# 3D versus 2D based indoor image matching analysis on images from low cost mobile devices

Nabeel Khan
Department of Computer Science
University of Otago, New Zealand
nabeel@cs.otago.ac.nz

Brendan McCane
Department of Computer Science
University of Otago, New Zealand
mccane@cs.otago.ac.nz

Steven Mills
Department of Computer Science
University of Otago, New Zealand
steven@cs.otago.ac.nz

***Abstract*—Because of the increasing popularity of camera-equipped mobile devices, image matching techniques offer a potential solution for indoor localisation problems. However, image matching is challenging indoors because different indoor locations can look very similar. In this paper, we compare two image-based localisation approaches on realistic datasets that include images from cameras of varying quality. The first approach is based on 3D matching and the second on 2D matching. The comparison shows that 3D image matching crucially depends upon on the quality of the camera and its correct image matching accuracy ranges from 62-92% depending on the dataset. In contrast, the matching accuracy of 2D image matching is consistent across all cameras and ranges from 80-95%. In terms of computational efficiency, the 2D method is five times more efficient, but both methods are fast enough for many applications. We further investigate the performance of the 2D approach on four realistic indoor datasets with 50 indoor locations, such as corridors, halls, atrium or offices. Four out of five test sets have correct acceptance greater than 85% showing that image-based methods are viable for indoor localisation applications.**

## I. Introduction

The Global Positioning System (GPS) is the most commonly used system for determining outdoor location, but it is not reliable indoors and it is also hard to extract reliable elevation details. Indoor localization without GPS is still an open research problem and alternative techniques have become a focus of research during the past decade. However, existing indoor localization systems are still limited because most of them either require expensive infrastructure (infrared) [1], provide low accuracy (audible sound) [2] or produce large errors (inertial sensors) [3]. On the other hand, computer vision offers the potential for a good solution because of the emergence of cheap smartphones equipped with cameras. Computer vision solutions either use 2D training images or 3D training models to match a query image to determine an indoor location. The primary question relating to such use, which we answer affirmatively in this paper, is if such solutions can be accurate enough for indoor positioning applications.

### A. Related Work

In computer vision, the problem of image based localisation has been addressed by several authors in the past, mostly considering outdoor scenes. The most successful image matching approaches rely on wide baseline matching techniques based on features [4]. The basic idea behind these techniques is to extract features from images and perform feature matching often followed by image geometry based verification to compute the position of a query image with respect to a database of registered images or views [4]–[6]. To handle most image deformations, such as viewpoint changes or illumination differences, these methods use robust features like SIFT [7].

One of the earliest image localization frameworks for an urban environment has been presented by Robertson and Cipolla [4]. The database contains a set of 200 rectified views of building facades, which are manually registered by the authors onto a city map. During localization, the system performs feature matching to identify a nearby database view followed by the computation of the relative pose between the nearby database and query views from planar homographies. In contrast, Schindler *et al.* present an automatic recognition system based on a visual vocabulary tree for image matching on a city-scale database of 30000 GPS annotated images [6]. The system identifies the most informative training features and uses them to develop a visual vocabulary for efficient retrieval of relevant images against a query image during localization.

Using a 3D model to represent a location offers the additional advantage that the full camera pose can be determined. For high localization accuracy, use of a 3D map of the environment is needed [8]–[11]. The basic idea is to compare 2D features of a query image with 3D points to determine 2D to 3D correspondences followed by pose estimation during localization. The 3D model giving a successful pose against a query image is picked as the best match and indicates the corresponding location.

Irchastra *et al.* generate a set of synthetic camera views from 3D models [8]. Original and synthetic views are then combined to form a database, which is then compressed by reducing the number of views based on spatial information. During localization, the system estimates the pose of a query image from compressed data of city scale environments and provides real time localization performance. The main problem with such approaches is that the search time for 2D to 3D correspondences increases with the size of 3D data, which decreases the overall efficiency. To address this problem, several schemes have been proposed to first reduce the size of 3D data and then use a smaller set of 3D points to identify 2D to 3D correspondences for pose estimation efficiently [9],

[10]. In these systems, the priority of a 3D point is directly related to the number of registered images in which a 3D point is seen.

