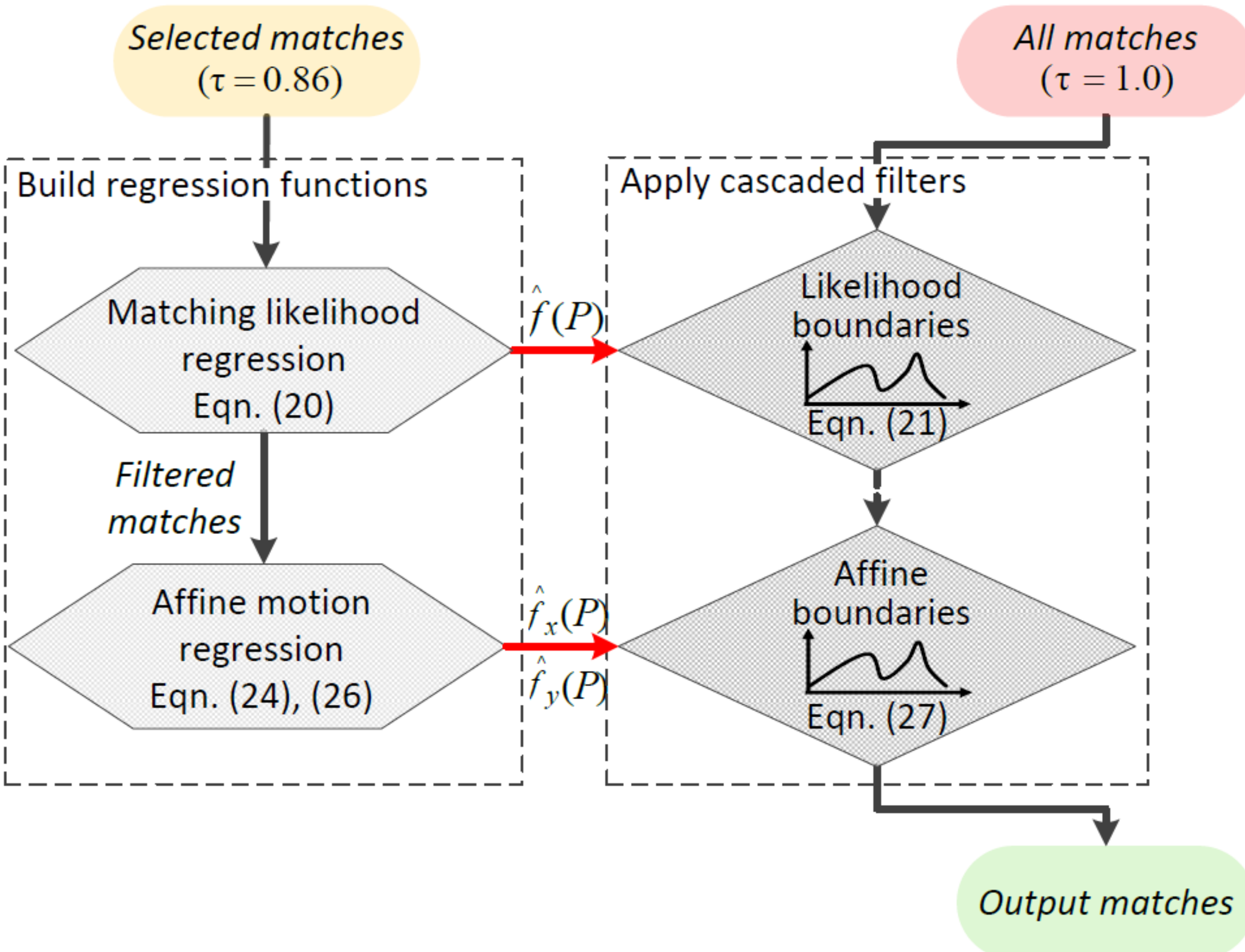
Traditional A-SIFT feature matching



CODE feature correspondence with the same A-SIFT features

Recent Robust Matchers (CODE)

- Idea



Recent Robust Matchers (CODE)

- Regression models

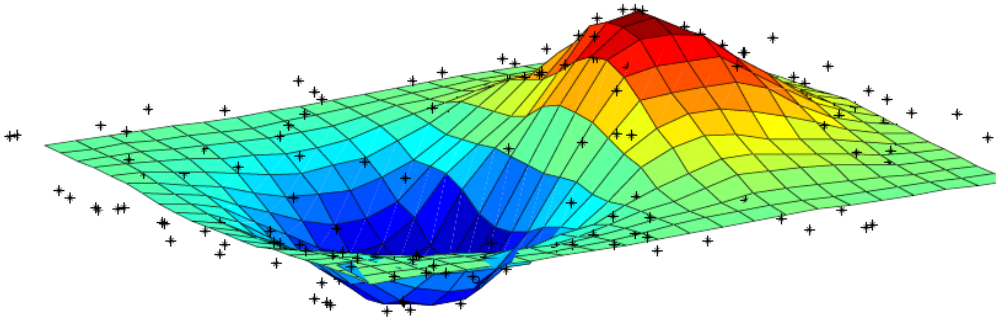


Fig. 2: Regression can be understood as finding a continuous surface that explains scattered data points (denoted by "+").

  - Likelihood Regression
  - Affine motion regression -> x
  - Affine motion regression -> y

Recent Robust Matchers (CODE)

- Likelihood Regression
  - Train Data
    - Selected distinctive correspondences(after ratio-test).
  - Test Data
    - All feature correspondences.
  - Features of a correspondence
    - $X_i = [x, y, dx, dy, T_1, T_2, T_3, T_4]$.
    - T is a transformation matrix of [s1, r1] to [s2, r2].
    - s means scale, r represents rotation.
  - Labels
    - 1 for all correspondences
  - Cost function
    - Huber function
  - Non-linear Optimization
    - Construct Gaussian Similar Matrix
    - X(Matrix with n x n elements), Y(Matrix with nx1 elements(1))
    - n is the number of train data

Recent Robust Matchers (CODE)

- Affine motion regression
  - Train Data
    - The inliers of train data in the likelihood model
  - Test Data
    - Correspondences filtered by the likelihood model
  - Feature Space
    - Same as the likelihood model
  - Label
    - X2, and y2.(x,y represents pixel position, 2 means the second image)
  - Cost function
    - Huber function
  - Non-linear Optimization
    - Same as before(Gaussian Similar Matrix).

Recent Robust Matchers (CODE)

- Insight (likelihood model)

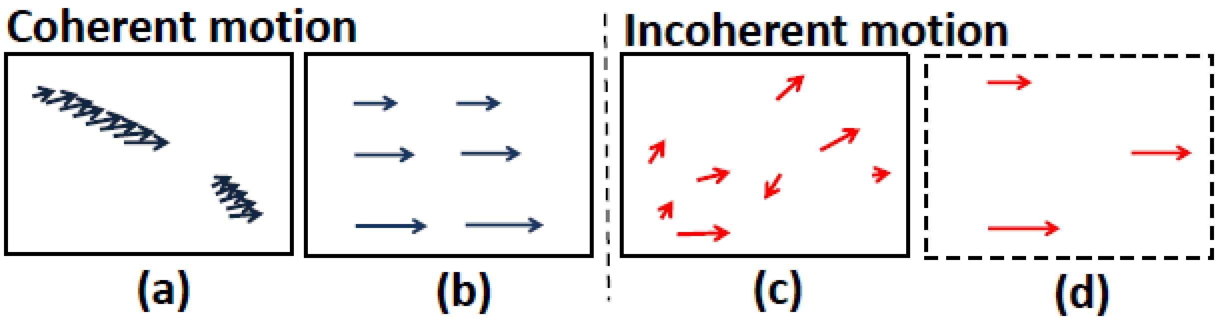
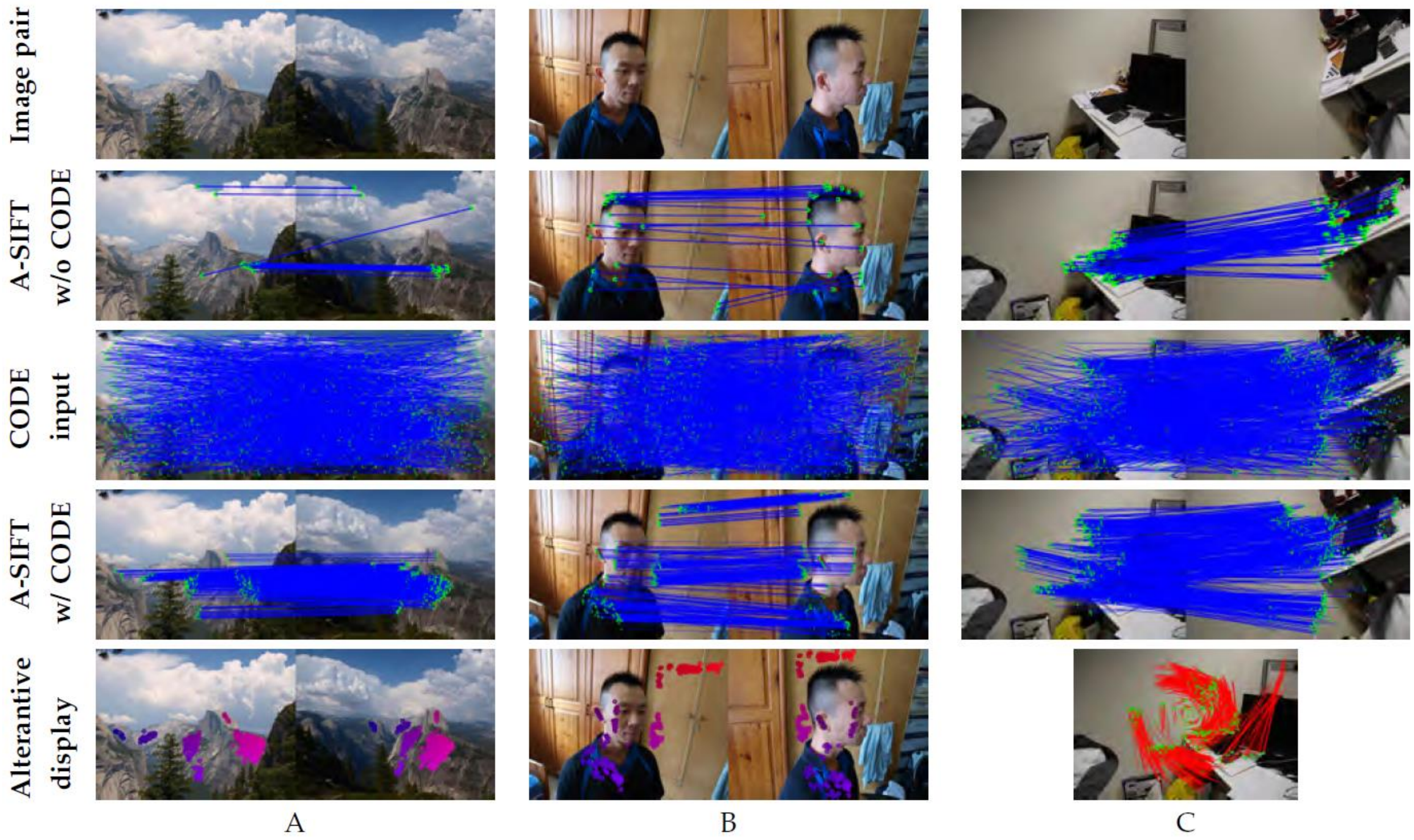


Fig. 3: Coherence based separation of true and false matches. Motions are considered coherent if (a) many local points make similar motions or (b) there is broad spatial support for the motion. This is enforced via the likelihood function in Eqn. (21). In contrast, feature matches in (c) and (d) do not give coherent motions, as the matches are not consistent in (c), while there are insufficient smoothly moving points to justify a long-range motion coherence model in (d).

