Article

Indoor positioning using PnP problem on mobile phone images

Hana Kubíčková [1], Karel Jedlička [2,*], Radek Fiala [2] and Daniela Beran [2]

[1] Plan4all; Horní Bříza; Czech Republic; hana.kubickova@plan4all.eu
[2] University of West Bohemia, Plzeň, The Czech Republic; jedlicka.krl@gmail.com; fialar@kgm.zcu.cz, dberan@kgm.zcu.cz
* Correspondence: jedlicka.krl@gmail.com; Tel.: +420-377-63-9210 (K.J.)

Received: date; Accepted: date; Published: date

Abstract: As people grow a custom to effortless outdoor navigation there is a rising demand for similar possibility indoors as well. Unfortunately, indoor localization, being one of the necessary requirements for navigation, continues to be problem without a clear solution. In this article we are proposing a method for an indoor positioning system using a single image. This is made possible using small preprocessed database of images with known control points as the only preprocessing needed. Using feature detection with SIFT algorithm we can look through the database and find image which is the most similar to the image taken by user. Pair of images is then used to find coordinates of database image using PnP problem. Furthermore, projection and essential matrices are determined allowing for the user image localization ~ determining the position of the user in indoor environment. Benefits of this approach lies in the single image being the only input from user and no requirements for new onsite infrastructure and thus enables a simpler realization for the building management.

Keywords: indoor positioning system; image-based positioning system; computer vision; SIFT; feature detection; feature description; cell phone camera; PnP problem; projection matrix; epipolar geometry; OpenCV

1. Introduction

Nowadays, determination of the location of many modern electronic devices is possibledesirable, especially the one we are accustomed to using every day - a mobile phone. Due to faster technological development of mobile phones and positioning services, mobile phones provided with GNSS that reliably works in outdoor environments became a matter of course.

Over time, there was an increasing need to locate these devices even inside buildings with the main goal of facilitating operations such as placing patients in a hospital, searching clerks in large office buildings or quick orientation in large shopping centers. One of the important requirements for indoor positioning is higher accuracy in contrast to outdoor use. If the positioning errors exceeded several meters, the user would not be able distinguish for example floor or room between each other and the service would provide information about distant places from actual position. However, not only higher user location accuracy is important for efficient indoor navigation, but also the simplicity of its determination. This is related to low acquisition costs, minimal maintenance, low maintenance costs and minimal use of new infrastructure. Taking these requirements into account, then we should omit techniques with enhanced infrastructure such as WiFi beacons, Bluetooth beacons, UWB or RFID, as the acquisition and maintenance costs are for large interior environments higher. The use of these technologies could be advantageous in warehouses and small enclosures areas where infrastructure can be relatively cheaply upgraded, including special hardware and client software, in exchange for higher accuracy. For individual navigation in free accessible interiors such as hospitals,

Peer-reviewed version available at ISPRS Int. J. Geo-Inf. 2020, 9, 368; doi:10.3390/ijgi9060368

airports or shopping centers cannot be counted with a client other than a regular mobile phone. In connection with the use of mobile phone, we therefore focus on techniques that determine the location with any other infrastructure needed. However, GNSS and inertial navigation itself do not give good results and they are therefore often complemented by other location determination techniques, which increases their cost and hardware and software requirements.

The latest technique, which does not use additional infrastructure for indoor positioning is image-based positioning technology. This technique uses a camera that is nowadays equipped with every mobile phone. The image-based positioning is thus characterized by low acquisition price and, in addition, according to the mentioned sources provides very good results in the determination user location. For this reason, we have found the image-based positioning technique for positioning via mobile phone very efficient and we decided to focus on the principles on which it occurs to calculate the position from single image, and the computer vision algorithms that are narrowly connected with this issue.

1.1. Recent works

Image based positioning is very comprehensive topic and there have also been several previous attempts at indoor image-based positioning. One of the oldest approaches to image-based positioning is exploitation of QR codes, which is in the context of positioning a data source containing a navigation map or text information regarding position. Replacing QR codes with images, we are talking about a method that is analogous to the so-called fingerprint method. (Namiot, 2015). This method is based on sending captured images to a web server and comparing them against the image database mapping the interior of the building. Ravi et al. (2006) claims that accuracy in receiving position by this approach is 1 m with more than 80% probability.

Hile et al. (2008) introduced one of the first image-based approaches using the mobile phone camera for positioning in corridors. Due to the presence of many repeating elements (corners, floor and wall transitions, doors) there have been located many natural tags. Instead of searching tags directly, authors used the image segmentation method. Their approach further consisted of finding those natural tags in the corridor floor plan database, where all important edges were stored and the subsequent calculation of the user's position was done via obtained feature correspondences between the captured image and the database. Based on the proposed procedure they reached a positioning error of around 0.30 m. Most works dealing with the issue of cell phone positioning based on database image retrieval use a SIFT algorithm. SIFT is a feature detection algorithm in computer vision to detect and describe local features in image, which is resistant to scaling, noise, and lighting conditions. (Lowe, 1999) An example of such a work is described in Liang et al. (2015). Authors tried to achieve an accuracy of less than 1 m using images taken with a mobile phone. Using SIFT features of images, the required positioning accuracy was achieved in more than 55% of taken images. A similar approach was used by Werner et al. (2011) with the difference of using SURF algorithm to find feature correspondences between images. Kawaji et al. (2010) discuss navigating in a museum environment via omnidirectional panoramic images taken at an interval of 2 m forming an image map. The main goal was to find the user's position with the highest accuracy in the shortest possible time. To solve this problem the authors used for feature detection PCA-SIFT algorithm. Based on the number of extracted features, best corresponding image was selected from the database and assigned to the user position (fingerprint method). The results of the study showed that the above procedure can be done estimating the position in 2.2 seconds with 90% accuracy.

