IMAGO: Image-guided navigation for visually impaired people

Stephan M. Jonas [a,*], Ekaterina Sirazitdinova [a], Jan Lensen [b], Deyvid Kochanov [a], Humaam Mayzek [d], Tjeu de Heus [c], Richard Houben [b], Hans Slijp [b], and Thomas M. Deserno [a]

[a] *Department of Medical Informatics, Uniklinik RWTH Aachen, Germany*
[b] *Applied Biomedical Systems, Maastricht, The Netherlands*
[c] *FileFlow/De Heus Beeldvorming, Sittard, The Netherlands*
[d] *I-Cane Social Technologies, Sittard, The Netherlands*

Abstract

Blind and visually impaired persons face many challenges due to isolation, the most important is lacking education due to social immobility. Yet, since the development of the well-known white-red cane, only few advances have been made to increase the mobility of blind people. GPS systems are often used for pedestrian navigation but lack precision, foremost in rural areas where navigation is most often needed. The aim of the IMAGO project is therefore the development of an inexpensive and unobtrusive navigation method for blind and visually impaired persons. Navigation is performed using structure from motion and image-based localization techniques. Route models are created as 3D point clouds through several steps: (i) image acquisition along the routes; (ii) bulk-transfer of images to a server; (iii) feature extraction; (iv) feature matching between images; (v) creation of 3D point cloud with structure from motion. Similarly, the navigation chains seven steps: (a) image acquisition while walking a route; (b) immediate transfer of image to a server; (c) feature extraction; (d) feature matching between image and 3D route model; (e) localization/camera matrix calculation; (f) navigation/calculation of direction based on localization; (g) transfer of direction to user. Current smartphones are used as devices both for recording of routes as well as navigation. Thereby, a high level of dissemination without additional costs is possible, both, within blind people for navigation, as well as seeing people for route creation. Additionally, haptic feedback can be used via a smart cane to reduce auditive feedback. The proposed system yields a high positioning accuracy of 80% of samples being located within 1.6 m. Thus, the system is usable for pedestrian navigation, especially for visually impaired persons.

Keywords: Visually impaired, navigation, structure from motion, image-based localization

1. Introduction

According to the World Health Organization (WHO), the population of blind and people with low vision was estimated to be 285 million worldwide in 2014 [28]. Since the introduction of the well-known white-red long cane, only small advances have been made to increase independence and social involvement through mobility of blind people. Guide dogs can help to cope with dangerous situations, but cannot support with mobility issues. Even the deployment of hand-held GPS units made only little impact to mobility for blind and visually impaired persons (VIP) because of insufficient precision and reliability. Therefore, VIP still depend on memorizing routes with the support of their fellow humans. Travelling outside these known routes requires companionship, energy and efforts. As a result, visual impairment creates social and economic isolation and lacking education [15]. Support in way-finding of VIP is therefore needed.

During the last years, the interest in pedestrian positioning and navigation has increased significantly [2, 17]. One common technique for positioning is global navigation satellite systems (GNSS) such as the Amer-

*Corresponding author. E-mail: sjonas@mi.rwth-aachen.de.

ican global positioning system (GPS) or Russian global navigation satellite system (GLONASS) [12]. Technological advances in microelectronics have reduced the size of receivers for GNSS and antennas from an order of cubic decimetres ten years ago to the size of a coin today, enabling mobile pedestrian navigation applications. However, without special measures, pedestrian navigation will suffer from limited availability and a low positioning accuracy especially in signal-degraded environments like urban areas. In urban situations, the GPS signal suffers from houses or other structures blocking and/or reflecting the signal, and from lacking clear view to the sky [20].

In navigation of VIP, two large challenges have to be overcome. First, positioning has to be very precise to make sure that the guided person is walking, for example, on the pavement and not in the middle of the street. Second, map-based feedback to the user has to be modified as displaying the route is not an option. Additionally, VIP are very reliant on their hearing sense to navigate through traffic [6]. Therefore, auditive feedback is not an appropriate option and haptic feedback has to be used.

1.1. State of the art

GPS is widely used in various navigational devices and applications. Being available on the most of modern smartphones, with help of additional context information (e.g., map-based graphical representation of a city area), GPS can provide assistance in navigation [12]. However, even with optimal calibration, GPS is not sufficiently accurate to guide VIP pedestrians. Especially in urban areas, where tall buildings block or reflect satellite signals, the positioning error of GPS is about 34 m [20]. While Wi-Fi networks in the area can be used to improve the localization, no report on the accuracy of such Wi-Fi-supported GPS exists to the knowledge of the authors. Since roads are shared with other participants like cars, deviation from safe routes is potentially harmful.

