$\mathrm{E}^2\mathrm{C}^2$: Efficient and Effective Camera Calibration in Indoor Environments

Huan Li
Zhejiang University
Zheda Road 38
310027 Hangzhou, China
lihuancs@zju.edu.cn

Lidan Shou
Zhejiang University
Zheda Road 38
310027 Hangzhou, China
should@zju.edu.cn

Pai Peng
Zhejiang University
Zheda Road 38
310027 Hangzhou, China
pengpai_sh@zju.edu.cn

Ke Chen
Zhejiang University
Zheda Road 38
310027 Hangzhou, China
chenk@zju.edu.cn

Hua Lu
Aalborg University
Selma Lagerlofs Vej 300
9220 Aalborg East, Denmark
luhua@cs.aau.dk

Gang Chen
Zhejiang University
Zheda Road 38
310027 Hangzhou, China
cg@zju.edu.cn

UbiComp/ISWC '15 Adjunct, September 7–11, 2015, Osaka, Japan.
ACM 978-1-4503-3575-1/15/09.
http://dx.doi.org/10.1145/2800835.2800841

Abstract
Camera calibration helps users better interact with the surrounding environments. In this work, we aim at accelerating camera calibration in an indoor setting, by selecting a small but sufficient set of keypoints. Our framework consists of two phases: In the offline phase, we cluster photos labeled with Wi-Fi and gyro sensor data according to a learned distance metric. Photos in each cluster form a "co-scene". We further select a few frequently appearing keypoints in each co-scene as "useful keypoints" (UKPs). In the online phase, when a query is issued, only UKPs from the nearest co-scene are selected, and subsequently we infer extrinsic camera parameters with multiple view geometry (MVG) technique. Experimental results show that our framework is effective and efficient to support calibration.

Author Keywords
Camera Calibration; Indoor Space; Metric Learning

ACM Classification Keywords
H.3.m [Information Storage and Retrieval]: Miscellaneous.

Introduction
Recent years have witnessed a popular trend of mobile augmented reality (AR) applications. Such applications enable users to interact with UI components bound to the surrounding objects. For example, a tourist in a museum can hold her phone in any pose with the camera view turned

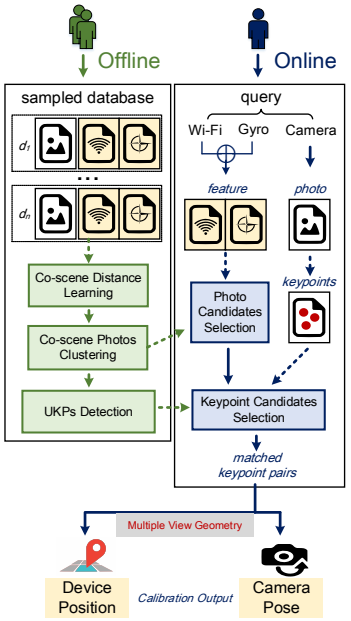

Figure 1: Framework Overview

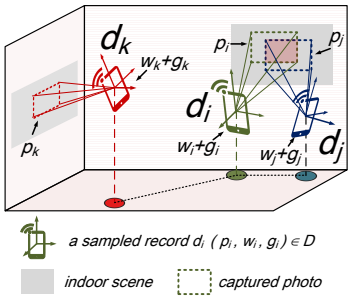

Figure 2: Sampled Database

on. In turn, the phone shows in real-time the information of all items being captured by the camera, as well as the 3D navigation information for the current tour. Functionality and user experience of such applications highly rely on camera calibration techniques in computer vision field.

Calibration [3] requires sufficient keypoints matched with the current camera view (also known as query photo) to estimate the extrinsic camera parameters. These keypoints may be distributed in multiple photos that are beforehand captured from different viewpoints. Searching those photos and determining keypoints are the key to the calibration. The simplest way to obtain those matched keypoints from a large photo database is to conduct a pairwise visual matching, which is known to be computationally expensive. As multiple built-in sensors well capture the characteristics of mobile phones [4, 5], we are thus motivated to make good use of these sensors (i.e. Wi-Fi+gyro) to accelerate the process of camera calibration. A recent work [6] uses Wi-Fi as a way of pruning candidate photos in order to conduct camera calibration. However, it needs a central server to build 3D model of the current indoor region with overwhelming SIFT points extracted from qualified video data, which makes it less applicable and scalable. Our work only needs a few sampled photos labeled beforehand with Wi-Fi and gyro information and quickly selects a small number of SIFT points for calibration, thus making our framework lightweight, scalable and extendable.