Recent Robust Matchers (CODE)

- Matching samples



Recent Robust Matchers (CODE)

- Structure from Motion

A set of multi-view images [43]

Agisoft [48]: A commercial 3D reconstruction software

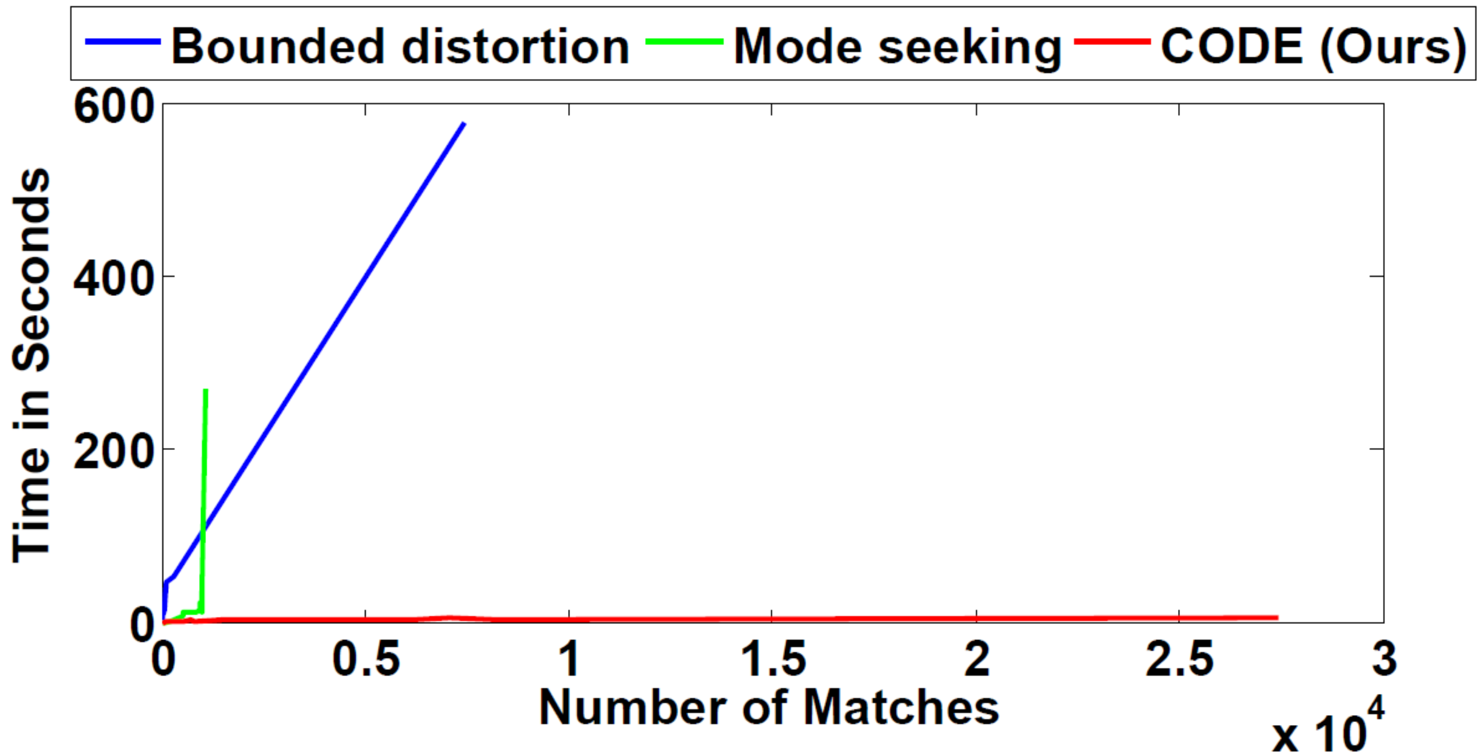
Visual SfM [3], [44], [45], [46], [47]

Visual SfM using feature matches returned by *A-SIFT w CODE*

C. Wu, “VisualSfM: A visual structure from motion system,” 2011[Online]. Available: http://ccwu.me/vsfm/

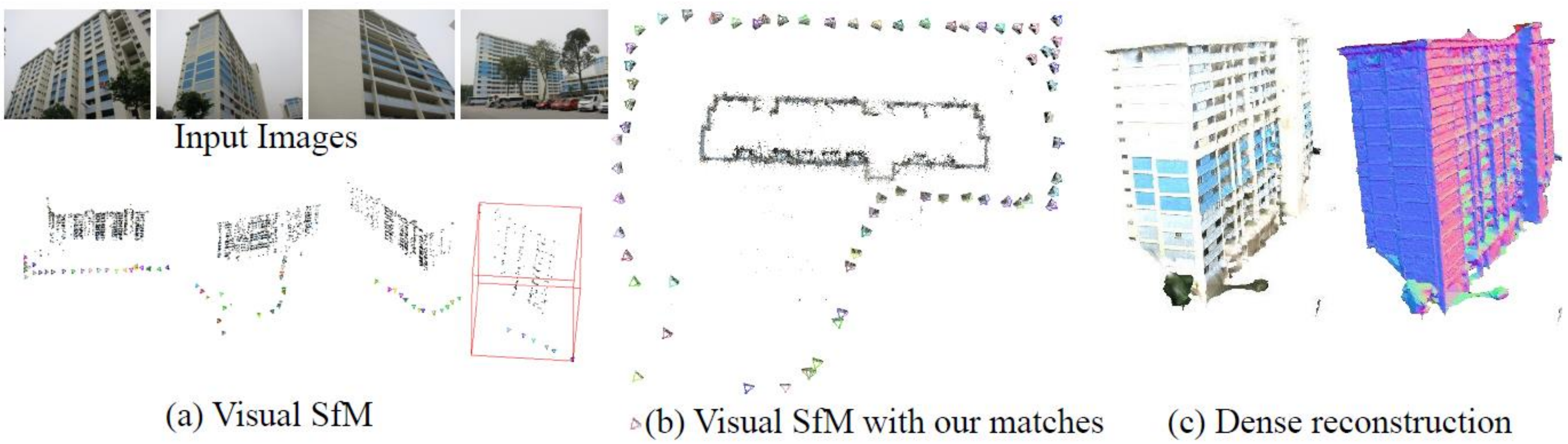
Recent Robust Matchers (CODE)

- Run time comparison



Recent Robust Matchers (RepMatch)

- RepMatch



Recent Robust Matchers (RepMatch)

- Repetitive Structure

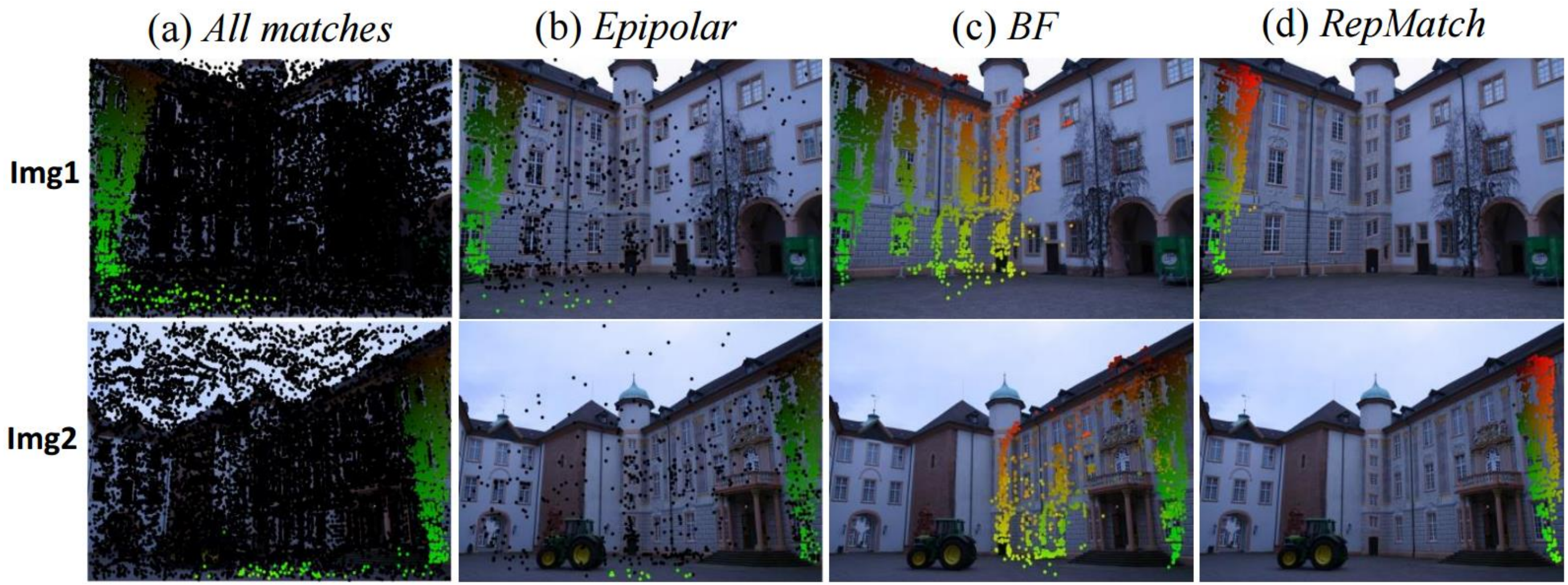
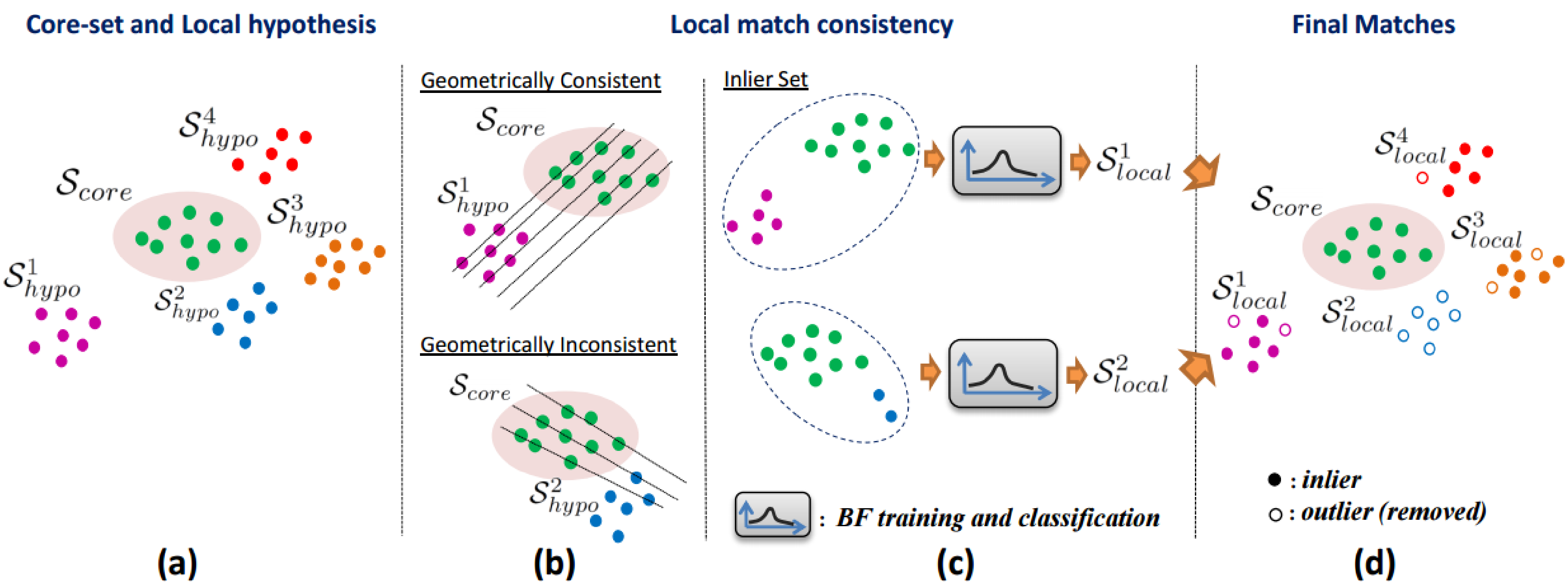


