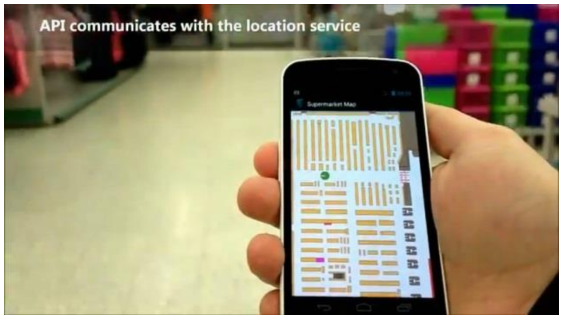
E^2C^2: Efficient and Effective Camera Calibration In Indoor Environments

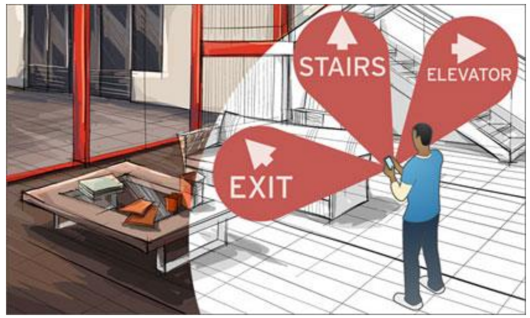
UbiComp '15

Huan Li, Pai Peng, Hua Lu, Lidan Shou, Gang Chen

Indoor & Augmented Reality

- Indoor spaces accommodate huge portions of people's lives.
- Mobile Augmented Reality (AR) applications enable users to better know the "mysterious" indoor spaces.

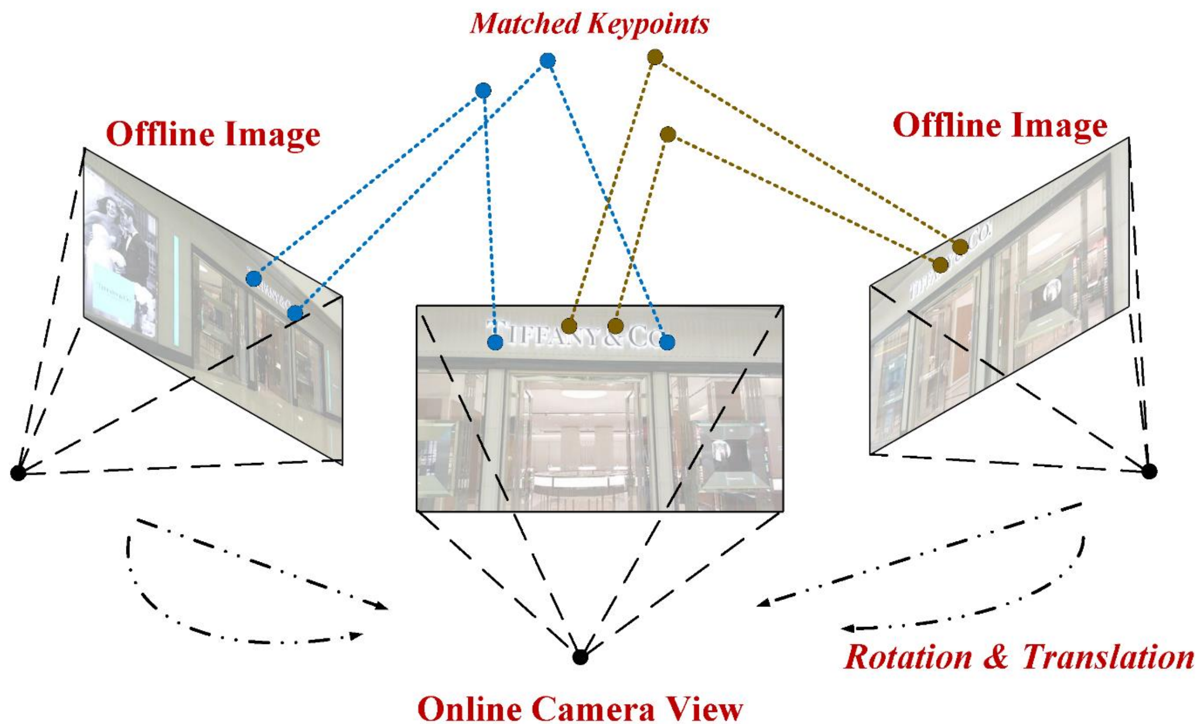




- Such Mobile AR applications highly rely on camera calibration techniques.

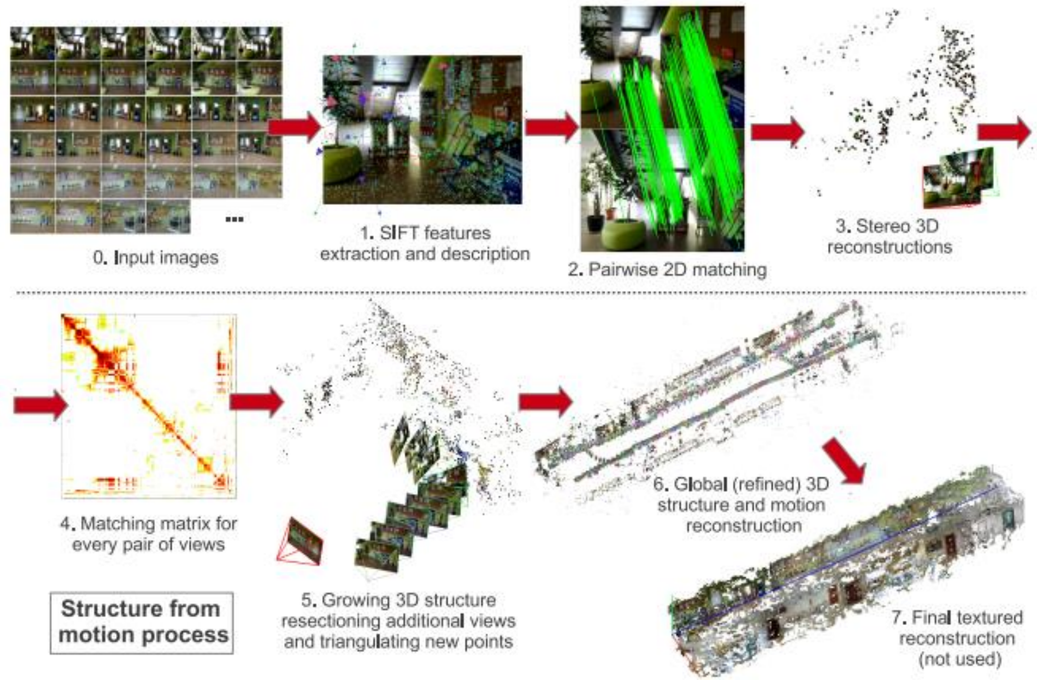
Camera Calibration[1]