Other navigational approaches use the number of steps detected by an accelerometer, reference points and a mobile compass for navigation assistance. Fallah et al. [3,7] presented a successful example of this method combining probabilistic algorithms with natural capabilities of visual impaired people to detect objects by touch. However, this system is designed for indoor environments, where maps are very accurate and clear landmarks (e.g., corners and doors) are available. Recent research exploits stereo vision cameras or depth cameras like the Microsoft Kinect to navigate around obstacles [8]. The range of these devices is limited only up to a few meters, and therefore, they are not fit for general navigation tasks requiring localization in outdoor environments.

More recently, radio frequency identification (RFID) technology has found its use in the research area of navigation of VIP. One of the latest systems in this area was proposed by Varpe & Wankhade [27]. On the user side they apply a mobile RFID reader, a transceiver for transmitting the tag's information, and an audio device to provide feedback to the user. To identify walking routes, an RFID passive tag network is employed on the path. Although, the accuracy of such systems might yield precisions of 1.55 - 3.11 m, it requires additional objects (i.e., RFID-tags), which makes this technology costly and not easily adoptable for new environments [18].

Given these points, most technologies that can be used in navigation of VIP today have major limitations. Some have poor reliability in different conditions because of inaccuracies in measurement devices, some suffer from being restricted to certain area, while others are too costly or too large and obtrusive for everyday use.

An alternative to the prior mentioned approaches is image-guided navigation. With this technique, images are captured at the current position and the position is calculated based on landmarks or prominent points on the image. So far, image-guided localization is used mostly in robotics, where maps of the environment are built while a robot is discovering its environment.

Simultaneous localization and mapping (SLAM) is used often in robotics to map a new surrounding for navigation [25]. This approach bundles building the map (mapping) and localization into one combined step. It requires multiple sensors or sources of information, for example a laser scanner and the known trajectory of the robot. One variety of this method is visual SLAM (vSLAM), which uses camera sensors as input [14]. The method usually requires two cameras for stereo vision, but single camera approaches do also exist (MonoSLAM) [4]. By tracking features from one frame to the next, the 3D geometry of the surrounding is calculated. Using only a single camera is computationally expensive due to the simultaneous calculation of the position and the map, especially in high-resolution images that would be needed to capture landmark structures from a distance.

Structure from motion (SfM) is a current 2D image to 3D reconstruction method that can be used to

generate city-scale models of real world scenes using unordered sets of 2D images [5]. SfM reconstructions provide compact 3D models in the form of sparse point clouds. To perform reconstruction, points of interest with accompanying features are extracted from the photographs. Extracted points from one image are matched with the points from every other image to find overlaps. A 3D point cloud is then incrementally updated with points from new images. Each image has to have at least two corresponding images in the cloud to compute a *triangulation* of feature points in 3D. 3D coordinates of each point are arbitrary, however, a similarity transform to real world coordinates can be estimated, for example if GPS data is available. By matching features from a new image to the points in the model, the position of the camera during acquisition relative to the model can be estimated [21].

1.2. Aim of the project

Since mobility is the limiting factor in social engagement of VIP, the goal of this project entitled IMAGO is to increase independence and social mobility by developing and testing a positioning and navigation technology, exceeding the quality, accuracy and applicability of positioning technology based on satellite data. The project focuses on technology research and development activities regarding the use of images for high accurate positioning within the boundary conditions of being unobtrusive, inexpensive and easy to deploy and maintain. To solve the accuracy problem of localization it relies on advances in SfM techniques. The input interface should be adapted to the user's needs and allow to enter route information via voice commands. Output of directions should be communicated in a non-obtrusive way for as many senses as possible. Thus, a tactile arrow located on the white cane has been chosen in this work.

Driven by the aim of providing an affordable and easy-to-carry tool to assist blind and visually impaired people, a smartphone-based navigational application for visually impaired people is created.

2. Methods

The approach of navigating a person based on images uses three steps: (i) route model preparation, (ii) localization and (iii) navigation. In the model generation step, images of a route have to be acquired and transmitted to a server. On the server, a 3D model representing the route is built from the images and additional data. In the localization and navigation step, a second smartphone application is used to pick the desired route to walk. Then again, image and sensory data is acquired by the app, send to the server and the current position is calculated. Based on the position, the navigation is performed in step three and directions are provided to the user.