Illustration on real images. Black dots in (a) & (b) indicate wrong matches.
Note: Common central tower belong to physically different parts of the building.

Recent Robust Matchers (RepMatch)

- Idea



Recent Robust Matchers (RepMatch)

- Structure from Motion

(a) Visual SfM

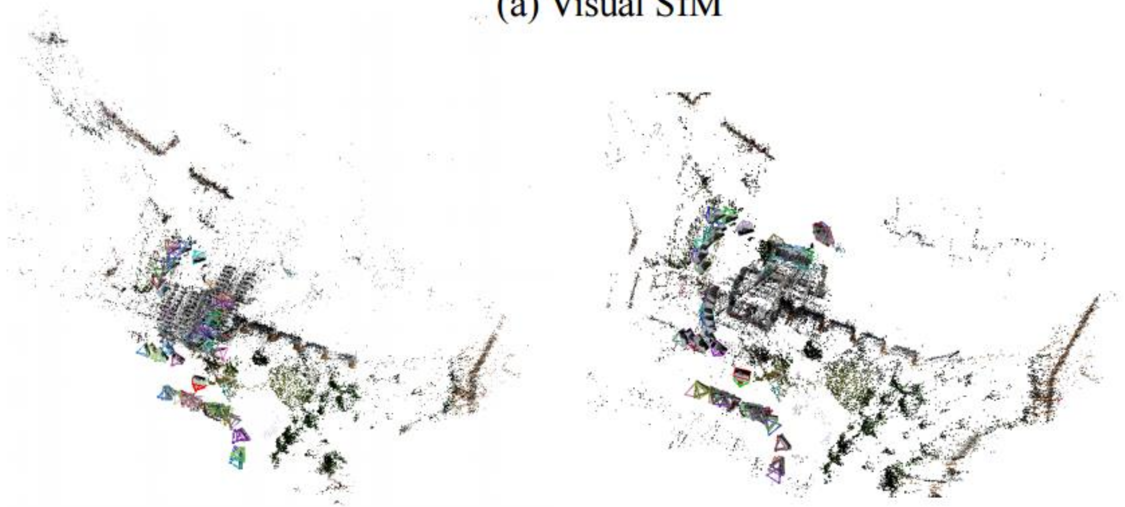
(b) Visual SfM with *BF* matches (c) Visual SfM with *RepMatch*

(i) Indoor scene

Recent Robust Matchers (RepMatch)

- Structure from Motion

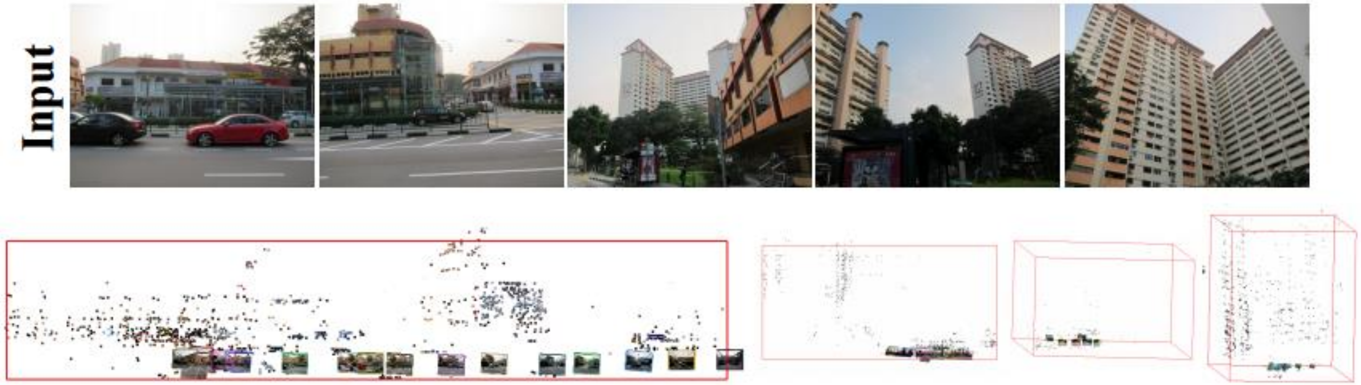
(a) Visual SfM



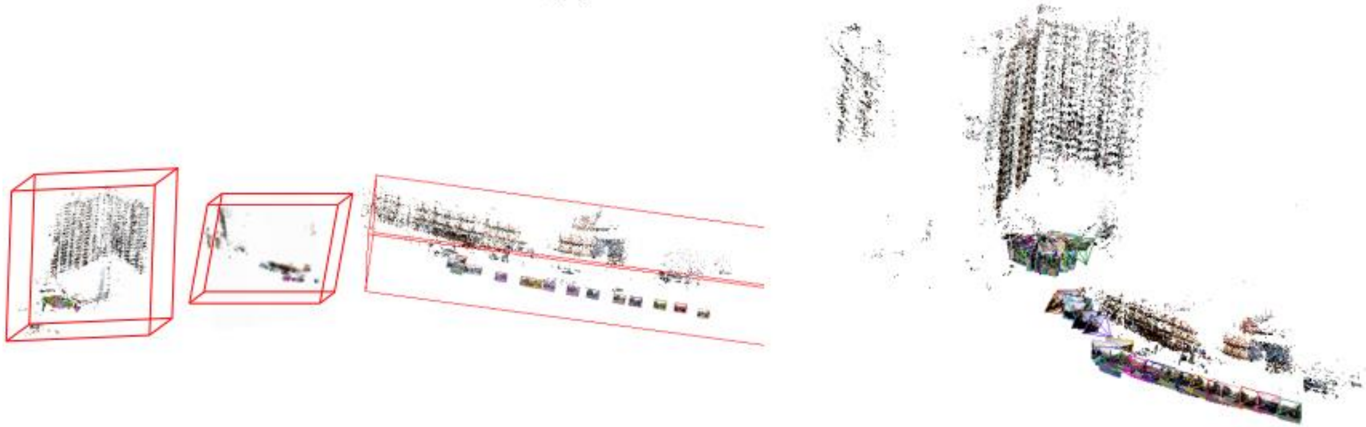
(b) Visual SfM with *BF* matches (c) Visual SfM with *RepMatch*

Recent Robust Matchers (RepMatch)

- Structure from Motion



(a) Visual SfM



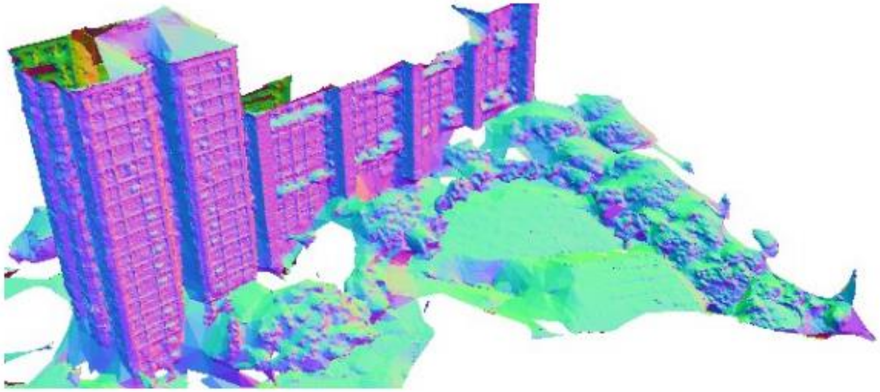
(b) Visual SfM with *BF* matches (c) Visual SfM with *RepMatch*

(ii) City street scene

Recent Robust Matchers (RepMatch)