- To estimate the extrinsic camera parameters: relative location and orientation of the online camera.
- Finding sufficient keypoints matched with current camera view (query photo) from multiple beforehand captured photos.
- Pairwise visual matching is computationally expensive and not suitable for real-time service.



Accelerate Online Camera Calibration

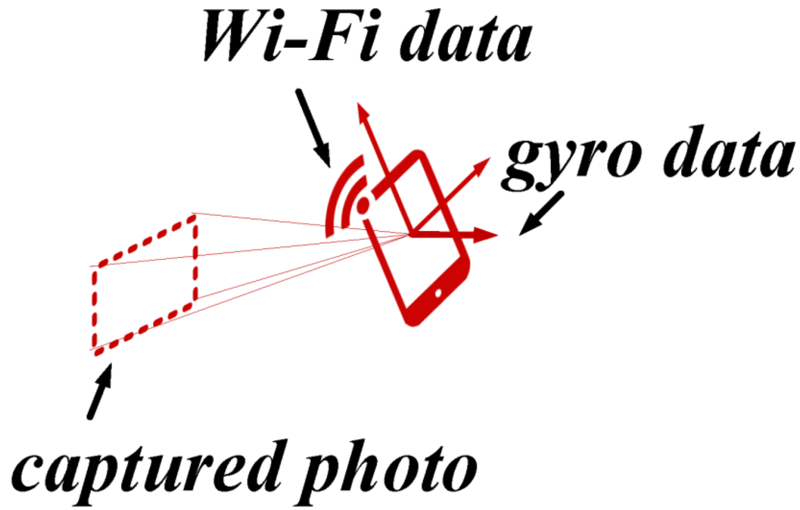
- Work[2] : Makes good use of rich sensors embedded in mobile phones.
- ① Offline Phase: record qualified video, and build a 3D model with overwhelming SIFT points



- ② Online Phase: Wi-Fi positioning is used as a way of pruning, only select a portion of SIFT for calibration.
- Drawback: less applicable and scalable; a central server is needed for 3D model building.

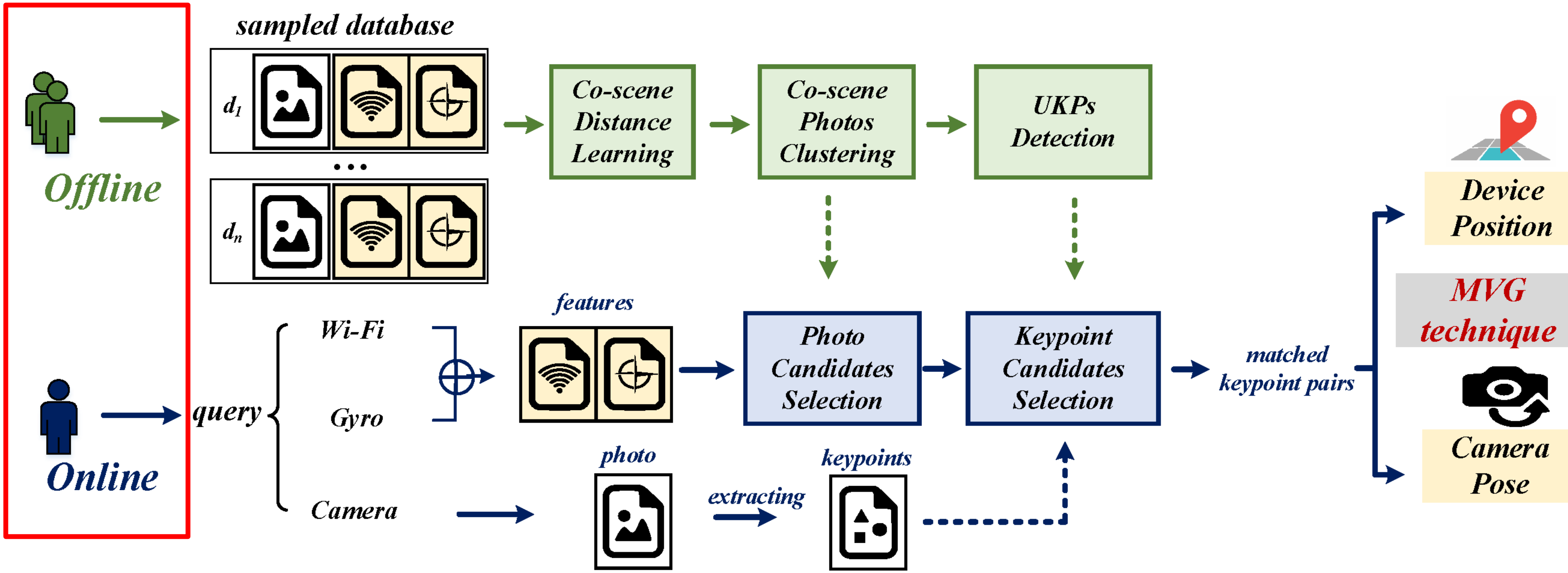
Our Work E^2C^2

- Motivation: make the online calibration efficient and effective, and keep the offline phase simple.
- Represent photos with other corresponding sensors.