2.1. Route model preparation

For model creation, a person has to walk the route at least once. During this walk, images are acquired with a specifically designed recording app (Fig. 1). Additional data, like weather and environment information, is embedded into the images and transferred to a server, where a 3D model is created using SfM. The created model can then be shared with the community of users. This allows a crowdsourcing of the model building, which could on one hand benefit the acceptance and dissemination of the system. On the other hand, no special hardware can be used if crowdsourcing is desired. We therefore focus on consumer grade hardware for the image acquisition.

2.1.1. Acquisition

To ensure a wide accessibility and availability of the proposed technique, smartphones have been chosen as image recording devices. Smartphones contain many sensors such as GPS and accelerometer that can also help to improve or speed up the calculation of localization and navigation. Specifically for the acquisition prototype, the iPhone 5 has been chosen as it features an 8 megapixel camera of fairly quality. It also features a wireless broadband connection, which is necessary for the first iteration of the prototype to transmit data between smartphone and server. Since the smartphone is mostly used as a remote camera and to connect peripherals like a tactile arrow, a similar Android or Windows operated devices could easily replace it.

It has been discussed that a fisheye lens is best for pose estimation as they cover a greater part of the scene [24]. However, as we are aiming at providing an affordable tool, it has been decided to avoid extra costs a user would have to bear by buying an extra fisheye lens. So, the built-in iPhone lens is used instead.

For image recording, the camera must have a good view at objects and facades in front of the pedestrian. It should always be in the same position on the user, front-facing, and be as stable as possible to reduce motion blur. To achieve this, a chest-mount is used to strap

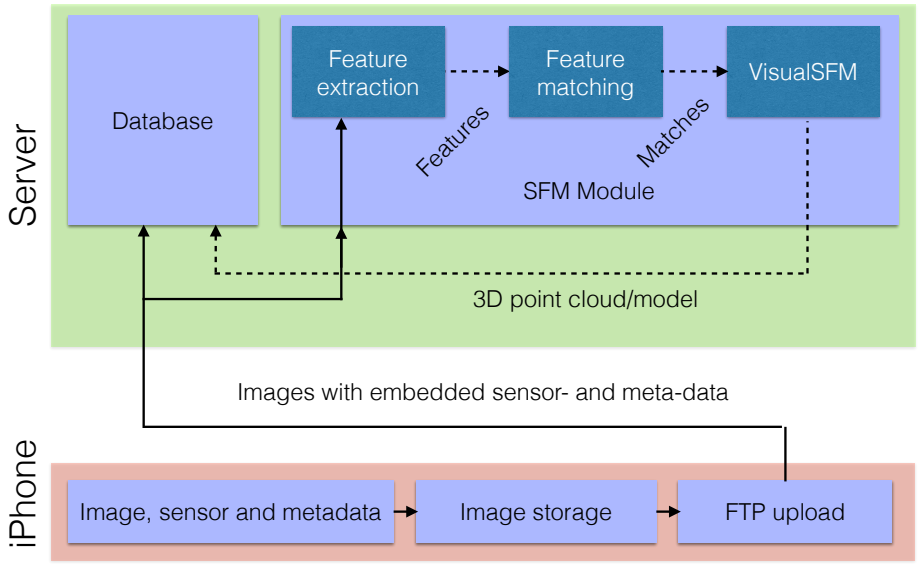


Figure 1. Workflow of image acquisition and model creation

the iPhone to a person. This has the additional benefit of freeing the pedestrian's hands (Fig. 2). Thereby, a stable and fixed position of the recording device is achieved.

Before recording a walk, the recording application requires the user to enter the following information:

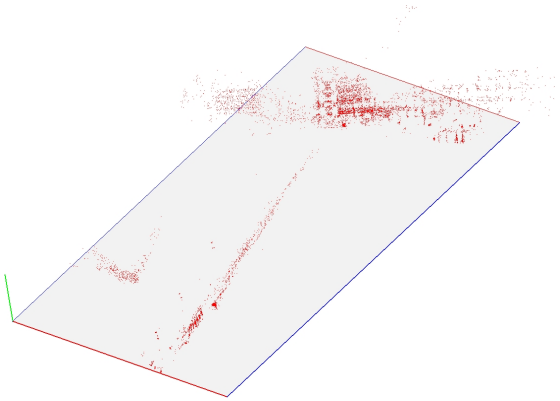
- Route name: a pseudo for the route to be recorded (e.g., "Maastricht downtown route" or "Aachen central market to main station").
- Transportation method: information on the used form of transportation, which impacts the distance covered between two images (e.g., walk or wheelchair).
- Weather: information on the time of day, lighting, weather and environmental conditions that are important for modelling (day/night, cloudy/sunny, wet/dry).