The underlying problem with 3D based image localization is that a rich 3D model needs to be generated. Otherwise, it will result in fewer 2D to 3D correspondences and a poor pose. On the other hand, 2D based approaches require relatively many correct correspondences to generate a hypothesis. Klas *et al.* proposed to use a hybrid set of feature correspondences including both 2D to 2D and 2D to 3D correspondences to estimate the pose of a query image during localization [12]. Most localization works have used high quality query images for localization in urban environments. However, Arth *et al.* use a smartphone having a 3.2 MP camera to capture indoor query images for 3D based indoor localization and obtain correct image matching accuracy up to 88% in several rooms.

In this paper, we evaluate 2D and 3D image matching approaches. Our 3D approach differs from other works because it incorporates both feature based matching and pose estimation techniques while other 3D based works mostly perform only pose estimation. On the other hand, our 2D approach is different from feature based approaches because it uses a layered architecture with a voting module, verification method and a post-verification step. Such a layered approach is useful for correct image matching in large self-similar indoor environments. To the best of our knowledge, there is no previous work that evaluates and compares 3D localization with 2D localization across indoor query images captured from different mobile device cameras. The contributions of this paper are two-fold. We compare 2D and 3D indoor localization approaches across different cameras and we develop several indoor datasets for evaluation.

The paper is structured as follows. Section II discusses the 3D based localization approach. Section III discusses 2D based localization approach followed by datasets and performance metrics in Section IV. The experimental results are presented in Section V and the article is finally concluded in Section VI.

## II. 3D based Image Localization

A scene reconstruction system processes a number of input images to generate a 3D model of a location. The accuracy of a reconstructed scene is proportional to the number of input images successfully registered with a 3D model. A 3D model of a location is a dense cloud of 3D points obtained from registered images. Each 3D point has a list of:

- corresponding 2D features from registered images, which are used to triangulate that 3D point.
- registered images in which the 3D point is detected, also known as "visibility".

Recent advancements in structure from motion (SfM) has made it possible to construct 3D models of large scale environments effectively [8]–[10]. In 3D based image localization, pose is estimated between a query image and pre-computed 3D models. A pose indicates the position of the camera and the number of correspondences between features of a query image and the 3D points of the 3D model, which have contributed to its computation, often called inliers. The 3D model giving a reliable pose (i.e. maximum number of inliers) against a query image is picked as the best match and indicates the indoor location. We will refer to 2D features of images simply as features and 3D points of 3D models simply as points in the remainder of this paper.

### A. 3D model generation

We have used the Bundler system [13] to generate 3D models of indoor locations. We use the idea of [10] and represent each point of the 3D model by averaging its corresponding features, which results in compact 3D point representations. Compact representations have been shown to improve image matching accuracy and efficiency during localization [8].

After building the 3D models, different inclusion criteria can be used to select a subset of points from these 3D models for matching with features of query images. Our criteria of point selection is based on the visibility of points from multiple images in the image set:

1) **All Visible (A)**:- All points from 3D models are picked.
2) **Less Visible (L)**:- Only those points which are seen in exactly two images are picked. These points should be highly discriminating.
3) **Highly Visible (H)**:- Only those points which are seen in more than two images are picked. These points should allow matching from multiple camera positions.

### B. 3D based image matching approach (3DM)

The proposed *3DM* approach consists of two subsystems: a naive matcher (*NM*) and a pose estimator (*PE*). During data acquisition, indoor 3D models with all visible (A), less visible (L) or highly visible (H) points are generated. Features are also extracted from registered images and are stored along with the 3D models. Registered images and 3D models are both annotated with location information.

*1) Naive Matcher (NM):* This module uses a 2D image matching approach. It extracts features from a query image and compares them with features of each registered image. Two compared features are considered similar if the Euclidean distance between them is less than 170 [14], an empirically determined similarity measure to ensure many correct feature matches. The registered image with the most feature matches is picked as the best match. This is one of the simplest 2D matching schemes (hence naive), but it tends to generate the highest number of correct matches.