- In offline phase: sample a few photos that are labeled beforehand with Wi-Fi and gyro information.
- In online phase: quickly select a small number of offline photos and their keypoints for calibration.
- Lightweight, scalable and extendable.

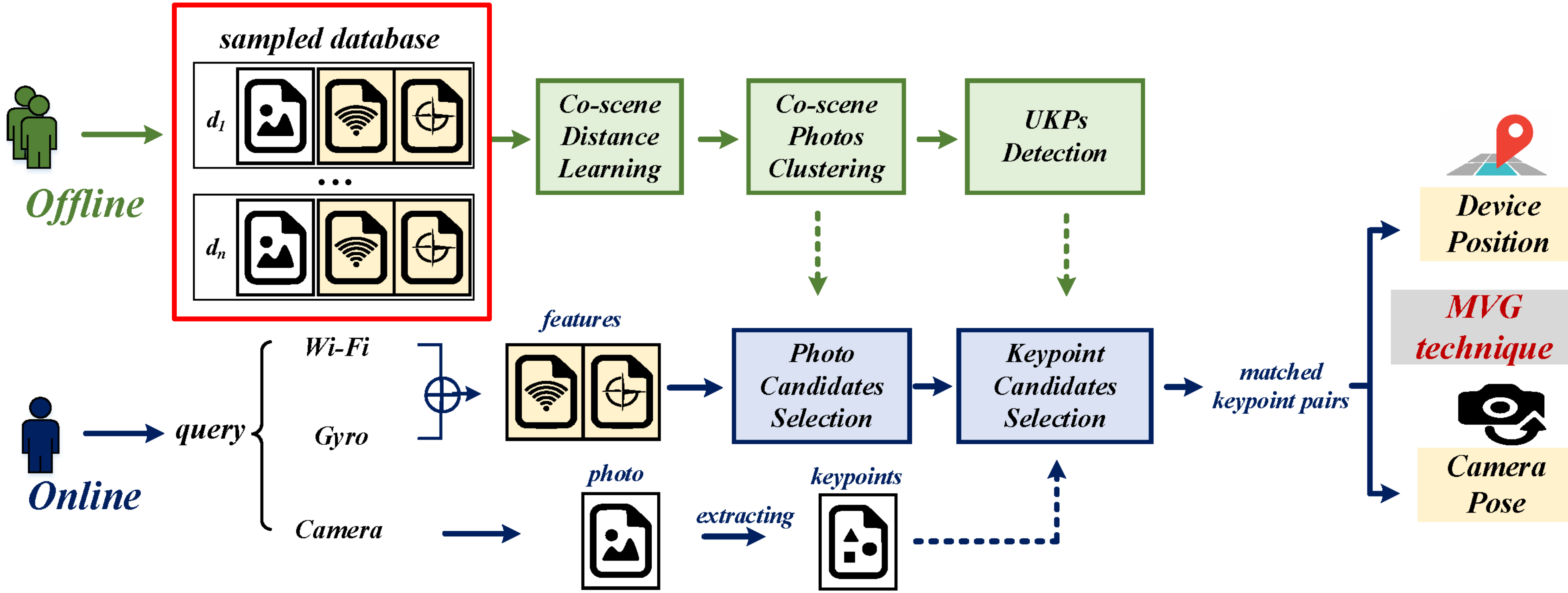
Framework



Offline model construction phase &

Online query processing phase.

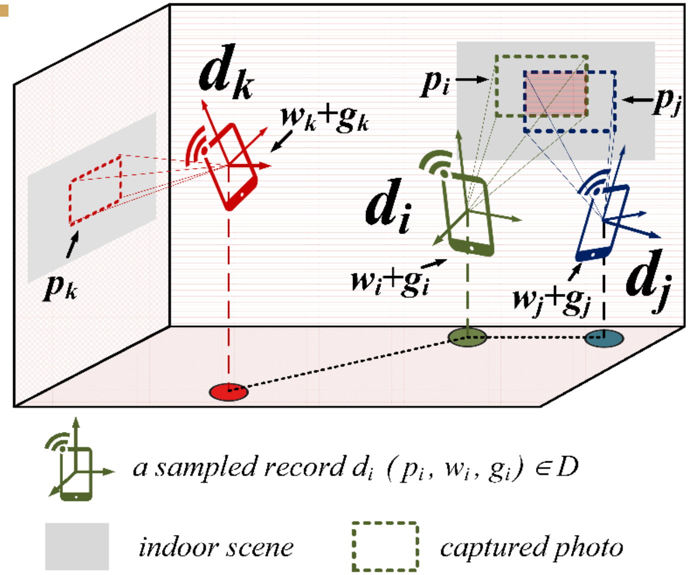
Framework



Sample the space by capturing a set of photos along with associated Wi-Fi and gyro sensor data.