In this work, we propose a novel framework E^2C^2 which consists of an offline model construction phase and an online query processing phase as shown in Figure 1. In the offline phase, we easily sample the space by capturing a set of photos along with associated Wi-Fi and gyro sensor data. Next, the captured photos are clustered according to an effective distance metric learned from the Wi-Fi and gyro data. Photos in each cluster form a *co-scene*. We further detect a subset of keypoints that are frequently appeared in each co-scene. These keypoints are thus called *"useful keypoints" (UKPs)*. When a query is issued in the online phase, photos in the nearest co-scene are selected as candidates. We ultimately obtain several keypoint matches for calibration by only comparing with the UKPs from the photo candidates. Therefore, we can quickly infer the extrinsic camera parameters (i.e., camera pose and device position) with multiple view geometry (MVG) technique [3]. As workloads in the offline and online phase are both reduced, our framework is more suitable for mobile scenarios.

Framework Overview

We start by introducing the data structure used in the framework. It consists of a database D and a constraint set C. Figure 2 shows a typical procedure of constructing D. Specifically, each record $d \in D$ has three fields: (1) a phone-captured photo p; (2) a set of consecutive Wi-Fi signals w; (3) device's gyro information g. Besides, the set C records some similarity and dissimilarity constraints between elements in D, where each entry $c \in C$ is a triple $(d_i^{(1)}, d_i^{(2)}, y_i)$. Particularly, $d_i^{(1)} \in D$, $d_i^{(2)} \in D$ and $y_i \in \{-1, +1\}$ determines if $d_i^{(1)}$ and $d_i^{(2)}$ are similar or not. Note that C has a small size (e.g., 40 or 60) in our system.

Offline Phase

As mentioned before, Wi-Fi and gyro are used to reduce the search space of candidate photos for calibration. However, it is not reasonable to directly compare the query with these feature vectors by a naive combination of cosine or euclidian similarities. Thus, we are motivated to learn a novel distance metric in order to bridge the gap between the query and the database. Next, we cluster photos into distinctive clusters and find out UKPs located in each photo cluster to make preparations for online query processing.

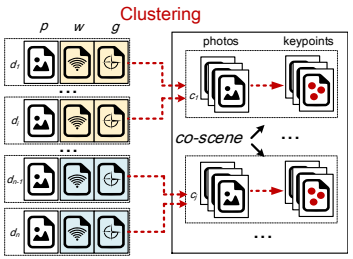

Figure 3: Co-scene Photos Clustering

Figure 4: Detected UKPs

Figure 5: A Few Sampled Photos

Co-scene Distance Learning. Assume a Wi-Fi signal vector $w \in \mathbb{R}^n$ is $w = (w_1, w_2, \ldots, w_n)$, a gyro feature $g \in \mathbb{R}^4$ is a quaternion [3] $g = (\alpha, \beta, \gamma, \omega)$. We create a synthetic feature vector ϑ by a linear combination:

$$\vartheta = (w, \lambda g) = (w_1, ..., w_n, \lambda\alpha, \lambda\beta, \lambda\gamma, \lambda\omega) \qquad (1)$$

Based on such synthetic features, we define the *co-scene distance* D_{cs} between two photos p_i and p_j by the definition of *Mahalanobis distance* as:

$$D_{cs}(p_i, p_j) = \sqrt{(\vartheta_i - \vartheta_j)^T M_\vartheta (\vartheta_i - \vartheta_j)} \qquad (2)$$

where the matrix M_ϑ is the metric parameter that we need to learn. One effective solution to optimizing such parameter is called ITML [1], which is to regularize M_ϑ to be as simple as possible (e.g., Identify matrix) while satisfying the similarity and dissimilarity constraints in the set C.

Co-scene Photos Clustering. After the metric is learned, we cluster the photos in our database into k groups. Figure 3 depicts the clustering process. Photos in each group form a co-scene. Note that the number k of groups is a hyperparameter that needs to be tuned by cross validation.

UKPs Detection. Camera calibration needs keypoints matched with the query photo as inputs. Although selected co-scenes can help us reduce photo candidates, the computational cost would still be expensive if we iterate through all keypoints located in one or several co-scenes. Thus, we can count the frequency of each keypoint by matching photos pairwisely in the same co-scene. Only those frequently appearing (matched) keypoints are useful for further calibration. Such keypoints are then called "useful keypoints" (UKPs). As shown in Figure 4, only a few UKPs (red) are detected from the whole local keypoints (blue).

Online Phase

The online query contains the same structure of a sampled record d in the offline phase. Now that we have already organized photos into co-scenes and detected UKPs, we are able to quickly pick out the keypoint pairs between the query and other photos by a two-level selection, i.e. *photo candidates selection* and *keypoint candidates selection*.

Photo Candidates Selection. Given a query, we first compose the synthetic feature ϑ_q according to Equation 1. It is easy to find the nearest co-scene by comparing ϑ_q with all cluster centroids based on the learned distance metric. As a result, all the photos in that co-scene are considered as the photo candidates.