*2) Pose estimator (PE) :* The problem with *NM* is that it can produce incorrect images matches. To minimize localization errors, the *PE* module validates the potential match from *NM*. The *PE* module selects the indoor 3D model based on the location information produced by *NM*. It compares features of the query image with points of the selected 3D model to determine 2D to 3D correspondences. To find a correspondence, we determine the two closest points from the 3D models using the Euclidean SIFT distance (this is a distance in feature space, not in 3D space). The correspondence of a query feature is accepted if its two best matched points pass the ratio test of 0.7 [10]. This ratio is determined empirically and ensures many correct 2D to 3D correspondences between a query image and a 3D model. Correct correspondences need to be picked to generate a reliable pose, but it is unreasonable to expect all 2D to 3D correspondences to be correct. To address this, *PE* uses the best 50 correspondences only and applies three different pose estimation algorithms: ITR [15], P3P [16] and EPnP [17] one by one. The reason for using more than one algorithm is to increase the probability of getting a reliable pose i.e. large number of inliers, which indicates a high level of consistency between features and points.

*PE* then selects the best pose (with maximum inliers) from each of the pose estimation algorithms and uses it to re-project those points of the 3D model, which have matched with query features. With a reliable pose, most re-projected points should overlap approximately with the corresponding query features. If the number of matching re-projected points is greater than 7 (an empirically determined value), the 3D model is considered a match. Otherwise, the match is not found and a "no-location" message is returned. The *PE* module tends to reject query images unless it is very confident in the match. Therefore we expect the correct match rate to be lower but the incorrect match rate to be almost zero.

## III. 2D BASED IMAGE MATCHING APPROACH (2DM)

2DM uses a visual Bag of Words based algorithm to match indoor query images against a database of annotated indoor images. Before the system can be used, the image database and index must be built.

### *A. Data acquisition and indexing*

Firstly, the user captures a number of images of indoor buildings (training images) from a digital camera and stores them along with location annotations. *2DM* then automatically generates the following data:

- SIFT features are extracted from training images and stored along with location annotations.
- Clustering is performed on features of training images with 30000 cluster centers. The obtained cluster centers are called visual words and a set of all visual words is called a visual vocabulary. The features of the training images are mapped to the closest visual words to obtain a visual word distribution for each training image. A data structure called an inverted index is then created which records visual word distributions for every training image. The inverted index helps for quick retrieval of relevant images against a query image and is a standard document retrieval data structure. *2DM* stores the visual vocabulary and the inverted index for later use.
- Every indoor location is represented with a reduced set of features (LTF features from [18]) and are stored along with location annotations.

### *B. Online localization*

To localize a query photo, *2DM* first uses the inverted index to quickly retrieve the most similar 200 training images. The distribution of visual words is then computed for the query image and is compared with the distributions of retrieved images to generate a ranked list of database images. The top fifty ranked images are then used to decide upon the final location using the following three modules:

**Voting scheme:** If the first three ranked images refer to the same location, the voting scheme simply returns that location. Otherwise, it means that the voting scheme cannot make a consistent decision and it calls the verification method.

**Verification method:** Image geometry information, such as the fundamental matrix [15] or planar homography [19], is used to match a query image with the first 50 ranked images one by one. The verification method stops once it matches a ranked image with a query image. It does not compare the remaining ranked images with the query image and returns the location of the matched ranked image. The fundamental matrix uses a full structure match between a query and a ranked image whereas planar homography considers planar matches only. The fundamental matrix computation is more physically plausible than homography but also more costly. There are three possible outputs from the verification method:

- It fails to find a match in all ranked images, which means that the location cannot be determined i.e. "no-location" message.
- It finds a match within the first 10 ranked images, then it simply returns the location of the best matched ranked image.
- It finds a match outside the first 10 ranked images, which means that the match is not confident. It then calls the post-verification method to validate the location output.