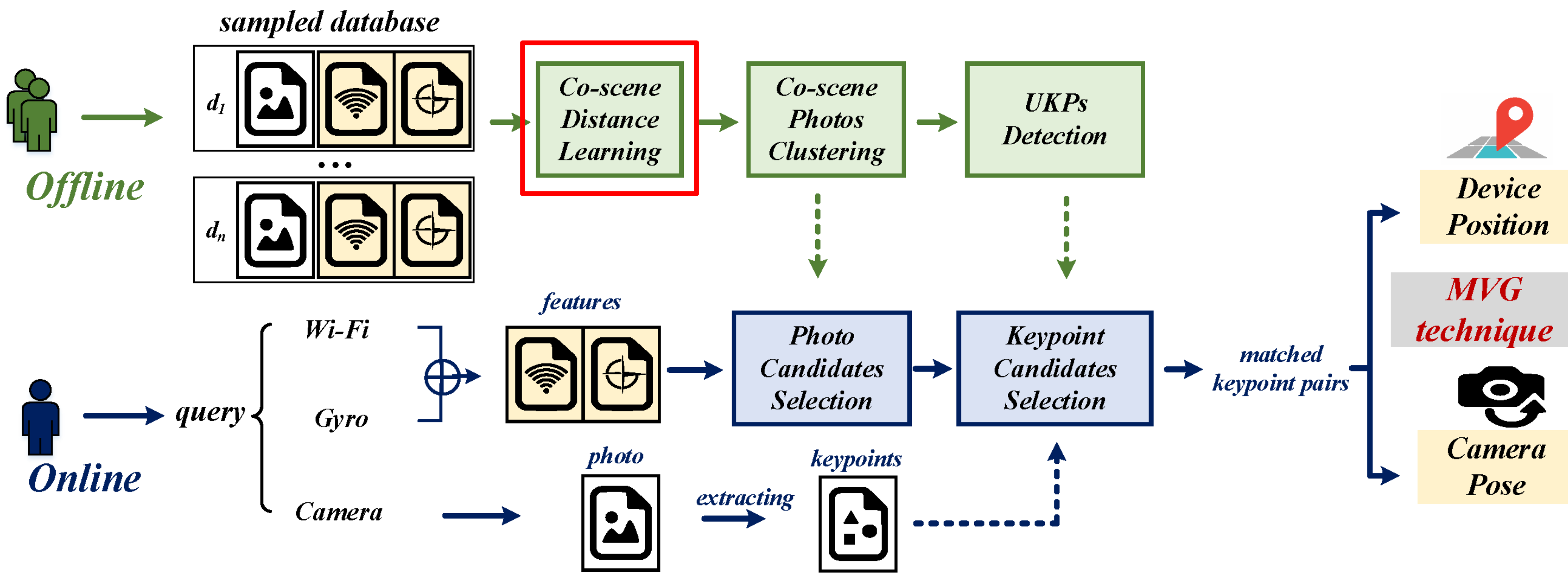
Data Sensing

- A database D , each record $d \in D$ has three fields:
  - a photo-captured photo p
  - a set of consecutive Wi-Fi signals w
  - device's gyro information g



- A constraint set C, each entry $c \in C$ records a similarity or dissimilarity constraint between elements in D :
  - a triple $(d_i^{(1)}, d_i^{(2)}, y_i)$
  - $d_i^{(1)} \in D$, $d_i^{(2)} \in D$ and $y_i \in \{-1, +1\}$ determines if $d_i^{(1)}$ and $d_i^{(2)}$ are similar or not
  - C is small-sized (e.g., 40 or 60)

Framework



Learn an effective distance metric called "co-scene distance".

Co-scene Distance Learning

- **Wi-Fi + gyro** are used to reduce the search space of candidate photos.
- Assume a Wi-Fi signal vector $\boldsymbol{w} \in \mathbb{R}^n$ is $\boldsymbol{w} = (\boldsymbol{w_1}, \dots, \boldsymbol{w_n})$, a gyro feature $\boldsymbol{g} \in \mathbb{R}^4$ is a quaternion $\boldsymbol{g} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\varpi})$.
- A **synthetic feature vector** by a linear combination:
- $$\boldsymbol{wg} = (\boldsymbol{w}, \boldsymbol{\lambda g}) = (\boldsymbol{w_1}, \dots, \boldsymbol{w_n}, \boldsymbol{\lambda\alpha}, \boldsymbol{\lambda\beta}, \boldsymbol{\lambda\gamma}, \boldsymbol{\lambda\varpi})$$
- Not reasonable to directly compare the query with these feature vectors by a naïve combination of Cosine or Euclidean similarities.

Co-scene Distance Learning

- **Mahalanobis distance** between two vector x, x' :
- $$D_M(x,x') = \sqrt{(x-x')^T M(x-x')} = \sqrt{(x-x')^T L^T L(x-x')}$$
- Based on the synthetic features, we define the **co-scene distance** D_{cs} between two photos p_i, p_j by the definition of ***Mahalanobis distance*** as:
- $$D_{cs}(p_i, p_j) = \sqrt{(wg_i - wg_j)^T M_{wg}(wg_i - wg_j)}$$
- To regularize the metric parameter M_{wg} while **satisfying the similarity and dissimilarity constraints** on set C.

Information-Theoretic Metric Learning (ITML) [3]

- **ITML** is effective to optimize the Mahalanobis distance based metric learning problem.

- Select a **target matrix** M_0 (usually simple, e.g., Identify matrix), keep M_{wg} as **close** as M_0 and **satisfy the constraints** on C :

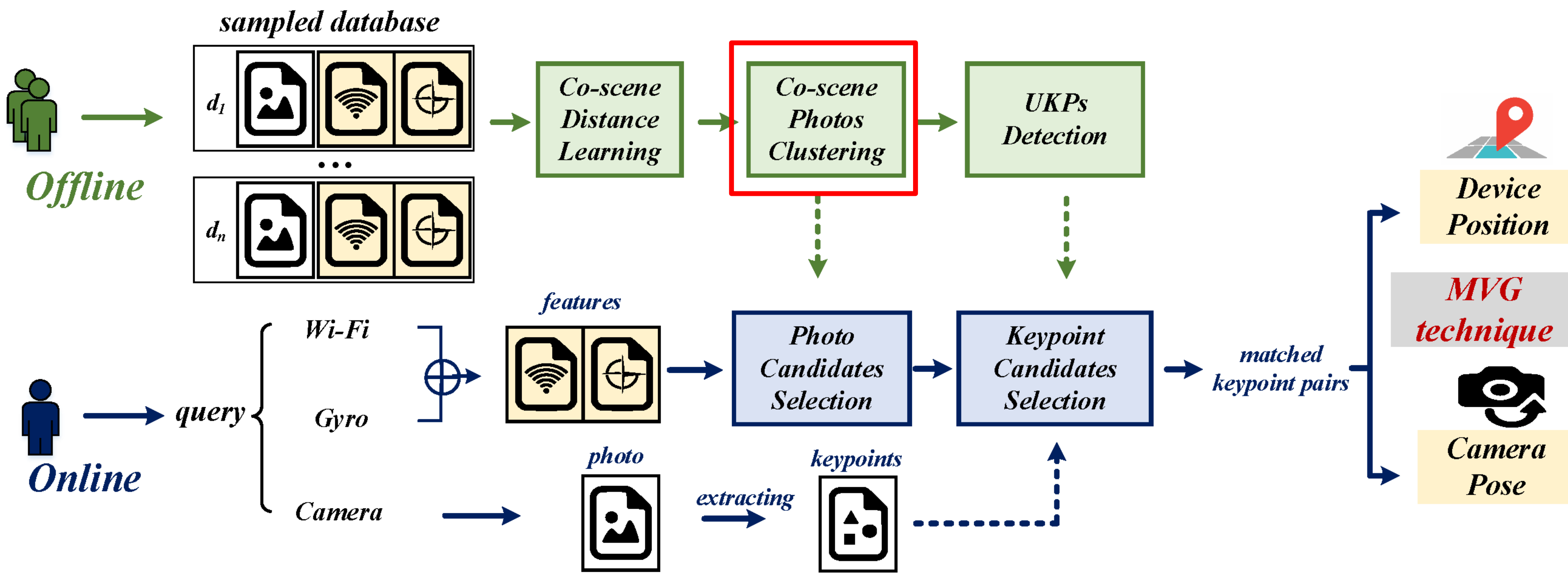
- $$\min_{M_{wg}} KL(p(d; M_0) \mid p(d; M_{wg}))$$