During acquisition, one image is taken every second and data from GPS, gyroscope, accelerometer and magnetometer is captured at 10 Hz. This data is embedded into each image along with all meta-data (user-entered data and device information). All images are stored locally on the smartphone. After a walk, locally stored images are uploaded onto the server via the file transfer protocol (FTP).

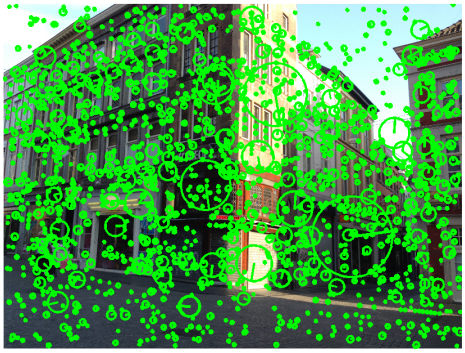
2.1.2. Reconstruction/Structure from motion

The reconstruction process consists of three major steps: feature detection and extraction, feature matching, and bundle adjustment (Fig. 3). Several packages implementing the full SfM pipeline are available. Here, VisualSFM [29] was chosen due to GPU support and reconstruction performance. Feature extraction and matching can be performed outside of VisualSFM to speed up matching and use additional features not natively supported by VisualSFM. Scale invariant feature transform (SIFT) features were chosen, as they have shown to perform reasonably and yielding good results [1,19]. The VisualSFM tool allows incremental model building, which is important in this particular setting since the entire data is acquired iteratively. It is modular and allows modification of some of it's components, which is helpful for these specific tasks.

Reconstruction results in an SfM workspace, stored into an n-view match (NVM) file, which contains input image paths and multiple 3D models, each being one 3D point cloud corresponding to an overlapping sequence of images. Multiple models per route are occurring regular, as quick turns during the acquisition can yield in too little overlap between subsequent images. Saving the reconstruction to a `[name].nvm` file also



(a) Resulting 3D point cloud of a street intersection



(b) Image with feature key points. Circle size and bar indicate scale and rotation respectively

(c) All feature matches between two images

(d) Strongest or so-called inlier-matches between two images

Figure 3. Route 3D model reconstruction

yields several `[name].i.ply` files, each containing the data for the *i*-th model. Each reconstructed model holds the number of cameras, list of cameras, number of 3D points, and list of feature points. For each camera, image file name, focal length, camera center, quaternion rotation and radial distortion are identified. Each point is defined by XYZ coordinates, RGB values, its number of measurements and a list of those measurements. To reduce the model size and thereby the number of matches required during the localization, the measurements for each individual point are

(a) Smartphone chest mount

(b) Image acquired with wide-angle lens

(c) Image acquired without wide-angle lens

Figure 2. Image Acquisition

averaged, reducing the number of matches needed approx. 5-fold.

One major limitation of this reconstruction method is the outcome consisting of several models. The specific acquisition technique impacts the way how a 3D model is built: as images are acquired when a user is walking, the image sequences produced are sometimes not dense enough to provide the necessary correspondences between extracted features. That typically requires multiple walks through a route and leads to multiple models in the outcome. A joint model needs to be computed to provide navigation routes.

2.1.3. Outlier removal

Comparing the large amount of points from model and from the current photograph is computationally expensive. Due to storage limitation, the number of points must be reduced without affecting the positioning accuracy. This can be achieved by removing outliers.

According to the definition of Grubbs [11], an outlying observation, or outlier, is "one that appears to deviate markedly from other members of the sample in which it occurs". Outliers in a 3D point cloud may be of different nature. Firstly, they may result from errors occurring during the reconstruction process, such as inherent inaccuracies in feature detection, false matching, and errors in estimation of fundamental and projection matrices. Second, non-static environment objects (e.g., cars, chairs and tables of street cafes, advertisements, market stalls) create reconstruction noise.