Details of the homography and fundamental matrix methods are as follows:

- *Homography:* Features in one image are related to corresponding matched features in another image by a homography. This verification method picks the 10 best feature correspondences based on Euclidean distances between feature descriptors. Potential homographies are then computed via Random Sample Consensus (RANSAC). For a computed homography, features of a ranked image are transformed to new locations (spatial coordinates) based on the homography. If at least seven transformed features of a ranked image come approximately at the same location as features of a query image and are also similar to each other (feature distance is less than 150) then a perceptive correspondence is recorded between query and ranked images. The ranked image giving at least three perspective correspondences against a query image is considered a match.
- *Fundamental matrix:* The fundamental matrix encapsulates the intrinsic projective geometry between images. If $\mathbf{x}$ and $\mathbf{x}'$ are matched features in two images, the fundamental matrix, $F$, relates the matched features by the following equation:

$$\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0 \tag{1}$$

The matched features between two images are used to compute a fundamental matrix and corresponding inliers. Inlier points are those points which match according to the fundamental matrix equation:

$$\mathbf{x_i}'^T\mathbf{F}\mathbf{x_i} < \epsilon, i = 0, 1, ..n \tag{2}$$

where $x'_i$ and $x_i$ are matched features in two images, and $\epsilon$ is an empirically determined threshold. A larger number of inliers indicates a reliable match between two images. This verification method uses the top 20% best feature correspondences to compute the fundamental matrix using the 8-point algorithm with RANSAC [15]. If at least 20% inliers are found, then the location of that image is returned.

**Post-verification method:** This method computes feature correspondences between features of every indoor location with the features of a query image. The indoor location giving maximum feature matches is picked as the best match for a query image. If the indoor location matches with the indoor location determined by the verification method, then the location output is considered authentic and is returned. Otherwise, the post-verification method returns "no-location".

## IV. Datasets and performance metrics

### A. 3D models

We developed 3D models of four places in the Computer Science building, which is a standard office building. We used four indoor locations because the initial plan was to compare the performances of *3DM* with *2DM* on a small scale. The statistics for all 3D models are shown in Table I, which indicate that fewer features are obtained for corridors due to the lack of textures. The less visible points represent about 72% of the overall points while the remaining are highly visible.

TABLE I: Statistics of indoor 3D Models

| 3D Model | Images | #Registered Cameras | # of 3D points |
|---|---|---|---|
| Coffee Room | 60 | 56 | 58985 |
| Graphics Corridor | 15 | 14 | 24948 |
| Graphics Library | 63 | 56 | 68541 |
| Office Corridor | 60 | 60 | 12825 |
| **Total** | **198** | **186** | **165209** |
| **Less Visible** | | | **118950** |
| **Highly Visible** | | | **46259** |

### B. 2D trained data

Table I shows that a total of 186 images out of 198 are registered with the four 3D models. We selected all registered images for the 2D training dataset, and a total of 53848 training features were extracted. All registered images were captured from a Nikon D3000 DSLR Camera.

### C. Query images

We captured 60 query images for each indoor location from seven different mobile devices. The query images were captured 2 months after capturing the images used for the 3D model reconstruction. Therefore, some degree of changes can be expected between registered and corresponding query images as shown in Figure 1.

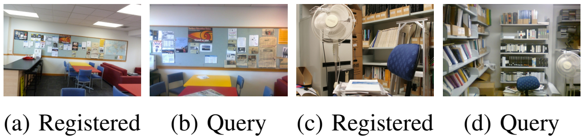
(a) Registered (b) Query (c) Registered (d) Query

Fig. 1: Sample registered and corresponding query images.

### D. Mobile devices

We used relatively cheap mobile devices in experiments rather than more expensive ones like the iPhone5, Samsung Galaxy S3 etc. The specifications of the used mobile devices are stated in Table II.

TABLE II: Mobile device specifications.

| | Operating system | Type | Camera | Release date |
|---|---|---|---|---|
| **Nokia N95** | Symbian | Phone | 5 MP | 2008 |
| **iPhone 3Gs** | iOS 6.0 | Phone | 3.0 MP | 2009 |
| **Galaxy I7500** | Android 1.5 | Phone | 5 MP | 2009 |
| **Nokia C3** | Symbian | Phone | 2 MP | 2010 |
| **IDEOS U8150** | Android 2.2 | Phone | 3.15 MP | 2010 |
| **Slim S7** | Android 2.2 | Tablet | 3.15 MP | 2011 |
| **HTC Wildfire S** | Android 2.3 | Phone | 5 MP | 2011 |

### E. Realistic indoor datasets

A larger, more representative indoor dataset was also created. The training images were captured by a Sony Cybershot camera (8MP) and all query images were acquired with the camera of a HTC Wildfire S (5MP). The query images were captured several weeks after capturing the training images. For every query location, there are multiple training images of the same location. The datasets are as follows:

1) **Owheo (OW):** Contains images of an office building at the University of Otago, New Zealand. It contains 1534 images for training and 750 images for testing. The dataset covers 25 indoor locations. The test images of OW dataset contains images taken during day and night times. We split the OW into two datasets based on test data:
    a) **Owheo-D**: Contains 537 day-time test images.
    b) **Owheo-N**: Contains 213 night-time test images.