Sadeghi et al. (2015) designed the OCRAPOSE positioning system, which is also based on feature recognition in the image and subsequent comparison of the newly acquired image to images stored in the database. Their approach to determine the resulting position differs from the above approaches. The location of the projection center of the camera is calculated through the PnP problem. The authors placed tables with text or numeric information into the rooms, which allowed them to use the advantages of Optical Character Recognition (OCR) method. Correspondence is thus between images through numeric or textual characteristics searched through SIFT more precisely. Coordinates of table corners have been used as input values for calculation of PnP problem. Research has shown

a mean positioning error of less than 0.50 m. A similar approach was taken by Deretey et al. (2015). In their method, they use one calibrated monocular camera whose position is independent on the camera position knowledge from the previous calculation. The difference from the previous solution is that newly taken images are compared to a pre-created 3D model of indoor environment providing 3D coordinates to calculate the EPnP problem.

Another approach based on image recognition was chosen by Van Opdenbosh et al. (2014). To reach maximum 1 m deviation, 16 images from different viewpoints were taken in each 1x1 m grid. However, this strategy does not provide good application in large indoor environment due to high number image dataset, which leads to high computing cost and increased memory consumption for mobile phones and web servers.

2. Materials and Methods

While designing our solution, we focused on two important requirements - the use of mobile phone camera and automation of the whole process. This in practice means, that a user should be able to determine his/her location from a single image taken by a mobile phone camera without need for further action - for example, marking the ground control points in the image. The main advantage of our solution lies in using a significantly smaller amount of images, comparing to the approaches mentioned above. The following text of Materials and Methods chapter refers to photogrammetric and computer vision fundament of our solution. As is widely known, the basic model forming the image in the camera is a perspective projection describing image structure using the so-called pinhole camera model. Although commonly used camera lenses are trying to bring perspective projection as close as possible, the real design differs substantially from this idealization. The position and attitude of camera when taking the image is determined by elements of exterior orientation. The (geometrical) properties of the camera by elements of interior orientation. All these elements are concealed in so-called projection matrix.

2.1 Projection matrix

Having 3D world coordinates selected as homogenous $\boldsymbol{X_i} = [x_i, y_i, z_i, 1]^T$ it is possible to introduce a projection matrix P, which can be expressed as follows:

$$\boldsymbol{P} = \boldsymbol{K}\,[\boldsymbol{R}|\boldsymbol{t}] = \boldsymbol{K}\boldsymbol{R}[\boldsymbol{I}, -\boldsymbol{C}]$$

The calibration matrix $\boldsymbol{K}$ contains elements of internal orientation and matrix $\boldsymbol{R}|\boldsymbol{t}$ describes the movement of the camera around a static scene or moving object in front of the static camera. Matrix $\boldsymbol{I}$ is a unit matrix and $\boldsymbol{C}$ indicates the position of the camera projection center. The position of the camera projection center can be considered as the position of person taking image.

The projection matrix can be estimated in two ways – either through a known scene or through an unknown scene.

In the known scene, 3D object coordinates and their corresponding 2D coordinates in the image are available. This is also known as PnP problem, when at least 6 tuples of 3D object coordinates and 2D image coordinates of its image must be available to estimate the projection matrix and derive the position of the camera projection center $\boldsymbol{C}$.

Position determination of a single image without knowing the 3D object coordinates and 2D image coordinates of ground control points is not possible. Therefore, in the simplest case of single image positioning, there is a need of image database containing location information, which will stand next to the user-taken image.

After finding the best matching image from the database, the position of the database image can be assigned to the user. However, if we wanted to assign this location to the user based on similarity of the input image and database image, the database should by composed of a large number of images, that would have a huge impact on the computational complexity of the proposed solution. Moreover, as in the case of Van Opdenbosh et al. (2014), location accuracy would depend on the size of image network forming the database. Therefore, to achieve the highest possible accuracy with low number of images, we must look at the positioning problem from a different perspective. After

receiving a pair of matching images, we can calculate the user's location through partly unknown scene.

In a case of unknown scene, at least two images with known correspondences of image points are needed to estimate the projection matrix. Using two images and its correspondences of image points for the projection matrix estimation, we are talking about epipolar geometry.

The epipolar geometry is the intrinsic projective geometry between two views. It is independent of scene structure, and only depends on the cameras' internal parameters and relative pose. The epipolar geometry of two cameras is usually motivated by considering the search for corresponding points in stereo matching.

Suppose a 3D point X is imaged in two views, at point x_1 in the first image, and x_2 in the second. Two cameras are indicated by their projection centers C_1 and C_2 and image planes. The camera centers, 3D point X, and its images x_1 and x_2 lie in a common plane π. The line through C_1 and C_2 intersects each image plane at the epipoles e_1 and e_2. Any plane π containing the projection centers is an epipolar plane, and intersects the image planes in corresponding epipolar lines l_1 and l_2.