In most vision-based city reconstruction approaches, outliers are removed only within the reconstruction process, and "cleaning" techniques are not applied directly to the point clouds [1,10,13,22]. The problem of outlier removal in 3D point clouds has not been evaluated from the perspective of a localization task before.

City-scale 3D point clouds are large arbitrary datasets, and, therefore, outlier removal shall be computationally efficient and being able to be performed without any additional user interaction. To assess the potential of computational speedup, an original automatic distance-based method of outlier detection in 3D point clouds was proposed.

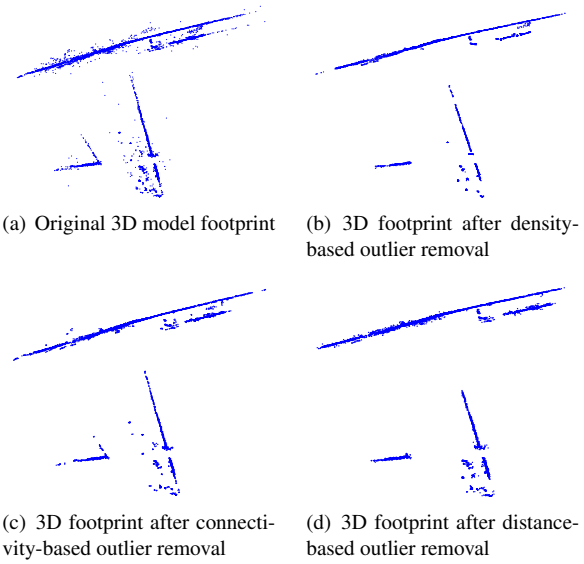
In the proposed approach, the notion of distance-based outliers DB proposed by Knorr and Ng [16] for data-mining applications is adopted: "An object O in a dataset T is a $DB(p, D)$ outlier if at least fraction p of the objects in T lies greater than distance D from O". Assuming that points belonging to buildings and man-made walls are normally distributed, a double-

threshold scheme is applied: first, the individual points far away from point clusters in the model are removed using the relative distance to the other points in the model. After eliminating such points, the second filtering factor based on the global mean over mean distances of each point's neighborhood is estimated. The choice of the parameters is explained by consideration of the following constraints.

Noise preserved in the model after removing outliers is used for qualification (evaluation). Noise level stands for the relative number of points or point clusters remaining in the model. Particularly this characteristic is important for alignment of the reconstructed facades (so-called building footprints) with a map. This approach is used for merging separate model fragments in the same coordinate space (Section 2.1.4). A high level of noise can adversely affect alignment, as the outlying points can drag a model towards the wrong walls.

Removing as many outliers as possible, the main constraint for selection of parameters for outlier detection was retainability of the model's structure, or, in other words, presence of all significant walls in the model after outlier removal. This constraint is important for navigation, because an area as big as possible shall be covered with with the models. At the same time, the correctness of the model alignment, again, highly depends on the footprint structure. In some cases, even a small object can solve ambiguity of scaling parameters and thus the right model placement. Therefore, it is rather important to preserve the majority of walls during outlier removal.

Adjusting the parameters, a trade-off between the level of noise and model's retainability is found (Fig. 4).



(a) Original 3D model footprint

(b) 3D footprint after density-based outlier removal

(c) 3D footprint after connectivity-based outlier removal

(d) 3D footprint after distance-based outlier removal

Figure 4. Comparison of outlier removal effect on 3D model. For a better visibility, the models are rotated upright according to the pre-computed model's gravity vector.

2.1.4. Model alignment

For model alignment, the methods of Strecha et al. [23] and Untzelmann et al. [26] were taken as guidelines. In those works, separate models are aligned to each other in the same coordinate space with a help of freely available geographical digital models. For that purpose, OpenStreetMap (OSM, www.openstreetmap.org/) was chosen as the ground model, and the alignment is performed initially by using the available GPS metadata.

To align a model to the corresponding building block in OpenStreetMap, first it is necessary to project the model to 2D by rotating it upright and removing structures not belonging to building facades. It results in a set of projected points corresponding to facades and forming footprints of buildings. Reconstructed cameras are projected together with the point cloud. A similarity transformation between reconstructed cameras' projected positions and their geographical positions translated from GPS into universal transverse mercator (UTM) format is applied for a rough alignment of the footprints with a map. This step shows that initial GPS positioning is not precise (in our experiments using the localization service including GPS and Wi-Fi information the error varies from 2 to 40 m). In order to correct for that, the transformation parameters are iteratively refined so that the projected footprint is optimally aligned with the building outline on the corresponding OpenStreetMap segment (Fig. 5). Since the refinement step involves point-to-wall distance minimization, outlier removal has to be performed prior to refinement.