- Structure from Motion

(d) *RepMatch* based reconstruction



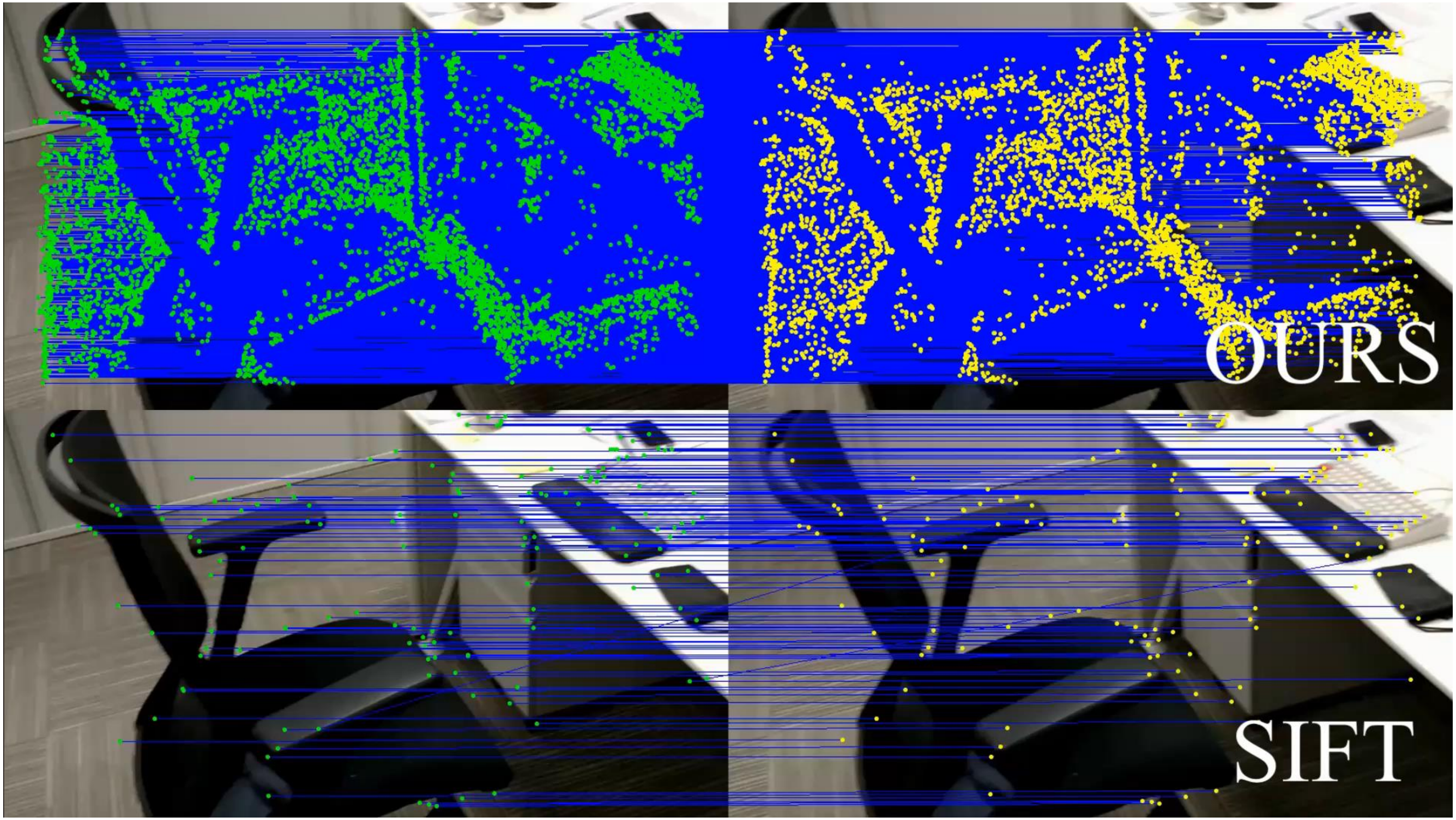
(e) *RepMatch* based normal map

(iii) Building scene

Fast and Robust GMS Solution

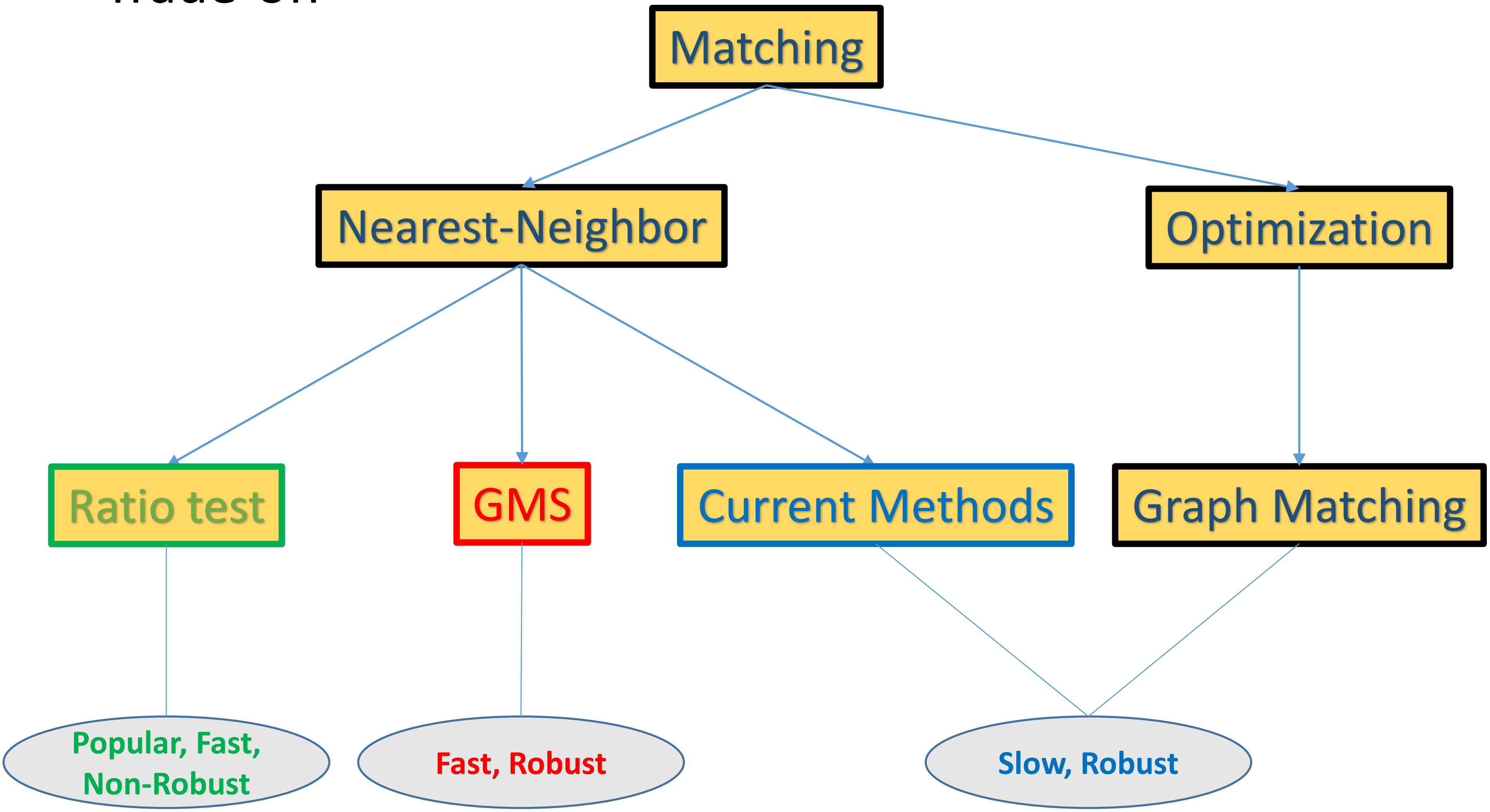
Video Demo

- ORB with GMS vs SIFT with Ratio

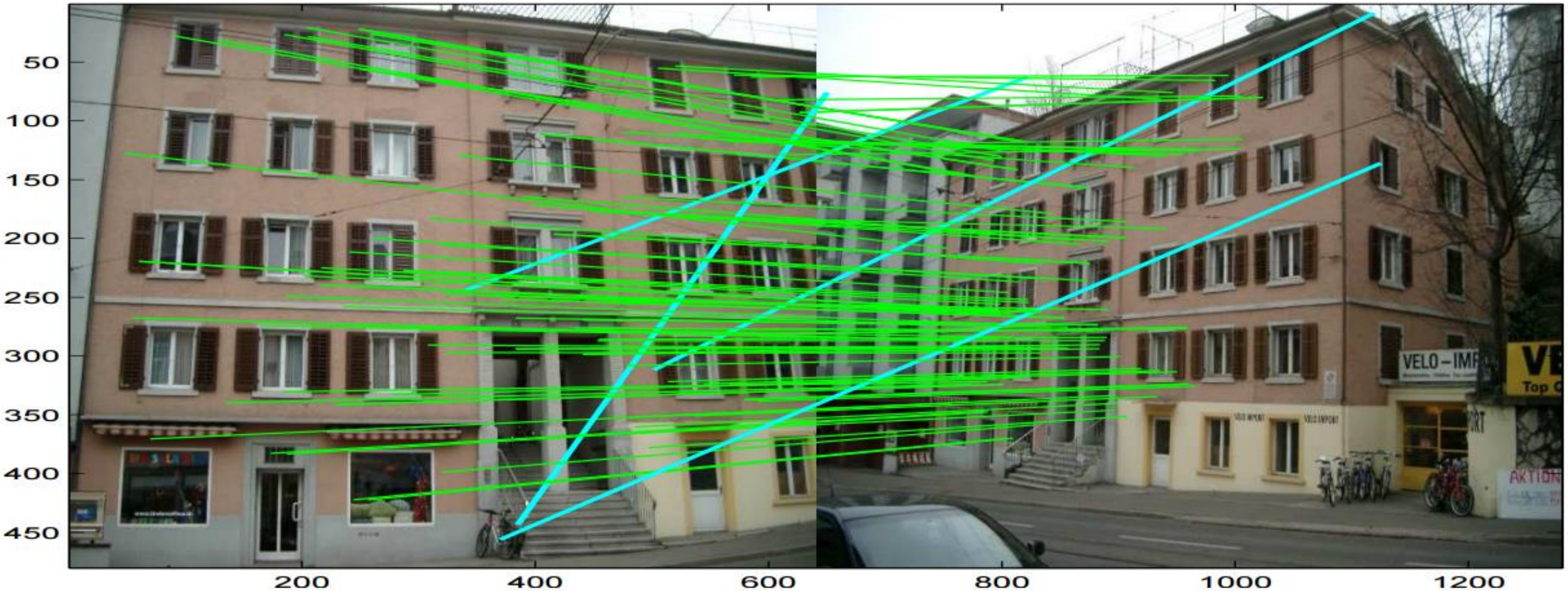


Motivation: Trade-off of quality and speed

- Trade-off



Methodology: Motion Smoothness



- Observation
  - True matches(green) are visually smooth while false matches(cyan) are not.