2.2 Fundamental matrix

The algebraic representation of epipolar geometry is the so-called fundamental matrix F of the size 3 × 3 with the rank 2. Fundamental matrix describes the translation of the point x_1 from the first image to the second image through the epipolar line l_1:

$$l_2 = \boldsymbol{F x_1}$$

The fundamental matrix estimation can be approached in two different ways, either by knowing the projection matrices of cameras P_1 and P_2 or by obtaining the point correspondence x_1 and x_2. For the second case, the methods for fundamental matrix estimation are divided according to the number of point correspondences obtained between image planes. These exist 7-point algorithm and 8-point algorithm.

2.3. Essential matrix

The essential matrix is the specialization of the fundamental matrix to the case of normalized image coordinates $\widehat{x_1}$ and $\widehat{x_2}$, where $\widehat{\boldsymbol{x_1}} = \boldsymbol{K_1}^{-1}\boldsymbol{x_1}$. Thus, the relationship between corresponding normalized image coordinates x_1 and x_2 is very similar to the fundamental matrix: $\widehat{\boldsymbol{x_2}}^T \boldsymbol{E} \widehat{\boldsymbol{x_1}} = 0$.

The relationship between the essential matrix and the fundamental matrix can be expressed as:

$$\boldsymbol{E} = \boldsymbol{K_2}^T \boldsymbol{F}\, \boldsymbol{K_1}$$

Projection matrices of cameras that capture the same scene from different angles can be estimated knowing their relative position - translation vector t and rotation matrix $\boldsymbol{R}$. Both information is contained in the essential matrix $\boldsymbol{E}$. The usual way to separate the translation and rotation is the SVD decomposition. (Hartley & Mundy, 1993), (Hartley & Zisserman, 2003)

2.4. Feature Correspondence detection algorithms

When calculating the fundamental and essential matrix, the *feature correspondences* must be found. In order to search feature correspondences between two images, it is necessary to search for *features* in the image first. There is no universal or exact definition of what constitutes a *feature* – features may be specific structures in the image such as points, edges or regions of points. The *feature* could be thus defined as an "interesting" area in the image that is sufficiently distinguishable from its surroundings. Features can be divided into two categories, depending on their origin and the detection method. The first category includes marker-less *natural features*, that naturally occur in the scene. The other category, *synthetic features* (eg. brightly colored geometric shapes), appear in the scene due to human intervention (like a high reflective objects added to the interior). (Hassabalah et al., 2016), (Lowe, 2004)

Features are searched for in the image by *feature detectors* (see SIFT or PCA-SIFT algorithms mentioned in recent works for examples of feature detectors). *Feature detector* works as a decision maker:

it examines every pixel in the image to determine, if there is a feature at that pixel. Once a feature is detected, a *feature descriptor* describes its characteristics to make *feature* recognizable. The *feature descriptor* encodes the characteristics into a series of numbers and acts as a kind of numeric "fingerprint" that makes possible to clearly distinguish the important elements of scene from each other. This information should be invariant within the image transformation. It ensures that the same *feature* is findable even if the image has been transformed (eg. scaling, skewing or rotating). Further, it is desirable that the same *feature* will be found despite photometric changes, such as a change of light intensity or brightness. (Lowe, 2004)

Several *feature detector* algorithms have been developed to automate the process of detection features in images. The best-known of these are SIFT, SURF and ORB. The SIFT algorithm was introduced by Lowe (1999). The SURF algorithm, which is less computationally demanding than SIFT, was developed by extending SIFT algorithm later. ORB, the youngest of these three algorithms, is an alternative to SIFT and SURF. Karami et al. (2017) compared the performance of SIFT, SURF and ORB by applying them to transformed and distorted images. Based on their research, ORB has been shown to be the fastest algorithm out of the three tested. However, in the most cases it detects features in the middle of the image. Conversely, the computational speed of the SIFT is not as good as the ORB and SURF, but shows the best results for most scenes and detects features across the whole image.

Once *features* are detected and described, a database of images needs to be searched to find feature correspondences across different images. The issue is also referred as "finding the nearest neighbor". The correspondence of two *features* can be determined on the basis of two corresponding *feature descriptors* represented as vectors in multidimensional space. Correspondence algorithms are expected to be able to search only true correspondences.

Although existing algorithms for feature detection and feature description in images are designed to be resistant to photometric changes and image transformations, not all identified features are always described properly. Therefore, false correspondences occur and must be removed. The RANSAC algorithm is a suitable complement to feature detection and description algorithms (Fischer and Bolles, 1981). RANSAC can be used to find the true feature correspondences between two images that are mixed together with (many) outliers.

The aim of the solution described further in the manuscript is to use the above mentioned fundament to:

- Identify features in the images and finding best matching image from database of images.
- Calculate (indoor) position of a moving agent by comparison of image from an agent camera to the created image database and determination of the camera position.

3 Designed and implemented solution

Our solution for user's position determination in the building interior is based primarily on combination of principles of epipolar geometry and PnP problem and is very similar to the solution of calculating camera position in SFM method.

The essence of our solution is summarized below and visualized in the **Figure 1**.

Database preparation consists of following steps (1,2):

1) Surveying of the object coordinates (3D coordinates) of *ground control points* and taking image of interior.
2) Creation of image database and XML files for each image, which is consisted of 3D *ground control points* coordinates and 2D coordinates of their images.

The rest of the process consist of selection of the best matching image from the database to input image (input image refers to image captured by user) using SIFT algorithm and user's position estimation:

3) Position of projection center of camera C1 (database camera position) estimation – PnP problem
4) Essential matrix estimation from features detected between input image and database image via SIFT

5) Estimation of rotation matrix and translation matrix between database image and user's image
6) Scale estimation
7) Projection matrix P2 of user's image estimation and user's location estimation.