2.2. Localization

Localization has two components running on the mobile device and the server (Fig. 6). The mobile application is again an iPhone application which operates on a route on the server. The route is requested in an initial step and the route model is stored in a session on a server. The client app is similar to the image acquisition application as it also captures one image per second. However, the image is immediately sent to the server from which it receives a direction based on the

Figure 5. Alignment of a model (points (green) and cameras (red)) to the OpenStreetMap outline (blue).

submitted image. The server is implemented in Matlab due to its availability throughout the project consortium. Closer to market introduction, the code will be reprogrammed into partial native smartphone code (Java, Objective-C, C#) and efficient server-side code (CUDA). Besides managing the sessions, the server also performs the image-based localization, orientation and navigation. A sensor-based localization module can be added in later stages of this work. Based on the localization and the given route, the navigation module calculates the direction.

2.2.1. Image-based localization

On the server side, during image-based localization, the camera position of a new image has to be localized within the aligned route model. Therefore, the first step is to extract SIFT features from the image. The features are then matched against all possible points within the 3D point cloud that is the route model. After matching, the server estimates the location based on the observed matches. Since some of the found correspondences will be outliers, the pose estimation procedure is wrapped in a random sample consensus (RANSAC) loop [9]. RANSAC picks a random subset of matches and uses them to generate a hypothesis about the pose. It then tests the hypothesis against the full set. If the number of matches is large enough, RANSAC terminates returning the set of inliers and a pose estimated from them. In the pose estimation procedure, the unknown internal and external camera parameters are estimated using six or more correspondences. In addition, a normalization of the data points is applied to improve the numerical stability. The normalization happens by centering the mean of the points at 0 and rescaling them with their root-mean-square (RMS) distance to the origin.

2.3. Navigation

The navigation combines the camera position with the predefined route from the database. Initial talks with VIP have shown that they are confident to walk on a sidewalk on their own. Therefore, a turn-based navigation is performed: the navigation only indicates changes when a turn is necessary or upcoming, or if the user has deviated too far from the route. In contrast, step-wise navigation gives continuous direction.

The navigation discriminates between four major states to calculate a direction for the user to walk:

- Off-route: if the user is not in close proximity to the route but can be localized, the direction is calculated based on the shortest way between the current position and the route. This step is repeated until the user gets into close proximity with the actual route.
- On route: the user is following the route along a straight segment. Since no direction change is necessary, the direction straightforward is returned.
- Close to turning point: the direction of the turn is indicated in advance of the actual turn so that the user can prepare for the change.
- On turning point: the actual turn is indicated.

In a final step, the result of the navigation, the direction for the user to walk, is being sent back to the smartphone and can be passed to the user via a Bluetooth command to the I-Cane Mobilo[1]www.i-cane.org) device as a steering command to the embedded tactile arrow.

2.4. Evaluation

The main goal of the IMAGO project is a precise image-based positioning of a person with a camera. Therefore, the evaluation of the system components had to adjust to this goal.