- $$s.t. \quad D_{CS}(p_i, p_j) \le l \qquad (d_i, d_j, 1) \in C$$

- $$D_{CS}(p_i, p_j) \ge v \qquad (d_i, d_j, -1) \in C$$

- l, v are threshold parameters, **KL divergence** measures the ″closeness″ between two distributions.

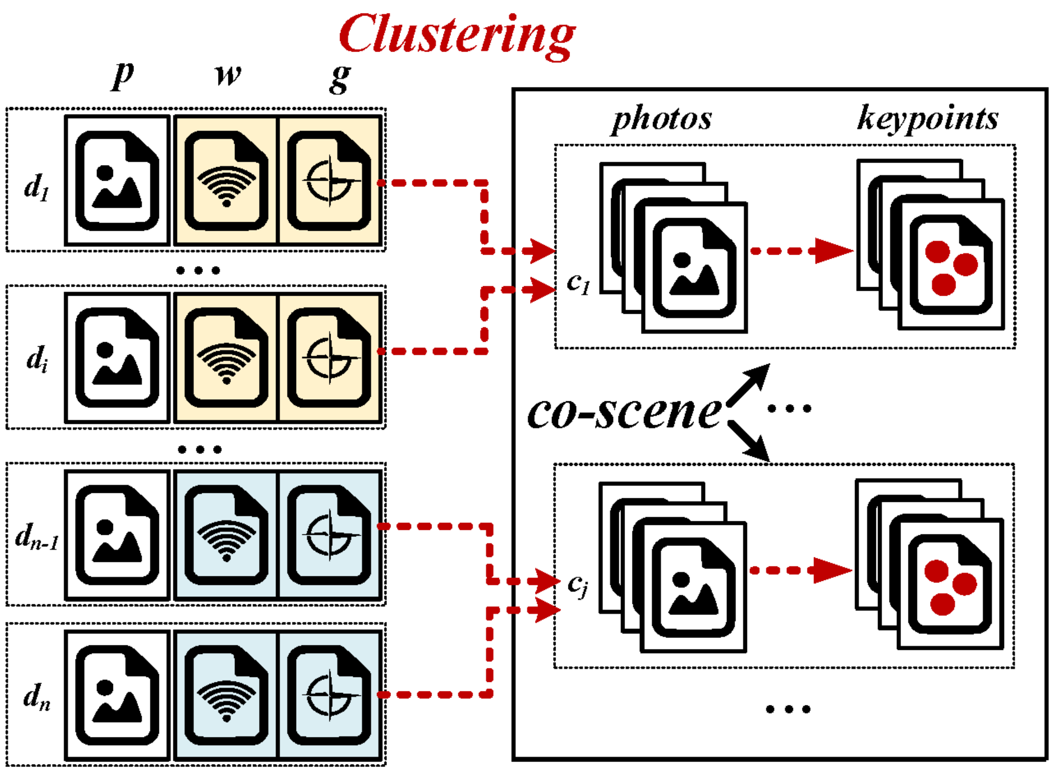
Framework



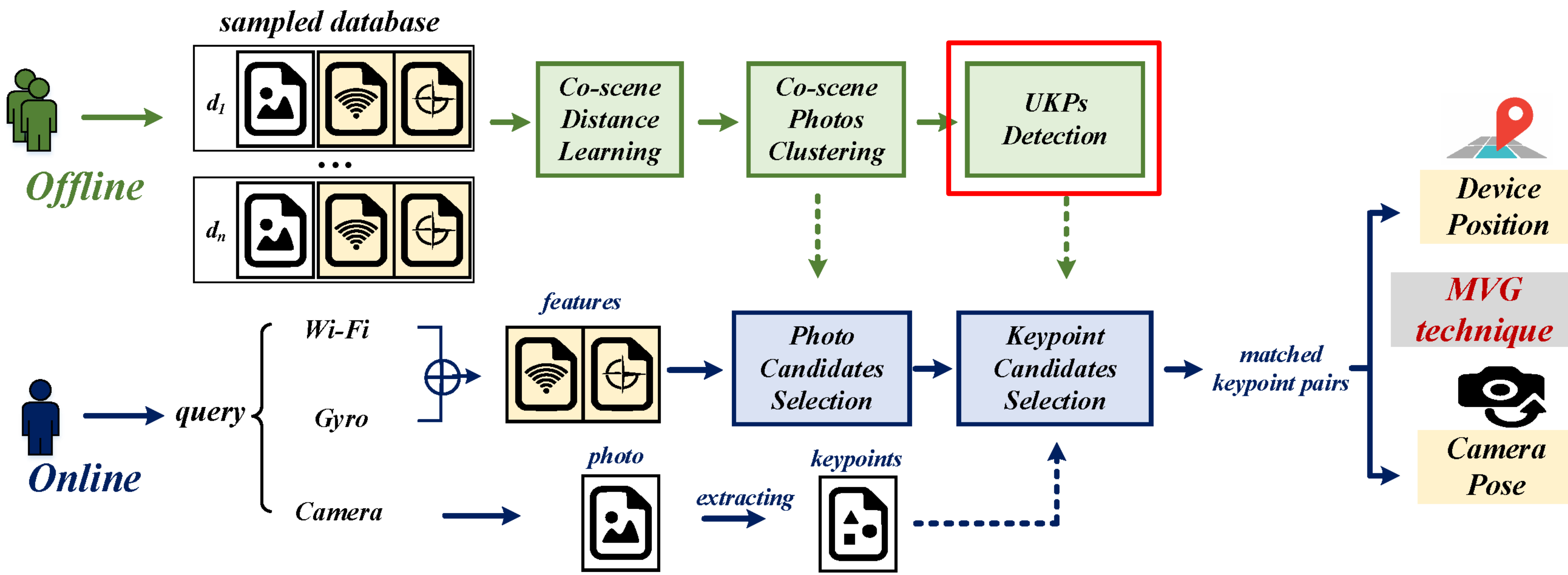
Cluster the captured photos according to the learned distance. Photos in each cluster form a *co-scene*.