Figure 1. Essence of the proposed solution

3.1 Database preparation

The purpose of creating image database is to obtain the reference position of the camera, which will be the basis for further user's position estimation. For testing of our proposed solution, we use a large office space. (See Figure 2)

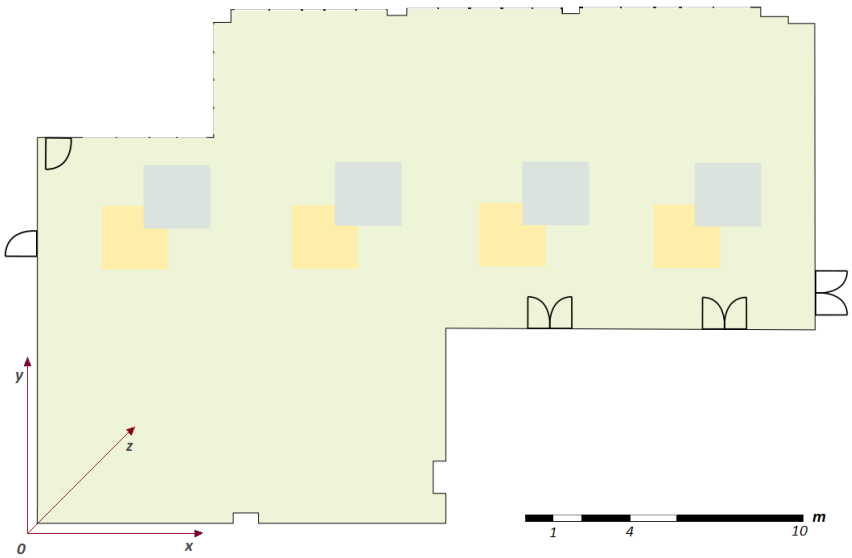

Figure 2. Office space

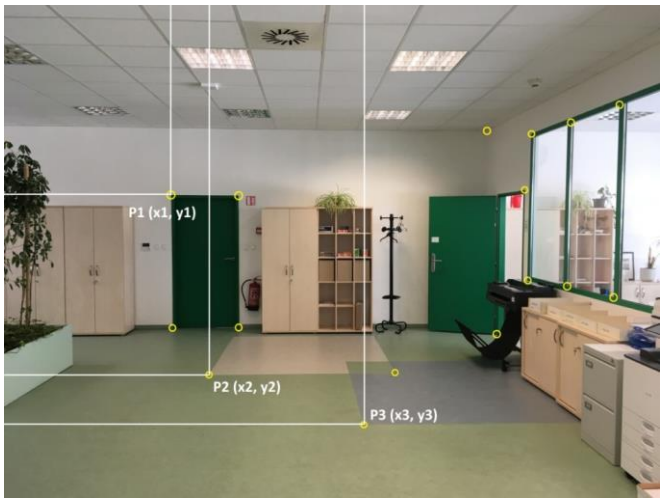

Figure 3. Distribution of ground control points in the database image

The database consists of imaginary directory describing the office interior and associated XML files storing the tuples of 3D surveyed ground control points coordinates together with their 2D image coordinates (see **Figure 4**). The interior of the building was photographed by IPhone SE mobile phone camera resulting in 60 images in total.

For position determination of the camera via single image, it is necessary to know at least 6 tuples. Ground control points were chosen to be invariant and well signalized (corners of windows/doors/room or floor patterns). While selecting ground control points in the images, it was also important to ensure that selected ground control points are not coplanar and are distributed as uniformly as possible (see **Figure 3**).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 April 2020 doi:10.20944/preprints202004.0032.v1

Peer-reviewed version available at ISPRS Int. J. Geo-Inf. 2020, 9, 368; doi:10.3390/ijgi9060368

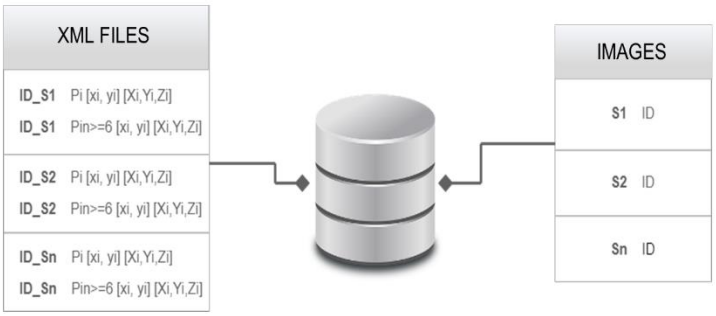

Figure 4. Database structure

3.2 Best matching database image selection

In order to select the best matching database image, we need to step through image database, detect features (if the features of database images are not pre-detected yet) in all images and assign correspondences between input image and database images. The best matching database image is the image with the highest number of possible correspondences retrieved. (See **Figure 5**)