    The number of training images are kept same for the above two datasets i.e. 1534.
2) **Commerce (CM):** Contains images of an office building at the University of Otago, New Zealand. It contains 864 images for training and 234 images for testing. The dataset covers 14 indoor locations.
3) **Otago Museum (OM):** Contains images of Otago Museum, New Zealand. It contains 1045 images for training and 135 images for testing. The dataset covers 7 indoor locations.
4) **Dunedin Stadium (DS):** Contains images from the Forsyth Barr Stadium, New Zealand. It contains 455

images for training and 75 images for testing. The dataset covers 4 places within the building.

### F. Performance metrics

The following definitions are used to define the performance metrics for *2DM* and *3DM*:

$Q_t$ Total number of query images.
$N_c$ No. of query images correctly matched.
$N_d$ No. of query images for which no decision is made.

The following evaluation metrics are used in this work:

$C_a$ Refers to the correct acceptance rate.

$$C_a = N_c/Q_t \tag{3}$$

Higher is better for this metric.

$W_m$ Refers to the wrong match rate

$$W_m = 1.0 - (N_c + N_d)/Q_t \tag{4}$$

Lower is better for this metric.

The values of above metrics ranges from 0-1.

## V. Experimental Results

### A. 3DM analysis

A query image is considered correctly matched if the corresponding matched training image belongs to the query location. We computed the correct acceptance rate (C$_a$) for *3DM* with 3D models as shown in Figure 2. The results show that *3DM* produces very low C$_a$ for mobile devices which have 2-3MP camera. The C$_a$ improves for the three 5MP cameras. However, the corresponding wrong match rate (W$_m$) is zero for *3DM* against all mobile devices. The results indicate that the C$_a$ for *3DM* improves as the resolution of the camera increases.

Figure 2 also shows that there is nothing much to differentiate between the performance of all visible (A) and highly visible (H) points. *3DM* is found to perform best with less visible (L) points and takes about 5 seconds on average to match a query image. We suspect that highly visible points are seen in almost every registered image due to similar texture or pattern followed in the office building, which may decrease the matching performance.

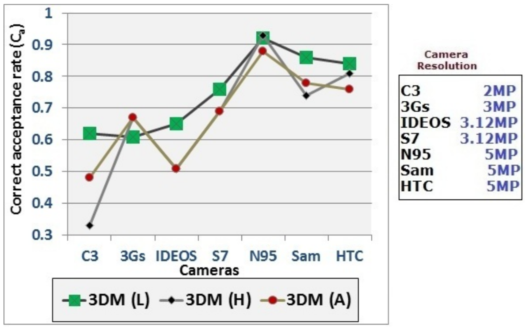

Fig. 2: The C$_a$ for *3DM* approach.

We initially suspected that the *NM* module was producing a number of wrong image matches and was the reason for the lower C$_a$ of *3DM*. To check this, we computed the C$_a$ for the *NM* module alone as stated in Table III, which shows that the *NM* module provides excellent C$_a$ and determines the correct location most of the time. This indicates that in failure cases, *PE* cannot estimate the pose between the query image and the selected 3D model despite the match being correct. There are two reasons for this: the 3D models are somewhat limited because it is difficult to construct them in some locations (especially corridors) due to the lack of discriminating features; poor quality query images produce poor 2D to 3D correspondences leading to poor pose as shown in Figure 2 and is the cause of significant error for low-resolution cameras.