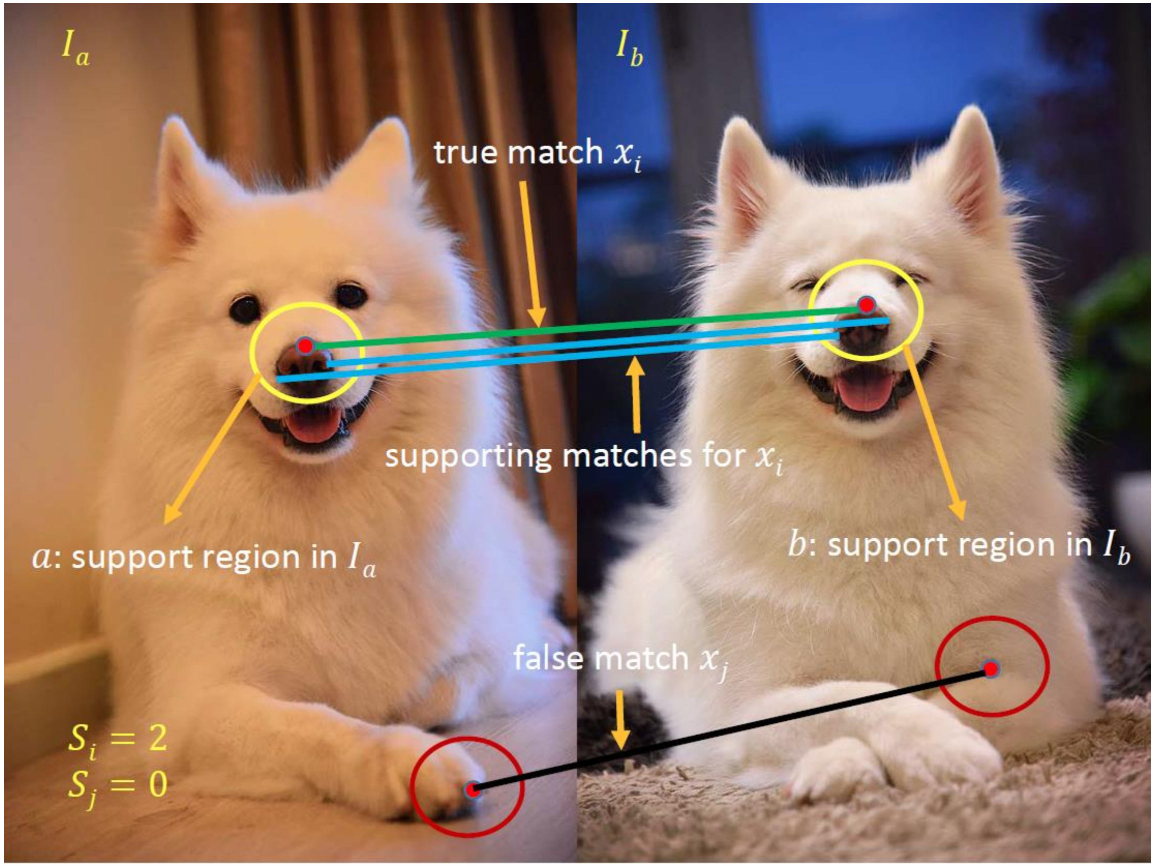
Methodology: Key idea

- Inference
  - According to the Bayesian rule, as true matches are smooth in motion space, consistent matches are thus more likely to be true.
- Key idea
  - Find smooth matches from noisy data as our proposals.
- Method

Motion Statistics → **Grid Framework** → **Motion Kernels**

Methodology: Motion Statistics

- Motion Statistics Model



$$S_i = |\mathcal{X}_i| - 1,$$

Methodology: Motion Statistics

- Distribution

$$S_i \sim \begin{cases} B(n, p_t), & \text{if } x_i \text{ is true} \\ B(n, p_f), & \text{if } x_i \text{ is false} \end{cases}$$

- Let f_a be one of the n supporting features in region a
- Let p_t , p_f be the probability that, feature fa's nearest neighbor is in region b, given $\{a, b\}$ view the same and different location, respectively,

Methodology: Motion Statistics

- Event

Event	Description
f_a^t	f_a matches correctly, $p(f_a^t) = t$
f_a^f	f_a matches wrongly, $p(f_a^f) = 1 - t$
f_a^b	f_a's nearest-neighbor is a feature in region b

- Assumption

$$p(f_a^b | f_a^f) = \beta m / M$$

Here, m is the number of features in region b and M is the number of features in second image. β is a factor added to accommodate violations of assumption caused by repeated patterns.

Methodology: Motion Statistics

- Probability

$$p_t = p(f_a^t) + p(f_a^f)p(f_a^b | f_a^f)$$
$$= t + (1-t)\beta m/M$$

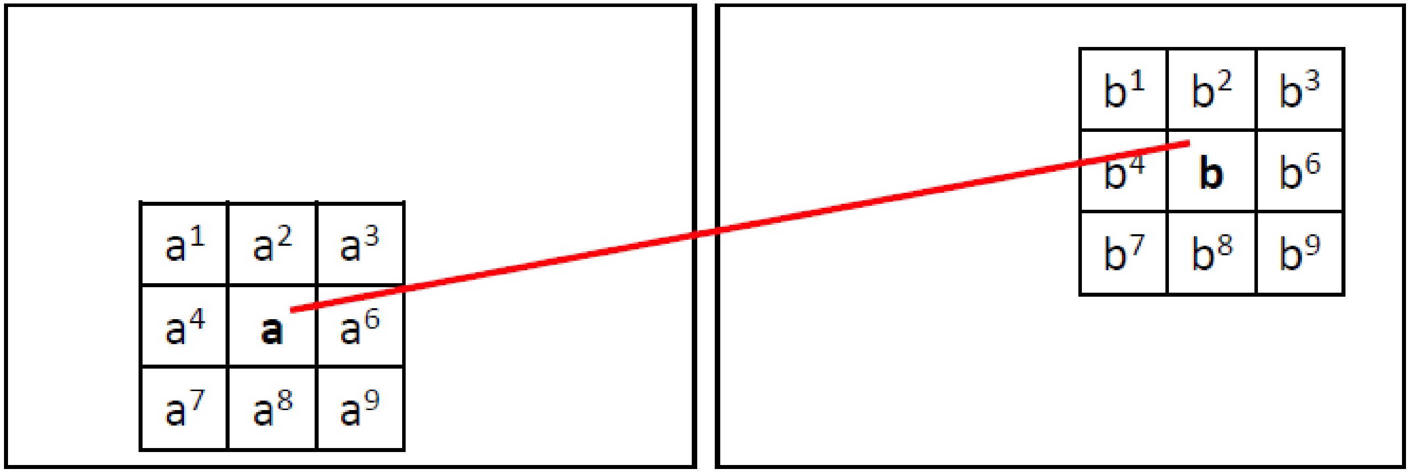
Explanation: If $\{a\ b\}$ view the same location, event f_a^b occurs when f_a matches correctly or it matches wrongly but coincidentally lands in region b.

$$p_f = p(f_a^f)p(f_a^b | f_a^f)$$
$$= \beta(1-t)(m/M)$$

Explanation: If $\{a\ b\}$ view the different location, event f_a^b occurs only when f_a matches wrongly and coincidentally lands in region b.

Methodology: Motion Statistics

- Multi-region Generalization



$$S_i = \sum_{k=1}^{K} |\mathcal{X}_{a^k b^k}| - 1$$

Methodology: Motion Statistics

- Distribution

$$S_i \sim \begin{cases} B(Kn, p_t), & \text{if } x_i \text{ is true} \\ B(Kn, p_f), & \text{if } x_i \text{ is false} \end{cases}$$

- Mean & Variance

$$\begin{array}{ll} \{m_t = Knp_t, s_t = \sqrt{Knt(1-p_t)}\} & \text{if } x_i \text{ is true} \\ \{m_f = Knp_f, s_f = \sqrt{Knp_f(1-p_f)}\} & \text{if } x_i \text{ is false} \end{array}$$

Methodology: Motion Statistics

- Analysis
  - Partionability

$$P = \frac{m_t - m_f}{s_t + s_f} = \frac{Knp_t - Knp_f}{\sqrt{Knp_t(1-p_t)} + \sqrt{Knp_f(1-p_f)}}$$

  - Quantity-Quality equivalence:

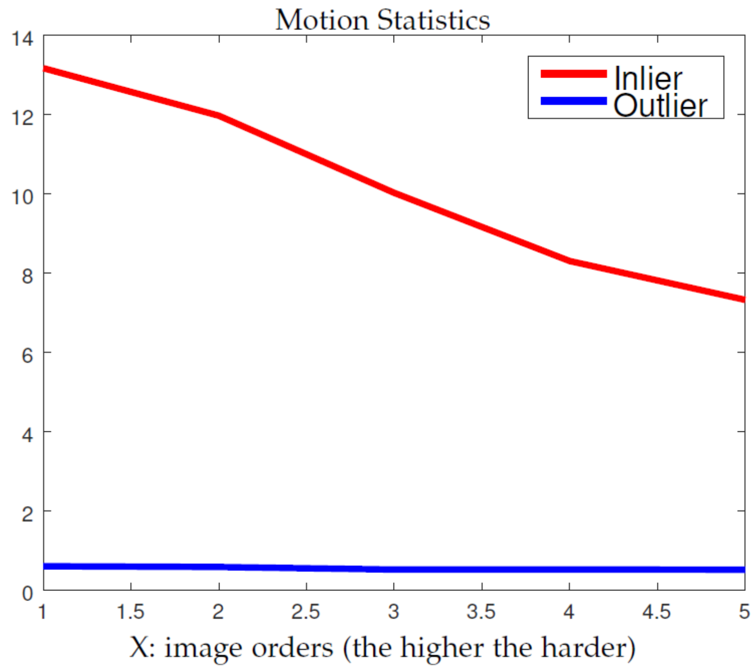
$$P \propto \sqrt{Kn}.$$

  - Relationship to Descriptors:

$$\lim_{t \to 1} m_t \to Kn, \quad \lim_{t \to 1} m_f \to 0, \quad \lim_{t \to 1} P \to \infty.$$