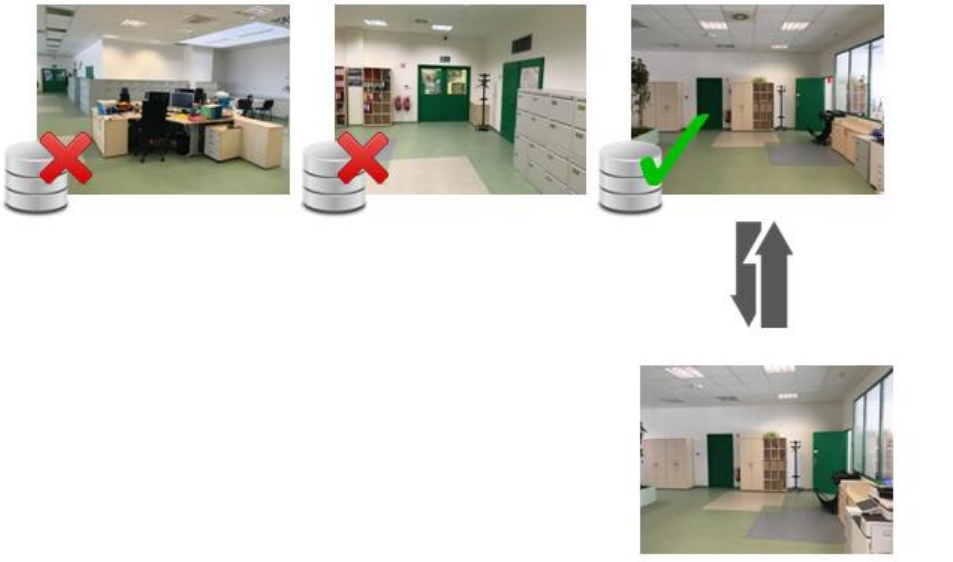

Figure 5. Best matching database image selection

Selection of suitable feature detection algorithm is crucial for our work, because detected features have not only function for searching of the best matching database image, but also play an important role in estimation of essential matrix. Although the computational speed of used algorithms is important factor in positioning process, it is more important to estimate the essential matrix from uniformly distributed and reliable detected and described features. For this reason, the SIFT algorithm was chosen for features' searching.

After feature detection and description, the next step is to find feature correspondences between input and database image (see **Figure 6**). Based on Muja & Lowe (2014) work, we use FLANN based matcher technique, that performs quick and efficient matching by using the clustering and search in multi-dimensional spaces module.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 April 2020 doi:10.20944/preprints202004.0032.v1

Peer-reviewed version available at ISPRS Int. J. Geo-Inf. 2020, 9, 368; doi:10.3390/ijgi9060368

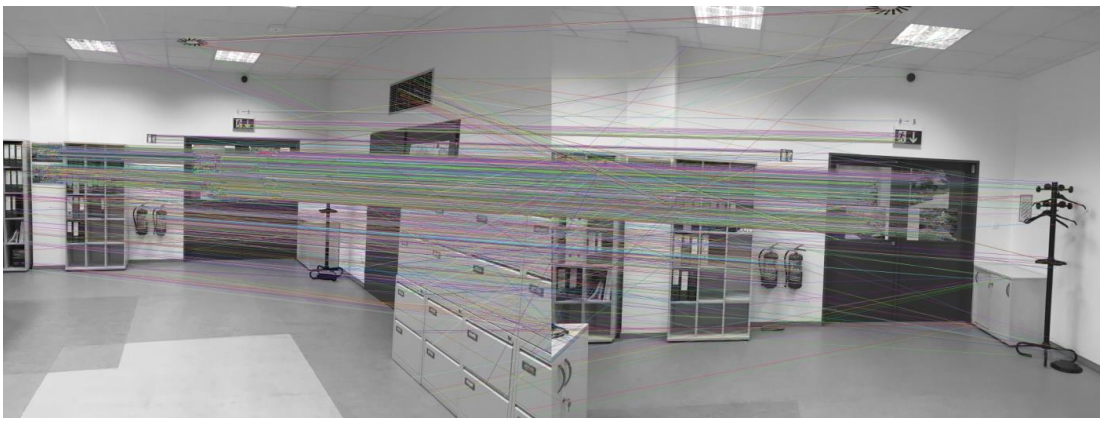

Figure 6. Feature correspondences between input and database image

3.3 Essential matrix E estimation

Having found the best matching database image and feature correspondences, we can proceed to the calculation of the fundamental matrix $\boldsymbol{F}$, which will be later useful for essential matrix $\boldsymbol{E}$ estimation. As shown in **Figure 6**, correspondences contain outliers. To remove outliers, the RANSAC algorithm is used. After fundamental matrix $\boldsymbol{F}$ estimation, we can estimate essential matrix $\boldsymbol{E}$, whose decomposition gives us a rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$ describing the relationship between positions of camera projection center $\boldsymbol{C}_1$ and camera projection center $\boldsymbol{C}_2$. Formula $\boldsymbol{E} = \boldsymbol{K}_2{}^T \boldsymbol{F}\, \boldsymbol{K}_1$ shows, that it is necessary to know the internal calibration elements of camera. Using a mobile phone camera, determination of internal calibration elements needs be carried out by calibration process. Agisoft Lens using a pinhole camera model for camera calibration was selected for this purpose.

After obtaining the values of essential matrix estimation, the next step is SVD decomposition of essential matrix. However, SVD decomposition provides two possible solutions for the rotation matrix $\boldsymbol{R}$ and two solutions for the translation vector $\boldsymbol{t}$. Combining two solutions for the rotation matrix $\boldsymbol{R}$ and the translation vector $\boldsymbol{t}$, we get four possible solutions for the projection matrix of user's camera. The way to find the right solution is a triangulation of image correspondence. If the triangulated point has a positive depth, i.e. lies in front of both cameras, the solution is considered as correct.