Co-scene Photo Clustering

- Cluster the sampled photos into k groups.
- Photos in each group form a co-scene.
- Co-scene is likely to share the same visual contents.
- k is a hyperparameter and tuned by cross validation.



Framework

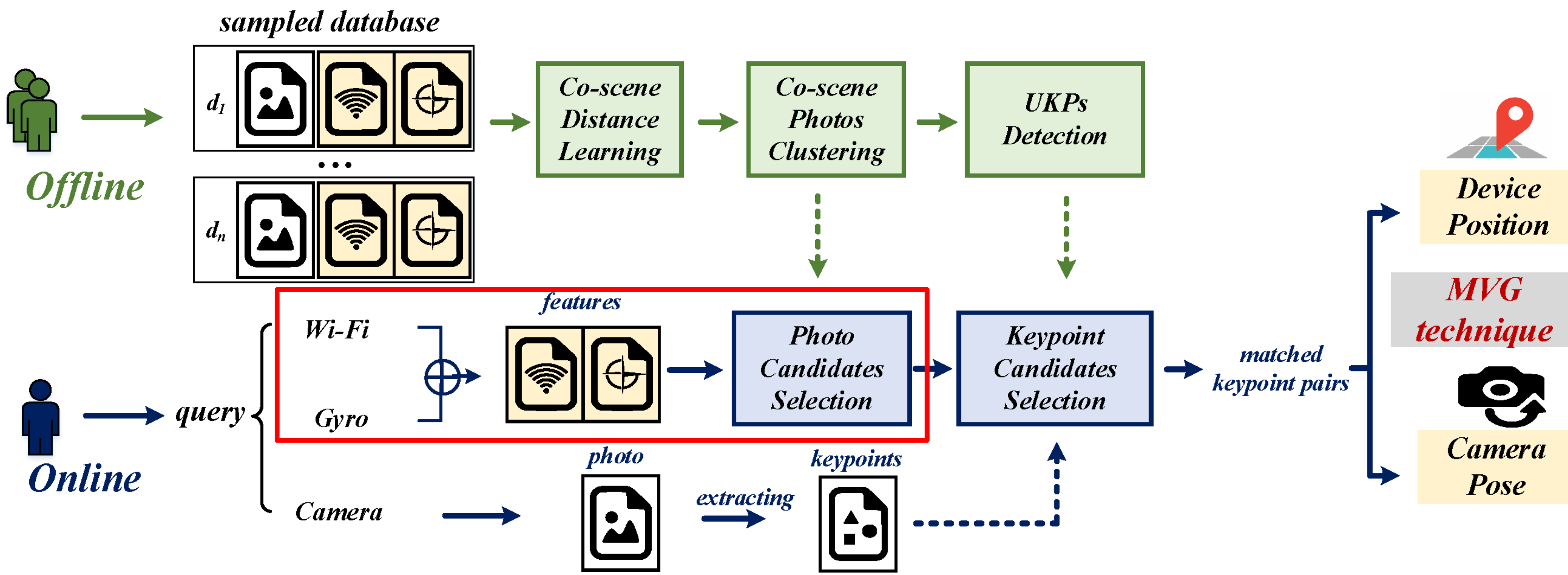


Detect a subset of keypoints that are frequently appeared in each co-scene. These keypoints are thus called "useful keypoints" (UKPs).

UKPs Detection

- **Selected co-scene** can help reduce photo candidates.
- NOT ENOUGH: if we **iterate through** all keypoints located in one or several co-scenes.
- To count the frequency of each keypoints by matching photos pairwisely in the same co-scene.
- Only **frequently appearing** (matched) keypoints are useful for further calibration.
- Such keypoints are called "**UKPs**".