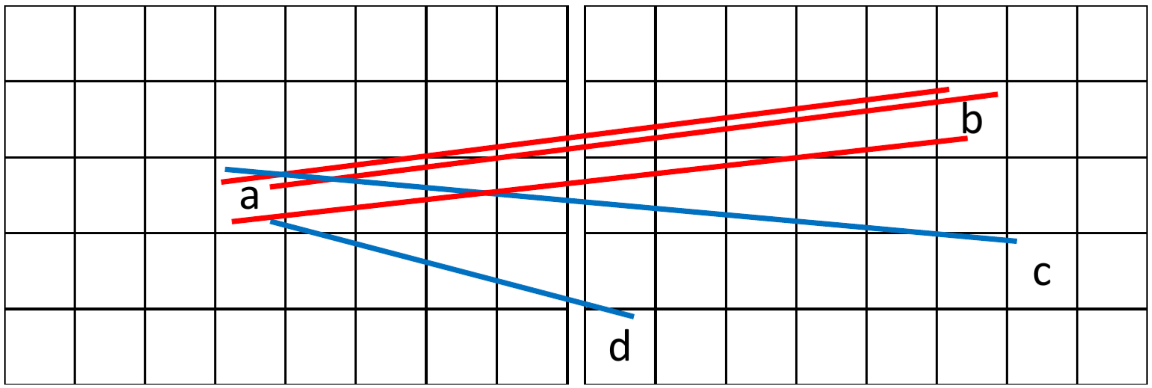
Methodology: Motion Statistics

- Experiments on real data:



The model is evaluated on Oxford Affine Dataset. Here, we run SIFT matching and label all matches as inlier or outlier according to the ground truth. we count the supporting score for each match in a small region.

Algorithm: Grid Framework

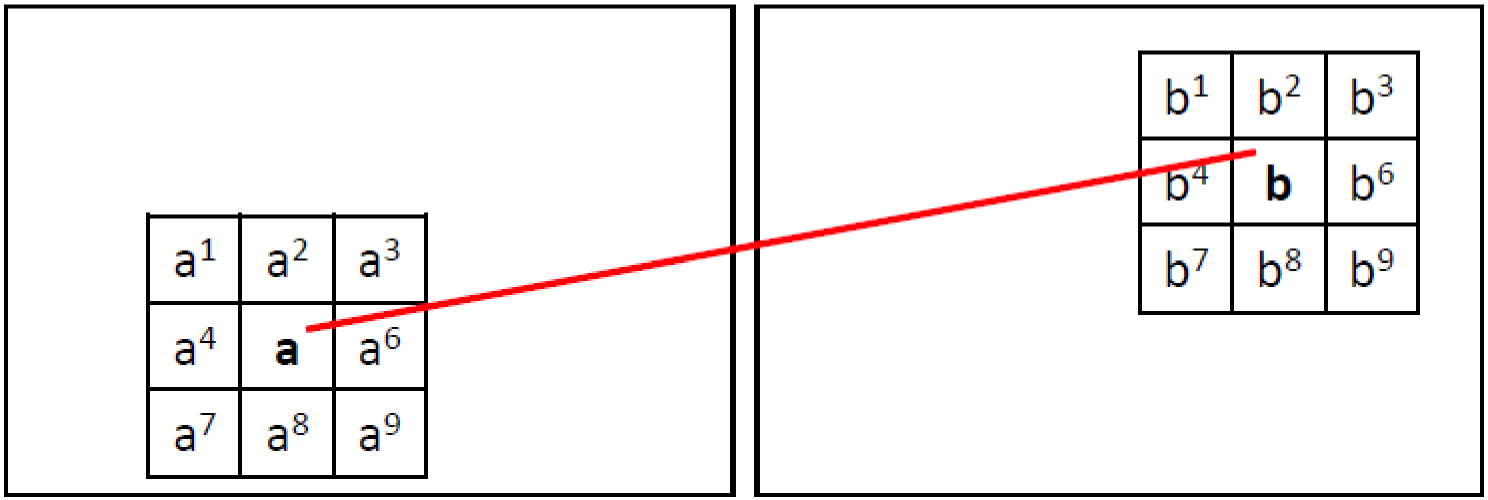


O(N) → O(1)!

- Grid Framework
  - Both images are segmented by a pre-defined grid.
  - Calculating the Motion Statistics for cell-pairs instead of each feature correspondence.

Algorithm: Motion Kernels

- Basic Motion Kernel



$$S_{ij} = \sum_{k=1}^{K=9} |\mathcal{X}_{i^k j^k}|$$

Algorithm: Motion Kernels

- Generalized Motion Kernels (Extension*)
  - Rotation

a^1	a^2	a^3
a^4	**a**	a^6
a^7	a^8	a^9

Fixed

b^1	b^2	b^3
b^4	**b**	b^6
b^7	b^8	b^9

(1)

b^4	b^1	b^2
b^7	**b**	b^3
b^8	b^9	b^6

(2)

b^7	b^4	b^1
b^8	**b**	b^2
b^9	b^6	b^3

(3)

...

b^2	b^3	b^6
b^1	**b**	b^9
b^4	b^7	b^8

(8)

  - Scale
    - Varying the cell size of the second image by a scale factor.

Algorithm: Empirical parameters

- How many grid-cells should be used?
  - Too fine: weak statistics and low efficiency.
  - Too coarse: low accuracy
  - The empirical results show 20 x 20 is a good choice.

- How to set the threshold?

$$\tau = m_f + \alpha s_f \qquad \tau \approx \alpha s_f \approx \alpha\sqrt{n}$$

$$\text{cell-pair } \{i,j\} \in \begin{cases} \mathcal{T}, & \text{if } S_{ij} > \tau_i = \alpha\sqrt{n_i} \\ \mathcal{F}, & \text{otherwise} \end{cases}$$

Algorithm: GMS

- Grid Motion Statistics Algorithm

Algorithm 1 Grid Motion Statistics

Input: $\mathcal{X}$, s, r {Correspondences, scale, rotation}
Output: $Inliers$

```
G_1, G_2 = GenerateGrids(s)
K = GenerateMorionKernel(r)
for i = 1 to |G_1| do
    j = 1;
    for k = 1 to |G_2| do
        if |X_ik| > |X_ij| then
            j = k;
        end if
    end for
    S_ij, τ_i = ComputeGMS(K) {Eq. (13)(14)}
    if S_ij > τ_i then
        Inliers = Inliers ∪ X_ij;
    end if
end for
Repeat algorithm with gird patterns shifted by half cell-
width in the x, y and both x and y directions.
return Inliers
```

Algorithm: Full Feature Matching

- Full feature matching pipeline

Algorithm 2 Feature Matching with GMS

Input: I_a, I_b, $Scale$, $Rotation$ {Two input images}
Output: $Inliers$

```
Extract Features and Descriptions: F_a, D_a, F_b, D_b
Find Nearest Neighbour Matches: 𝒳
Initialise Inliers and number
number = 0
for all s ∈ Scale do
    for all r ∈ Rotation do
        inlier = gms(𝒳, s, r)
        if |inlier| > number then
            number = |inlier|
            Inliers = inlier
        end if
    end for
end for
return Inliers
```

Algorithm: Run time

- Run time on Image pairs
  - ORB feature extraction(about 35ms on cpu)
  - Nearest Neighbor Matching(106ms on cpu, 25ms on gpu)
  - GMS(1ms on cpu)
  - Overall : 1000 / (2 * 35 + 25 + 1) = 10.42fps

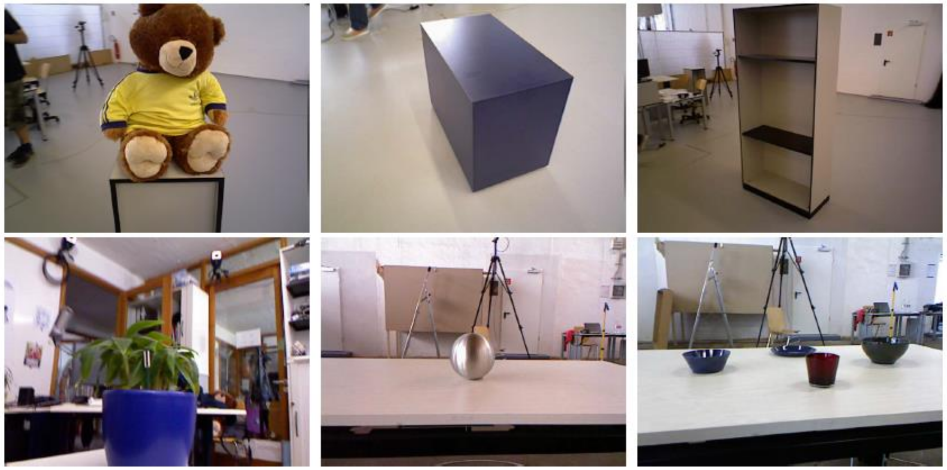
- Real time on Video data
  - ORB and NN can run parallelly on video sequence.
  - Overall : 1000 / 35 = 28.57fps

Evaluation

- Dataset

Dataset	*TUM* [38]	*Strecha* [37]	*VGG* [25]	*Cabinet* [38]
Full name	RGB-D SLAM Dataset and Benchmark	Dense Multiview Stereo Dataset	Affine Covariant Regions Datasets	A subset of TUM dataset
Image pairs	3141	500	40	578
Ground truth	Camera pose, Depth	Camera pose, 3D model	Homography	Camera pose, Depth
Description	Including all image condition changes	Well-textured images	Viewpoint change, zoom+rotation,blur	Low-texture images with strong light