3.4 Projection matrix P1 estimation

Despite having the correct solution for the rotation matrix $\boldsymbol{R}$ and translation vector of user's camera relative to database camera, we cannot determine the user's location (projection center of camera $\boldsymbol{C}_2$) until we know the location of database camera (projection center of camera $\boldsymbol{C}_1$) in the object coordinate system. For database camera projection matrix $\boldsymbol{P}_1$ estimation, we use principles of PnP problem. The necessary tuples for PnP problem solution are stored in the XML file belonging to database image. Having estimated the projection matrix of database camera, we can extract the position of the database camera projection center $\boldsymbol{C}_1$ based on given formula:

$$\boldsymbol{P} = \boldsymbol{K}\,[\boldsymbol{R}|\boldsymbol{t}] = \boldsymbol{K}\boldsymbol{R}[\boldsymbol{I}, -\boldsymbol{C}]$$

$$\boldsymbol{Q} = \boldsymbol{K}\boldsymbol{R}$$

$$0 = \boldsymbol{P}\boldsymbol{C} = [\boldsymbol{Q}, \boldsymbol{t}]\begin{bmatrix}\boldsymbol{C}\\ 1\end{bmatrix} = \boldsymbol{Q}\boldsymbol{C} + \boldsymbol{t}$$

$$\Rightarrow \boldsymbol{C} = -\boldsymbol{Q}^{-1}\boldsymbol{t}$$

3.5 Scale estimation

Despite having the correct direction of the translation vector $\boldsymbol{t}$, we encounter fundamental problem. The estimated translation vector is unitary and we need to achieve its correct size. To reach the scale, we triangulate two ground control points having stored in our database image XML file. Using a pair of ground control points, it is possible to determine the distance of triangulated ground control

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 April 2020 doi:10.20944/preprints202004.0032.v1

Peer-reviewed version available at ISPRS Int. J. Geo-Inf. 2020, 9, 368; doi:10.3390/ijgi9060368

points in the camera coordinate system. Let us call this distance d_{img}. Furthermore, their actual distance d_{obj} in the object coordinate system can be easily calculated from 3D coordinates of control points. Finally, the resulting scale is determined using relationship:

$$s = \frac{d_{obj}}{d_{img}}$$

3.6 Projection matrix P2 estimation – user's position estimation

Finally, the projection matrix of user's camera can be estimated based on the following relationship:

$$\boldsymbol{P}_2 = \boldsymbol{K}_2\boldsymbol{R}\boldsymbol{R}_1[\boldsymbol{I}| - (\boldsymbol{C}_1 + \boldsymbol{t})]$$

In this case, the calibration matrix $\boldsymbol{K}_2$ of user's camera is the same as calibration matrix $\boldsymbol{K}_1$ of database camera. After substituting all needed values into above formula, we can extract the user's position the same way as in the case of database camera projection center $\boldsymbol{C}_1$.

4. Results and Discussion

Our proposed solution was tested on two different views (see **Figure 7**). A total of 12 input images were tested for the view 1. Input images were tested against a set of 14 database images. For the view 2, 5 input images were tested against database of 9 images. After finding the corresponding database image and estimating the user's position, the pair of input image and database images was switched. It means that the best matching database image was used as an input image and input image as a best matching database image. The assumption was that we obtain very similar result for location error. Although our theory was confirmed in most cases, there were some significant location errors.

Figure 7. Tested view 1 (left image), tested view 2 (right image)

Table 1. Results

	Input image	Database image	A total number of feature correspondences	Projection center position of database image			Reference position of projection center of input image			Estimated user's position			Position error
				X[m]	Y[m]	Z[m]	X[m]	Y[m]	Z[m]	X[m]	Y[m]	Z[m]	mxy [m]
1	I_4898	I_4962	2710	16.90	8.44	1.59	18.41	9.75	1.67	18.31	9.68	1.63	0.09

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 April 2020 doi:10.20944/preprints202004.0032.v1

Peer-reviewed version available at ISPRS Int. J. Geo-Inf. 2020, 9, 368; doi:10.3390/ijgi9060368