Framework

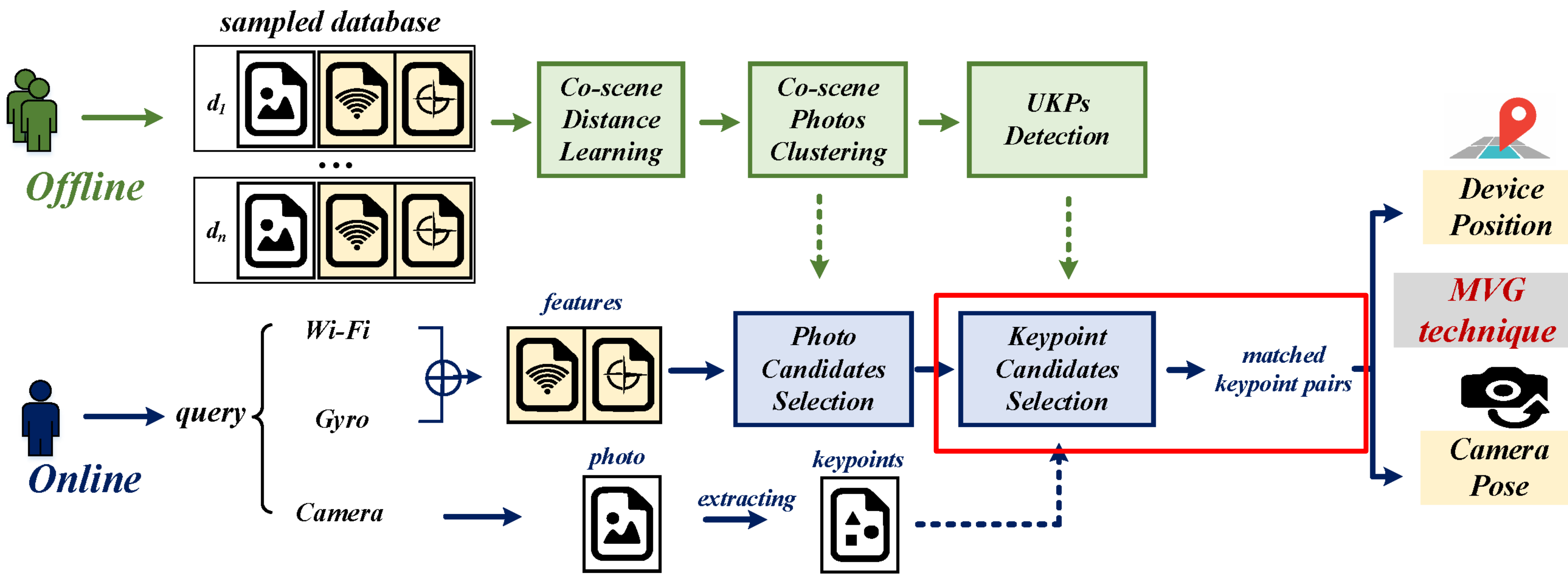


Photos in the nearest co-scene are selected as candidates.

Photo Candidates Selection

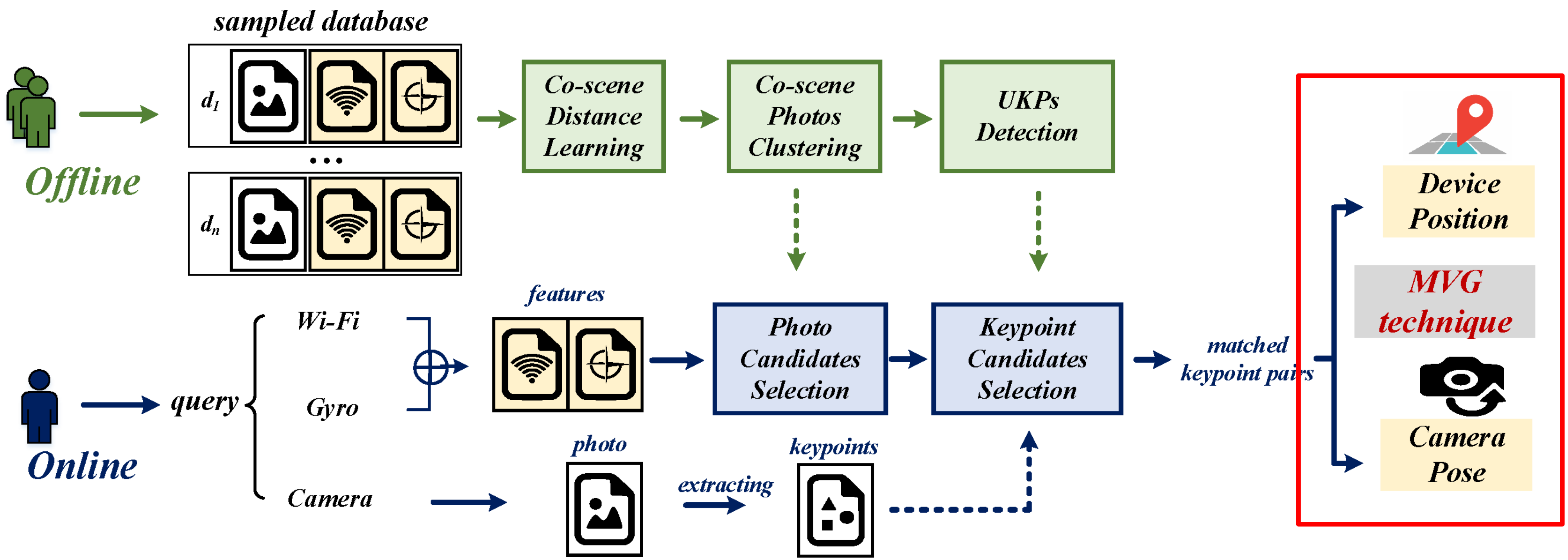
- For an issued query:
- First compose its synthetic feature according to its Wi-Fi and gyro data.
- $$wg_q = \left(w_q, \lambda g_q\right) = (w_{q1}, \dots, w_{qn}, \lambda\alpha_q, \lambda\beta_q, \lambda\gamma_q, \lambda\varpi_q)$$
- Find the nearest co-scene by comparing wg_q with all cluster centroids based on the learned metric.
- All the photos in that selected co-scene are considered as photo candidates.

Framework



Compare the keypoints in the query with these UKPs detected from the photo candidates to confirm the ultimate matched keypoint pairs.

Framework



Infer the extrinsic camera parameters (camera pose and device position) with multiple view geometry techniques [2].

Data Set

- Develop an Android App, which records photos together with Wi-Fi signals and gyro data at shooting time.
- 4 Volunteers, (3 shopping malls + 1 office building)
- Require volunteers indicate a few similar and dissimilar pairs through the interface.
- 50 queries for each place by 2 other volunteers.