- Capture of TUM dataset



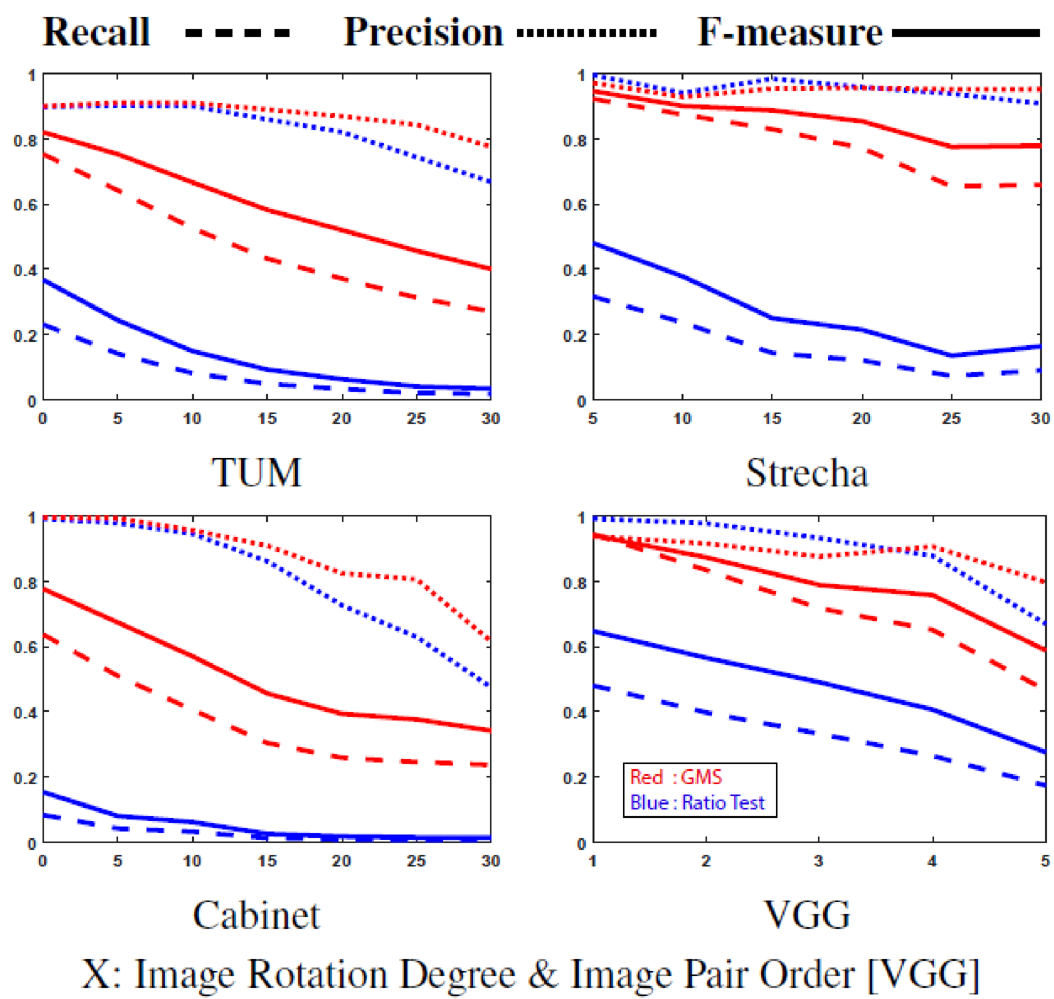
Evaluation

- Capture of Strecha dataset

- Capture of VGG dataset

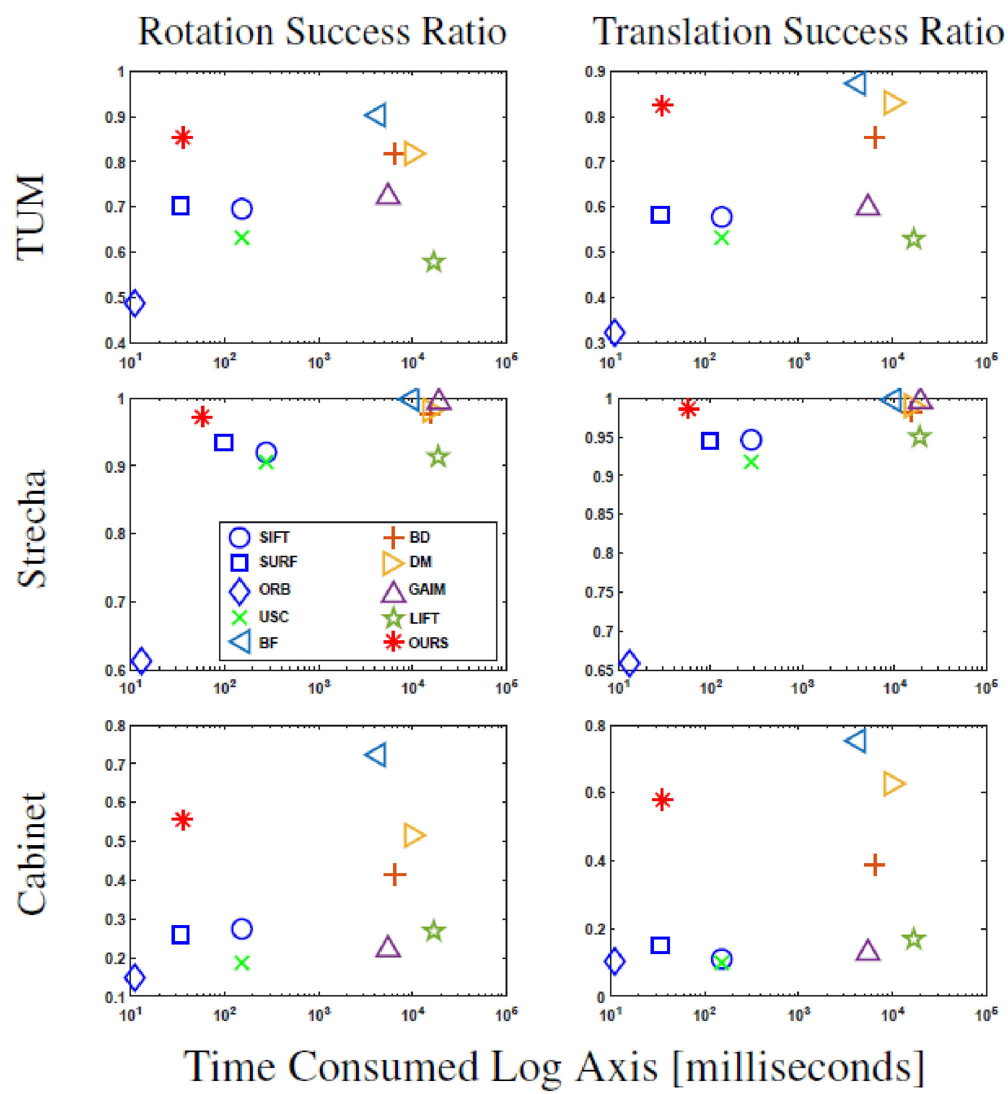
Evaluation

- Matching ability



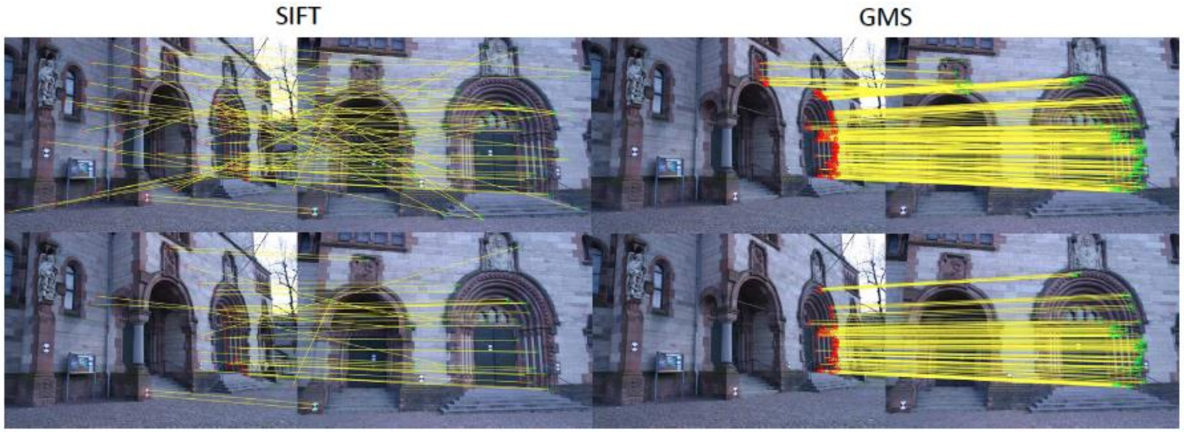
Evaluation

- Pose Estimation

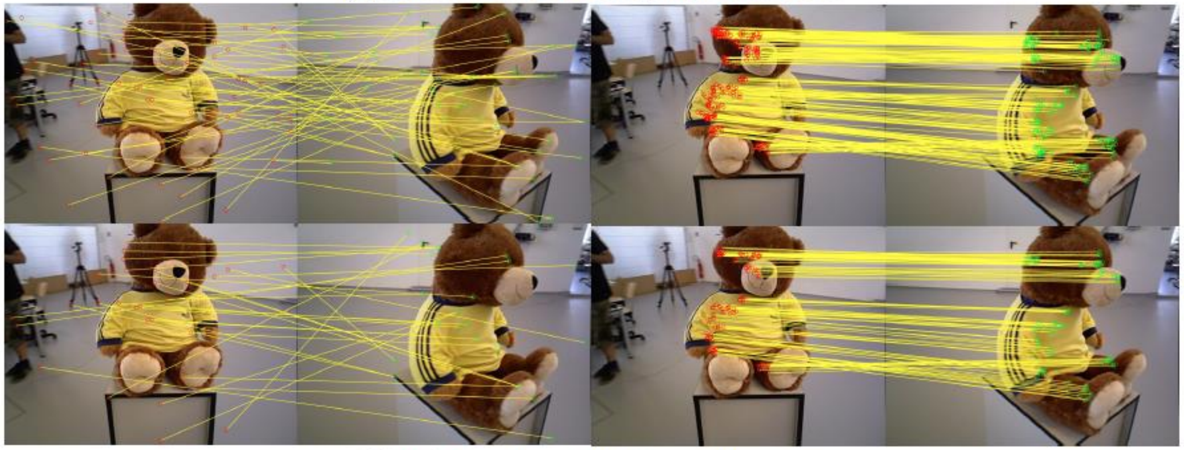


Evaluation

- Wide-baseline matching



(a) Wide baseline matching on planar scenes

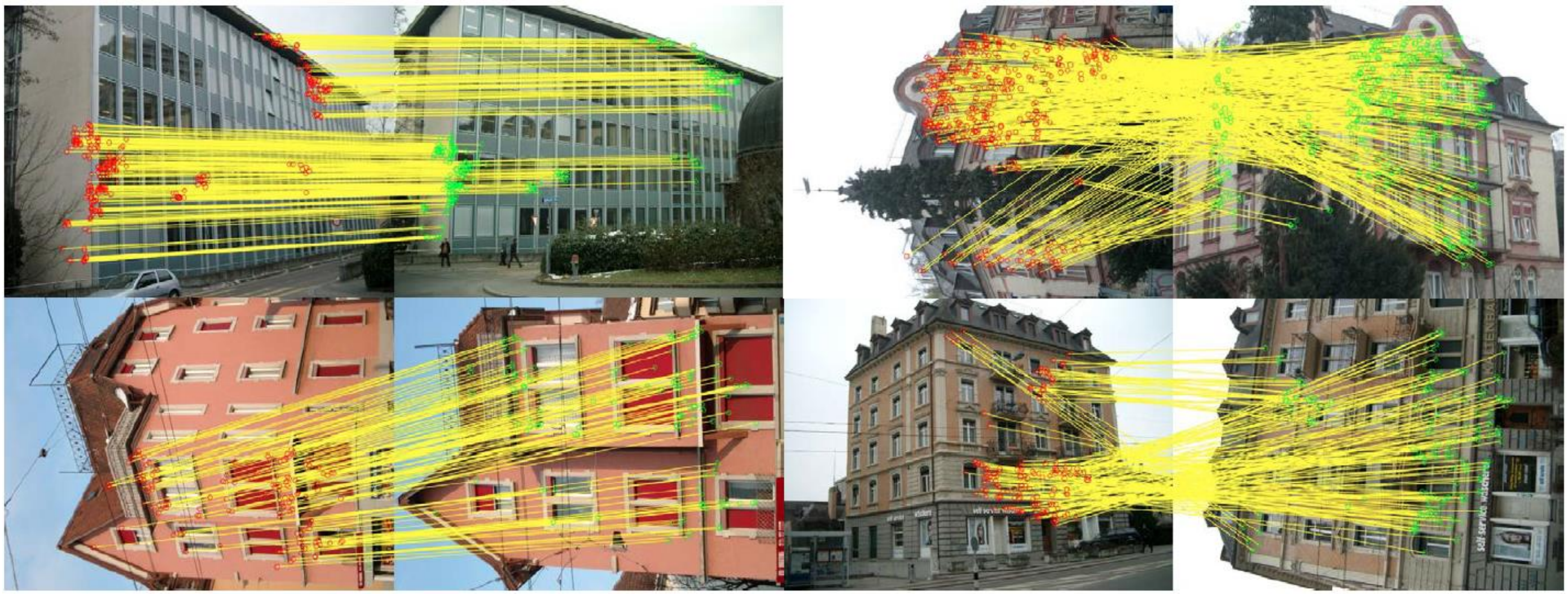


(b) Wide baseline matching on 3D scenes

In both graphs, the first row shows initial results and the second row illustrates the matches after RANSAC.

Evaluation

- GMS on Images with Repetitive Structures