	I_4962	I_4898	2470	18.41	9.75	1.67	16.90	8.44	1.59	16.95	8.35	1.66	0.07
2	I_4297	I_4898	2621	18.41	9.75	1.67	16.14	8.72	1.64	16.21	8.92	1.77	0.15
	I_4898	I_4297	2638	16.14	8.72	1.64	18.41	9.75	1.67	18.25	9.62	1.64	0.15
3	I_4745	I_4743	2414	15.55	10.34	1.65	17.29	8.57	1.60	17.20	8.78	1.60	0.16
	I_4743	I_4745	2579	17.29	8.57	1.60	15.55	10.34	1.65	15.82	10.16	1.56	0.23
4	I_4954	I_4958	1769	12.40	8.42	1.63	12.64	9.66	1.59	12.76	9.90	1.57	0.19
	I_4958	I_4954	1502	12.64	9.66	1.59	12.40	8.42	1.63	12.52	8.78	1.60	0.27
5	I_4909	I_4908	2532	9.75	9.30	1.62	9.76	9.20	1.61	9.49	9.10	1.64	0.20
	I_4908	I_4909	2868	9.76	9.20	1.61	9.75	9.30	1.62	9.67	10.35	1.62	0.74
6	I_4956	I_4955	2435	12.49	9.68	1.65	12.23	8.44	1.69	12.47	8.12	1.63	0.28
	I_4955	I_4956	2516	12.23	8.44	1.69	12.49	9.68	1.65	12.33	10.05	1.70	0.29
7	I_4955	I_4954	3065	12.64	9.66	1.59	12.49	9.67	1.64	12.83	9.80	1.58	0.26
	I_4954	I_4955	2762	12.49	9.67	1.64	12.64	9.66	1.59	12.34	9.62	1.63	0.21
8	I_4961	I_4962	2852	16.90	8.44	1.60	15.17	8.39	1.60	15.25	8.36	1.60	0.06
	I_4962	I_4961	2835	15.17	8.39	1.60	16.90	8.44	1.60	16.76	8.44	1.60	0.10
9	I_5071	I_4956	1790	12.25	8.44	1.68	11.92	8.06	1.82	12.41	7.64	1.63	0.46
	I_4956	I_5071	1933	11.92	8.06	1.82	12.23	8.44	1.69	11.92	8.06	1.82	0.35
10	I_5072	I_4908	1637	9.75	9.30	1.62	9.85	9.21	1.60	9.71	9.21	1.60	0.10
	I_4908	I_5072	1739	9.85	9.21	1.60	9.75	9.30	1.62	9.86	8.78	1.60	0.38
11	I_5074	I_4743	2674	15.55	10.34	1.65	15.82	9.23	1.58	15.47	9.58	1.63	0.35
	I_4743	I_5074	2669	15.82	9.23	1.58	15.55	10.34	1.65	15.52	10.18	1.56	0.12
12	I_5082	I_4909	1577	9.76	9.20	1.61	9.72	9.22	1.59	9.81	9.21	1.60	0.06
	I_4909	I_5082	1473	9.72	9.22	1.59	9.76	9.20	1.61	9.60	9.24	1.75	0.12
13	I_4740	I_4742	1156	7.62	8.16	1.59	8.67	8.70	1.62	8.79	9.38	1.63	0.49
	I_4742	I_4740	881	8.67	8.70	1.62	7.62	8.16	1.59	7.93	7.61	1.62	0.45
14	I_5084	I_4913	1338	8.43	8.08	1.65	8.57	7.71	1.5	8.61	7.66	1.50	0.05
	I_4913	I_5084	1186	8.57	7.71	1.5	8.43	8.08	1.65	8.49	8.32	1.63	0.17
15	I_5086	I_4944	702	6.72	9.65	1.61	6.58	10.00	1.67	6.71	9.65	1.62	0.26
	I_4944	I_5086	718	6.58	10.00	1.67	6.72	9.65	1.61	6.74	9.98	1.67	0.23
16	I_4914	I_4913	2332	8.42	8.08	1.64	8.45	8.01	1.66	8.38	8.15	1.65	0.11
	I_4913	I_4914	2101	8.45	8.01	1.66	8.42	8.08	1.64	8.36	7.31	1.72	0.55
17	I_4944	I_4913	893	8.42	8.08	1.64	6.72	9.65	1.61	6.90	9.17	1.59	0.36
	I_4913	I_4944	1030	6.72	9.65	1.61	8.42	8.08	1.64	8.06	8.56	1.64	0.42

The

Table 1 above show that this problem applies to pair of images 5 and 16. Despite higher number of feature correspondences (2868) in case of pair number 5, we received positioning error 0.74 m. The higher value of positioning error was caused due to wrong translation vector estimation. Wrong translation vector determination was noted especially in cases where the length of the baseline between projection centers of cameras was very short and the projection centers were almost identical. In the case of pair number 5, the baseline was 0.1 m long as well as for the pair number 16. Such a case can be detected and excluded from an automatic positioning algorithm.

The values of mean coordinate errors in the graph below confirm that the positioning error decreases with longer baseline between projection centers of user camera and database camera. Generality of this trend can be assumed however it has not been sufficiently statistically tested.

If we look at the number of feature correspondences found and the mean coordinate error in positioning, we find cases when positioning error is higher despite higher number of feature correspondences. The fact that higher number of feature correspondences does not always guarantee higher position accuracy led to another test of our proposed solution. The idea of next test is to estimate user's position with decreasing number of feature correspondences.

From the above results it is evident that the number of retrieved feature correspondences does not affect the resulting positioning accuracy if the scene is the same. Based on given results, the image database for view 1 was limited from the original 14 images to 5 images and the test was performed again.

For more than 50% of tested input images, the position was determined with the positioning error up to 0.1 m, for almost 30% the positioning error up to 0.20 m and for the remaining 20% of input images the positioning error up to 0.50 m. Moreover, these results show a very important insight. The lower the number of stored images in the database capturing the same scene of the interior, the higher the probability of accurate positioning. In other words, a low number of stored images in the database reduces the probability of retrieving the best matching image from database with a similar projection center to user's camera.

Our proposed solution for location estimation in building interiors using mobile phone camera was introduced. However, the proposed solution has its limitations that are needed to be discussed. The first is the accuracy determination of position in real use. If the user wants to verify the accuracy of estimated position, a possible solution could be taking multiple images from one location to multiple sides and the comparing obtained positions.

Furthermore, the same mobile phone camera was used when taking input images and database images. We have already had detected the internal calibration parameters of given mobile phone camera by Agisoft Lens software. In real deployment can be expected to differ internal calibration parameters of mobile phones. For this reason, the user would need to find out calibration parameters of his/her mobile phone.

It was found, that the most feature correspondences are found in image areas where regular patterns, pictures or text characteristics occur. Although one of our tests showed the indirect proportion between the number of feature correspondences and positioning error, there could be a case of uninteresting scene (e.g. white walls) in terms image processing. This scene would therefore miss sufficiently contrasting features that could result in low number of feature correspondences and in impossibility of position estimation. It would be therefore appropriate to add pictures or text characteristics as mentioned in the article of authors Sadeghi et al. (2015).