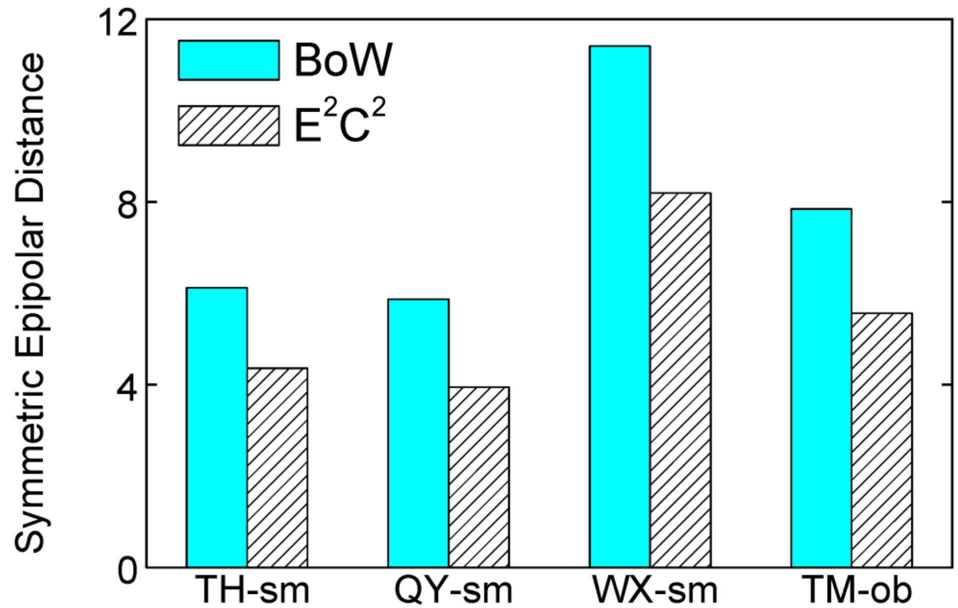
dataset	#images	#floors
TH-sm	968	6
QY-sm	1151	6
WX-sm	1674	7
TM-ob	823	3

Baseline Approach

- The before-mentioned work[2] requires elaborative video recording, CANNOT conduct calibration using our lightweight dataset.
- State-of-the-art Bag-of-visual-words(BoW)[4] is used as baseline.
- Generate a dictionary with 5K visual words.
- When a query photo is issued, find the nearest photos and compare these photos with the query.
- Locate the matched keypoint pairs for calibration.

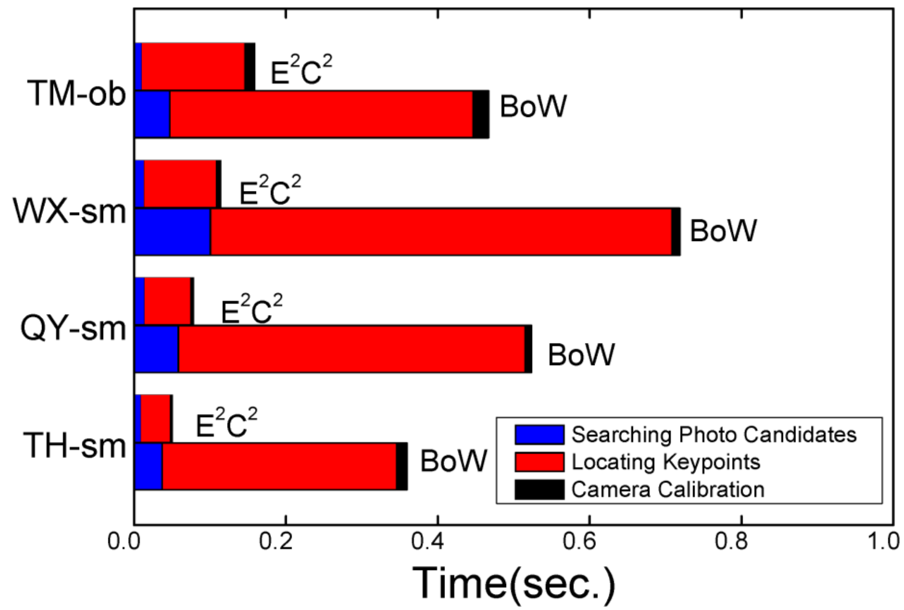
Effectiveness of Our Framework

- **Symmetric Epipolar Distance[5] is an averaged error to evaluate the calibration procedure.**
- **Lower values indicate a better estimation.**
- **Our approach outperforms BoW in all 4 datasets.**
- **The detected UKPs are more robust and discriminative.**



Efficiency of Our Framework

- Our framework beats the baseline with significantly reduced online cost.
- We lower the cost of searching photo candidates since our search space is reduced to cluster centroids.
- The cost of searching keypoints is decreased remarkably due to our UKPs detection algorithm.



Conclusion

- This work aims at **accelerating** camera calibration in an **indoor setting**, by selecting a **small but sufficient** set of keypoints.
- We use **Wi-Fi and gyro sensor data** to learn a **useful metric** for fast search of co-scene photos and locating UKPs, which get rid of expensive pairwisely visual matching.
- Our detected UKPs are robust and discriminative.
- The whole framework is **efficient** to support real-time indoor calibration.

Reference

[1] Hartley, R., and Zisserman, A. Multiple view geometry in computer vision. Cambridge university press, 2003.

[2] Ruiz-Ruiz, A. J., Lopez-de Teruel, P., and Canovas, O. A multisensor lbs using sift-based 3d models. In IPIN, IEEE (2012), 1-10.

[3] Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. Information-theoretic metric learning. In ICML, ACM (2007), 209-216.

[4] Yang, J., Jiang, Y.-G., Hauptmann, A. G., and Ngo, C.-W. Evaluating bag-of-visual-words representations in scene classfication. In Proc. ACM MIR, ACM (2007), 197-206.

[5] Fathy, M. E., Hussein, A. S., and Tolba, M. F. Fundamental matrix estimation: A study of error criteria. Pattern Recognition Letters (2011), 383-391.

Thank U ;-)