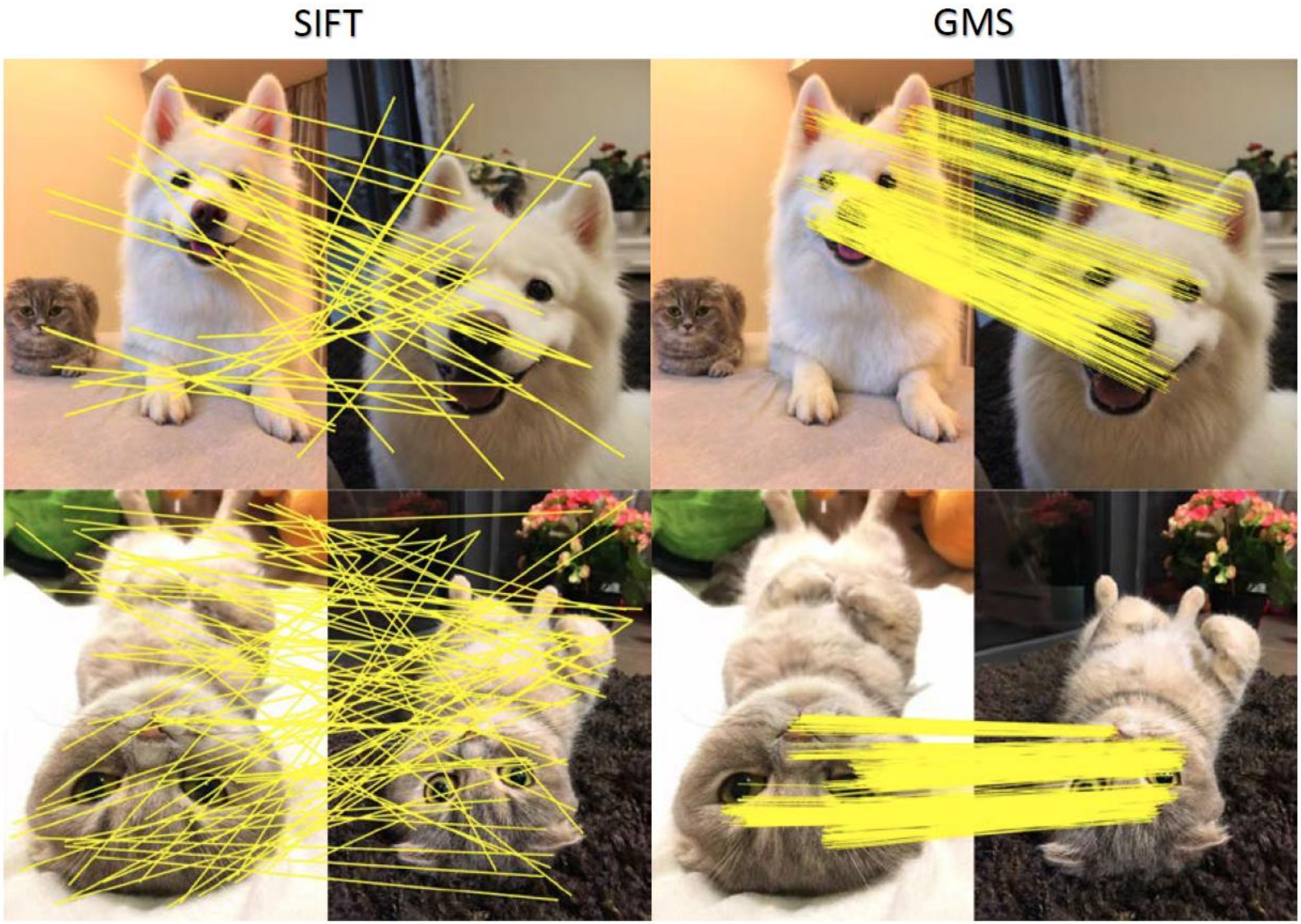


Images are selected by [1], where many state-of-art matchers fail and SIFT fails all.

[1] Epipolar Geometry Estimation for Urban Scenes with Repetitive Structures, IEEE TPAMI, 2014, Kushnir et. al.

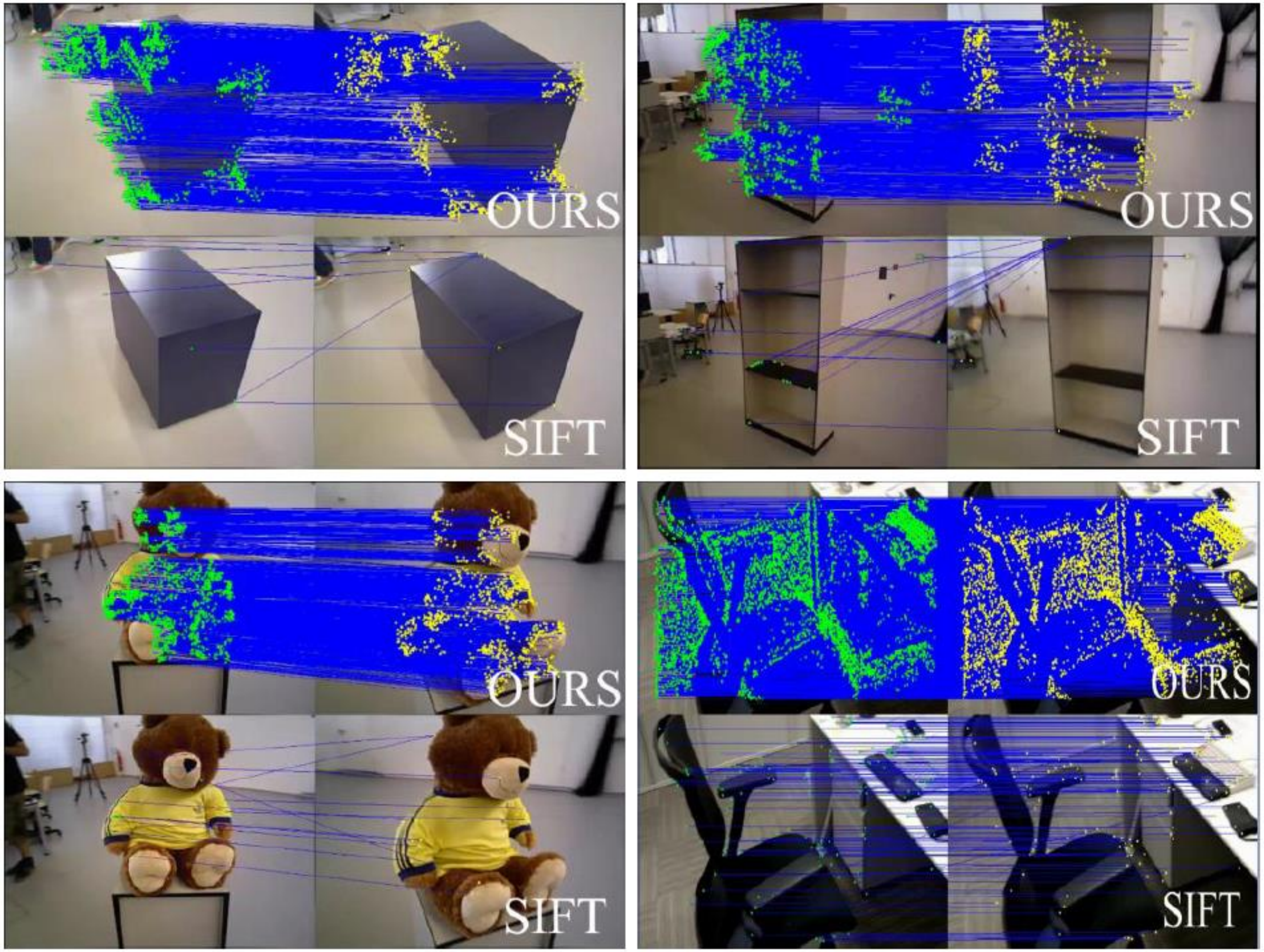
Evaluation

- Non-rigid object



Evaluation

- Video Demo(screen shot)



Share

- JiaWang's Home Page
  - http://jwbian.net/
- Project Page
  - http://jwbian.net/gms/
- Code on GitHub:
  - https://github.com/JiawangBian/GMS-Feature-Matcher
- Videos on YouTube:
  - https://youtu.be/3SlBqspLbxI
- Links to CODE and RepMatch
  - http://www.kind-of-works.com/

Q&A