The computational time of SIFT algorithm is another drawback of our proposed solution. Feature detection and description in a single image takes approximately 3 minutes using hardware with following parameters – 1.8 GHz CPU, 2 GB GPU, 6 GB RAM. Despite low number database images, there would be no real use in terms of such computational demands on hardware. Computational demands of the proposed solution could be reduced by having detected and described features stored in the database. Feature detection and description with SIFT algorithm would be related to user-captured image only. Another solution for reducing the computational complexity could be hierarchization of images. This means that individual database images would be evaluated according to the frequency of their selection from the database. In addition, in real life deployment would have been the case of client-server architecture that separates the client and server via computer network. Request for location estimation would be sent through the application to a web server equipped with more powerful hardware.

As for the appearance of the interior of the building, it should be also noted that it could change over time. Some equipment may be moved, removed or vice versa added. Such environmental changes would unfavorably affect the number of inlier feature correspondences between database

image and input image. As a result, this would happen to inaccurate or even impossible positioning. Image database would therefore need to be updated in real use.

5. Conclusions

As showed in the Recent works chapter indoor localization is a problem with many possible solutions. It is still to be discover witch will prove to be usable in everyday use and thus step outside of the academic ground. We believe that the reason for slow transition to wireless beacon localization is the need for new onsite infrastructure. Our solution does not require any new infrastructure and therefore should prove to be easier to implement. Taking a single image is a simple action that should not bother the user heavily and as seen in Results and Discussion the localization is reliable even with very small pool of database images. Further research is of course necessary, e.g. the dependency among unchanged image features needs to be further tested.

Funding: the research reported in this paper has been supported by Project LO1506 of the Czech Ministry of Education, Youth and Sports.

Conflicts of Interest: The authors declare no conflict of interest.

Author Contributions: Hana Kubíčková is the main author of the manuscript. The manuscript is output of her master thesis, which was led by Karel Jedlička, he is the author of the overall vision. Radek Fiala worked on photogrammetric and mathematical backgrounds together with Hana. Daniel Beran interpreted the results and formulated the discussion and conclusion.

References

1. Agisoft LLC. (2018) Agisoft Lens.Version 0.4.1. Petrohrad.
2. Deretey, E., Ahmed, M. T., Marshall, J. A., & Greenspan, M. (2015, October). Visual indoor positioning with a single camera using PnP. In 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN) (pp. 1-9). IEEE.
3. Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381-395.
4. Hartley, R. I., & Mundy, J. L. (1993, September). Relationship between photogrammmetry and computer vision. In Integrating photogrammetric techniques with scene analysis and machine vision (Vol. 1944, pp. 92-106). International Society for Optics and Photonics.
5. Hartley, R., & Zisserman, A. (2003). Multiple view geometry in computer vision. Cambridge university press.
6. Hassaballah, M., Abdelmgeid, A. A., & Alshazly, H. A. (2016). Image Features Detection, Description and Matching. Studies in Computational Intelligence, 11–45. doi:10.1007/978-3-319-28854-3_2
7. Hile, H., & Borriello, G. (2008). Positioning and orientation in indoor environments using camera phones. IEEE Computer Graphics and Applications, 28(4), 32-39.
8. Karami, E., Prasad, S., & Shehata, M. (2017). Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images. arXiv preprint arXiv:1710.02726.
9. Kawaji, H., Hatada, K., Yamasaki, T., & Aizawa, K. (2010, October). Image-based indoor positioning system: fast image matching using omnidirectional panoramic images. In Proceedings of the 1st ACM international workshop on Multimodal pervasive video analysis (pp. 1-4). ACM.
10. Lowe, D. G. (1999, September). Object recognition from local scale-invariant features. In iccv (p. 1150). Ieee.
11. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110
12. Liang, J. Z., Corso, N., Turner, E., & Zakhor, A. (2015). Image-based positioning of mobile devices in indoor environments. In Multimodal Location Estimation of Videos and Images (pp. 85-99). Springer, Cham.
13. Muja, M., & Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. IEEE transactions on pattern analysis and machine intelligence, 36(11), 2227-2240.

14. Namiot, D. (2015). On indoor positioning. International Journal of Open Information Technologies, 3(3), 23-26.
15. Ravi, N., Shankar, P., Frankel, A., Elgammal, A., & Iftode, L. (2005, August). Indoor localization using camera phones. In Seventh IEEE Workshop on Mobile Computing Systems & Applications (WMCSA'06 Supplement) (pp. 1-7). IEEE.
16. Sadeghi, H., Valaee, S., & Shirani, S. (2015, April). Ocrapose: An indoor positioning system using smartphone/tablet cameras and OCR-aided stereo feature matching. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 14731477). IEEE.
17. Van Opdenbosch, D., Schroth, G., Huitl, R., Hilsenbeck, S., Garcea, A., & Steinbach, E. (2014, October). Camera-based indoor positioning using scalable streaming of compressed binary image signatures. In 2014 IEEE International Conference on Image Processing (ICIP) (pp. 2804-2808). IEEE.
18. Werner, M., Kessel, M., & Marouane, C. (2011, September). Indoor positioning using smartphone camera. In 2011 International Conference on Indoor Positioning and Indoor Navigation (pp. 1-6). IEEE.
19. Nishkam Ravi, Pravin Shankar, Andrew Frankel, Ahmed Elgammal, and Liviu Iftode, "Indoor Localization using Camera Phones," in Mobile Computing Systems and Applications, 2006