Keypoint Candidates Selection. Next, we need to determine which keypoints in the photo candidates are matched with the query photo. Note that UKPs have already been detected from those photo candidates. Thus, we only need to compare keypoints in the query with these UKPs to confirm the ultimate matched keypoint pairs. Since the UKPs are only a small portion of all the keypoints, the online matching is accelerated. Finally, we use those few good matched keypoint pairs to estimate the query photo's camera pose as well as the position by a MVG process [3].

Experiments

Datasets. We develop an Android app to collect data, which records photos together with Wi-Fi signals and gyro information at the shooting time. We employ 4 volunteers to sample data in 3 downtown shopping malls and 1 office building in Hangzhou, China. It is worth noting that volunteers are required to indicate a few similar and dissimilar pairs in order to construct the constraint set C. A few sampled photos are shown in Figure 5 and the statistics of our dataset are summarized in Table 1. We also ask 2 other volunteers to collect a query set, 50 queries for each place.

Baseline Approach. As work [6] requires elaborative video recording, it cannot conduct calibration using our lightweight dataset. Thus, we implement a state-of-the-art

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

Table 1: Real Datasets

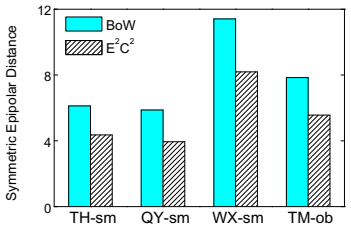

Figure 6: SED Comparison

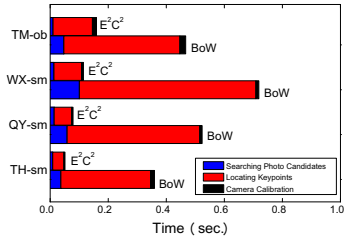

Figure 7: Running Time

bag-of-visual-words (BoW) approach [5] as the baseline. We cluster all the extracted SIFT points into a dictionary with 5K visual words, and then all photos in both the training set and query set are represented as a 5K-dimensional BoW feature vector. When a query photo is issued, we find the nearest photos and compare these photos with the query to locate the matched keypoint pairs.

Effectiveness of Our Framework. We adopt *Symmetric Epipolar Distance* (SED) [2] widely used in the field of camera calibration to evaluate our framework. In a nutshell, SED is an averaged error in the pixel space when estimating the camera calibration parameters. Lower values of SED indicate a better estimation. We plot the SED of BoW (green) and our framework E^2C^2 (shaded) in Figure 6 in 4 different datasets. Clearly, our approach outperforms BoW in each dataset. Thus, the detected UKPs are more robust and discriminative than a simple brute-force matching.

Efficiency of Our Framework. Generally, each query processing consists of three successive steps, i.e., searching photo candidates, locating keypoints and camera calibration. We plot the averaged running time for each dataset and also show the cost of each processing step in Figure 7. The following observations are drawn: (i) our framework outperforms the baseline with a significantly reduced online processing cost; (ii) we lower the cost of searching photo candidates since our search space is reduced to cluster centroids (each represents the photos in that cluster) instead of complete photos; (iii) the cost of searching keypoints is decreased remarkably due to our UKPs detection algorithm, which gets rid of unnecessary visual comparison between local features. The calibration cost does not make much difference after we have located the keypoint pairs.

Conclusion

In this work, we propose a novel framework E^2C^2 that aims at accelerating the camera calibration in an indoor setting. We use Wi-Fi and gyro sensor data to learn a useful metric for fast searching co-scene photos and locating UKPs. Experimental results show that our framework is effective to improve the calibration accuracy and meanwhile efficient in doing real-time calibration in mobile scenarios.

Acknowledgements

This work is supported by the National Program on Key Basic Research Project of China (Grant No. 2015CB352400) and the National High-tech R&D Program of China (Grant No. 2013AA040601).

REFERENCES

1. Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. 2007. Information-theoretic metric learning. In *ICML*. ACM, 209–216.
2. Mohammed E Fathy, Ashraf S Hussein, and Mohammed F Tolba. 2011. Fundamental matrix estimation: A study of error criteria. *Pattern Recognition Letters* (2011), 383–391.
3. Richard Hartley and Andrew Zisserman. 2003. *Multiple view geometry in computer vision*. Cambridge university press.
4. Pai Peng, Lidan Shou, Ke Chen, Gang Chen, and Sai Wu. 2013. The knowing camera: recognizing places-of-interest in smartphone photos. In *SIGIR*. ACM, 969–972.
5. Pai Peng, Lidan Shou, Ke Chen, Gang Chen, and Sai Wu. 2014. The knowing camera 2: recognizing and annotating places-of-interest in smartphone photos. In *SIGIR*. ACM, 707–716.
6. Antonio J Ruiz-Ruiz, PE Lopez-de Teruel, and O Canovas. 2012. A multisensor LBS using SIFT-based 3D models. In *IPIN*. IEEE, 1–10.