TABLE III: The C$_a$ for the *NM* module.

| | C3 | 3Gs | IDEOS | S7 | N95 | Sam | HTC |
|---|---|---|---|---|---|---|---|
| C$_a$ | 0.97 | 0.93 | 0.93 | 0.97 | 0.98 | 0.97 | 0.95 |

### B. 2DM analysis

We have tested three variants of *2DM*:

1) **2DM(H):** With homography verification method.
2) **2DM(F):** With fundamental matrix verification method.
3) **2DM(base):** With no layered architecture. It works similar to standard visual BoW in which the first top ranked image is considered the best match. This is used as the baseline method for comparison with other variants.

We computed the correct acceptance rate (C$_a$) for *2DM* against query images of all mobile devices as shown in Figure 3. The results show that *2DM* produces higher C$_a$ than *3DM* across queries of all mobile devices, because it rejects fewer query images than *3DM*. As a consequence it suffers from some wrong matches (W$_m$) as shown in Table IV. This is expected because it is very hard to reject all wrong image matches in 2D image matching.

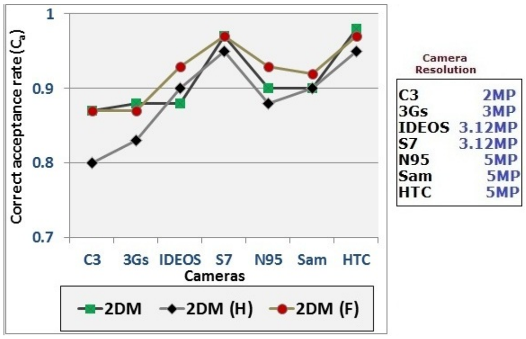

Fig. 3: The C$_a$ for *2DM* variants.

TABLE IV: The W$_m$ for the *2DM*.

| | C3 | 3Gs | IDEOS | S7 | N95 | Sam | HTC |
|---|---|---|---|---|---|---|---|
| 2DM(base) | 0.13 | 0.12 | 0.12 | 0.03 | 0.1 | 0.1 | 0.02 |
| 2DM(F) | 0.13 | 0.07 | 0.05 | 0.02 | 0.07 | 0.08 | 0.02 |
| 2DM(H) | 0.05 | 0.05 | 0 | 0.02 | 0.03 | 0.03 | 0.02 |

All variants take about 1 second to find an image match and perform comparably to the base method i.e. *2DM(base)*, but with fewer wrong image matches. The reason is that *2DM(base)* selects the first top ranked image as the best match without performing any geometric verification and hence gives more wrong matches. *2DM(H)* performs best because it offers a lower $\mathrm{W}_m$ and reasonable $\mathrm{C}_a$ compared to other variants.

*2DM* has performed better than *3DM* for low resolution cameras such as 2-3MP. *2DM* is efficient and provides a high correct acceptance rate but at the cost of slightly higher wrong match rates compared to *3DM*. *2DM* offers a reasonable trade-off between image matching performance and efficiency, and therefore seems to be a better choice for image matching especially for smartphones having low resolution cameras.

For more analysis, we further evaluate the performance of *2DM* on large realistic indoor datasets. In these datasets, all training images are treated as a single training collection and the performance of *2DM* variants for image matching are shown in Figure 4. The results show that *2DM(H)* provides low $\mathrm{W}_m$ and its $\mathrm{C}_a$ is also comparable to other variants. *2DM(F)* also does better than *2DM (base)* but its $\mathrm{W}_m$ is slightly higher than *2DM(H)*. *2DM(H)* offers 60-93% correct image matching performance, while its wrong image match rate varies from 3-11%. All variants perform poorly with the Owheo-N dataset because query images were taken at night. The OW building has many glass windows, which significantly changes the scene at night due to glass reflections and results in poor performance. One possible solution to tackle this problem may be to capture training images at night as well.

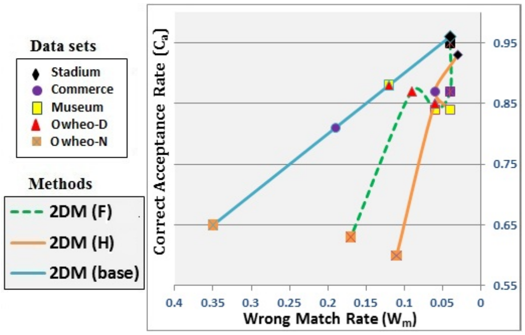

Fig. 4: The results for *2DM* variants. The curves that are higher and further to the right indicate a better performance.

### C. Runtime performance

*3DM* takes about 5 seconds to match a query image. On the other hand, *2DM(H)* and *2DM(F)* both are quite efficient and take about 1 second. The longer match time of *3DM* is due to its *PE* module, which performs an expensive operation and alone takes about 2 seconds on average. On realistic indoor datasets, *2DM(F)* and *2DM(H)* both take less than 2 seconds to match a query image on average.

## VI. Conclusion

In this paper, we show that it is possible to achieve indoor localization using a single image from off-the-shelf camera phones with 2D or 3D matching, which offers a cheap solution compared to other technologies, such as infrared or inertial sensors. We first propose and compare *3DM* with *2DM* across query images from different cameras and our evaluations show that 3D matching does not suit low resolution cameras, such as 1-3 MP. However, its performance improves with the 5MP camera. On the other hand, 2D image matching is efficient and correctly matches many query images but at the expense of some wrong image matches.

The results show that 3DM works reasonably well for high resolution cameras and therefore is probably a good option for the future as new smartphones have higher resolution cameras. On the other hand, *2DM* seems to be a better choice for our current indoor positioning system due to it efficiency and good matching performance across all resolution cameras.

## References

[1] NorthernDigital, "Optotrak," http://www.ndigital.com/industrial/optotrak.php, 2008.

[2] A. Mandal, C. V. Lopes, T. Givargis, A. Haghighat, R. Jurdak, and P. Baldi, "Beep: 3D indoor positioning using audible sound," in *IEEE Consumer Communications and Networking Conference*, vol. 10, no. 2, 2005, pp. 38–50.

[3] P. Robertson, M. Angermann, and B. Krach, "Simultaneous localization and mapping for pedestrians using only footmounted inertial sensors," in *International Conference on Ubiquitous Computing*, 2009, pp. 93–96.

[4] D. Robertson and R. Cipolla, "An image-based system for urban navigation," in *British Machine Vision Conference*, 2004, pp. 819–828.

[5] W. Zhang and J. Kosecka, "Image based localization in urban environments," in *Proceedings of International Symposium on 3D Data Processing, Visualization, and Transmission*, 2006, pp. 33–40.

[6] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *IEEE Conference on In Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–7.

[7] D. G. Lowe, "Distinctive image features from scale invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[8] A. Irschara, C. Zach, J.-m. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *CVPR*, 2009, pp. 2599–2606.

[9] Y. Li, N. Snavely, and D. P. Huttenlocher, "Location recognition using prioritized feature matching," in *11th European conference on Computer vision: Part II*, Heidelberg: Springer-Verlag, 2010, pp. 791–804.

[10] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2D-to-3D matching," in *International Conference on Computer Vision (ICCV)*,2011, pp. 667–674.

[11] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg, "Wide area localization on mobile phones," in *IEEE International Symposium on Mixed and Augmented Reality*, 2009, pp. 73–82.

[12] K. Josephson, M. Byrod, F. K. Kahl, and K. Astrom, "Image-based localization using hybrid feature correspondences," in *CVPR*, 2007, pp. 1–8.

[13] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3D," *ACM Transactions of Graphics*, vol. 25, no. 3, pp. 835–846, July 2006.

[14] N. Khan, B. McCane, and G. Wyvill, "SIFT and SURF performance evaluation against various image deformations on benchmark dataset." in *Digital Image Computing Techniques and Applications*, 2011, pp. 501–506.

[15] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, April 2004.

[16] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Transactions Pattern Analaysis Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[17] F. Moreno-Noguer, V. Lepetit, and P. Fua, "Accurate non-iterative O(N) solution to the PnP problem," in *ICCV*, 2007.

[18] N. Khan, B. McCane, and S. Mills, "Feature set reduction for image matching in large scale environments." in *International Conference on Image and Vision Computing (IVCNZ)*, 2012, pp. 68–72.

[19] N. Khan, B. McCane, and G. Wyvill, "Homography based visual bag of word model for scene matching in indoor environments." in *IVCNZ*, 2011.