[1](

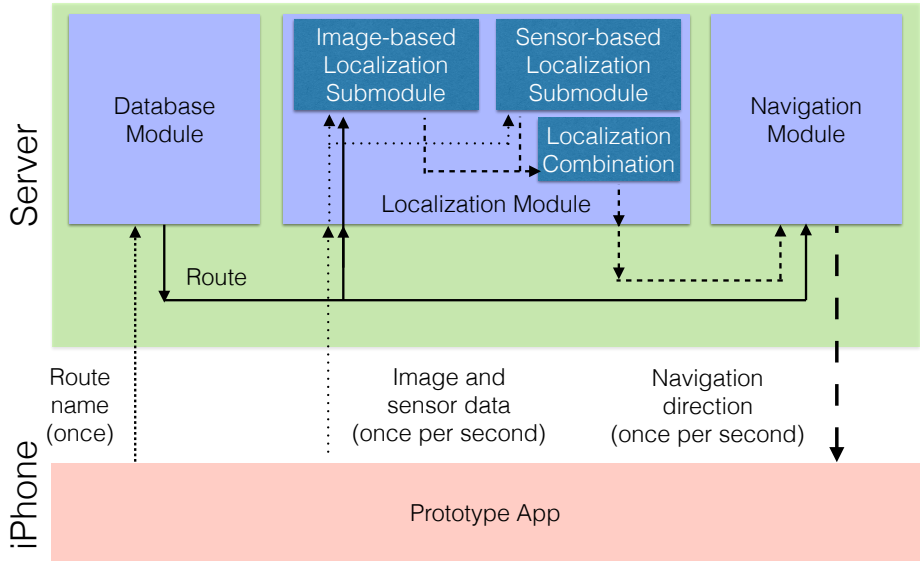


Figure 6. Workflow of image-based localization and navigation

2.4.1. Data

Evaluation was performed on a dataset recorded at the downtown of Maastricht, the Netherlands. The dataset results from 7 walks with a recording device (iPhone 5 with acquisition application running on it) attached with a chest mount utility to the body of the person acquiring images. Within a walk an image was sequentially acquired every second. A total of 3291 images were recorded. Recordings were acquired on several dates, daytimes and weather conditions.

The route passes by several landmarks in the center of Maastricht. The main characteristics of the location are a large number of pedestrians, high vehicle traffic, narrow streets and houses located close to the road. Additionally, the route changes most during spring and summer as street cafes are active, and numerous shops and stores constantly change decorations in and around showroom windows.

Processing with VisualSFM resulted in a dataset consisting of 17 separate models. Each model represents a reconstructed set of building walls as a sparse 3D point cloud. The models contain from 200 to 12792 points.

2.4.2. Localization and outlier removal

In order to show whether outlier removal has indeed a positive impact on image-based localization, outlier removal was incorporated into the testing pipeline. Localization performance associated with models before outlier removal and after was investigated.

For evaluation, a model from the dataset allowing for the best automatic alignment to the real world coordinates was selected as a reference. This model was then reconstructed again by 10-fold cross-validation: all images used in the reference model were randomly partitioned into 10 sub-samples of equal size. For each new reconstruction, a newly selected single subsample containing 10% of original images was used as test data, the remaining 90% of images were used to reconstruct a model.

To test the positioning performance, the following sequence of steps was applied to each test reconstruction:

1. Align each model to the map to estimate their scaling factors relatively to the real world coordinate system.
2. Align the test reconstruction to the reference reconstruction. For that, the estimated scaling parameters are applied to the test and the reference models. Roughly estimate translation between the models by calculating the difference between the models' centroids.

3. Refine translation and rotation by applying the iterative closest point (ICP) algorithm [30].
4. Estimate a position of each test image not used for the reconstruction and record matching time.
5. Use the corresponding positions of the reconstructed images from the reference model to estimate the localization error of each image. The error is calculated as the distance between the estimated position and the reference position in 2D (as the user is localized in 2D, the z-component is omitted).
6. Apply outlier removal to the aligned test reconstruction.
7. Repeat the three previous steps with the resulting models.

Efficiency and quality indicators are distinguished that should be balanced.

Efficiency indicators refer to performance in terms of time and space and estimate the parameters matching time T_m (in seconds) and model size S_m (in KB) accordingly. In order to show the changes in performance caused by the application of outlier removal, the parameters for changes in matching time ΔT_{m0j} and space requirements ΔS_{m0j} are defined as

$$\Delta T_{m0j} = \frac{T_{m0} - T_{mj}}{T_{m0}} \times 100\%, \tag{1}$$

$$\Delta S_{m0j} = \frac{S_{m0} - S_{mj}}{S_{m0}} \times 100\%, \tag{2}$$

where $j = 1$ and $j = 2$ correspond to a model before and after outlier removal respectively.

To measure matching time, the localization experiment was conducted 10 times on each of the test cases. All tests were performed on a single core of a computer equipped with the Intel Core i7 CPU running at 2.00 GHz.

Quality indicators, i.e. matching rate R and matching error E, describe localisation performance associated with a certain model. Finally, based on these two indicators, the weighted matching error E_w is estimated , which is used as an ultimate indicator for the quality of localization.

Let n be a total number of test images associated with a certain tested model. Given a test image contained in the reference model, an image is considered as *matched* if it is possible to reconstruct its position p in the tested model. Accordingly, n_m is the total number of matched images in the model. A match is considered as a *correct* match if the positioning error, estimated as a distance between a reconstructed position p and its corresponding position p_0 in the reference model, is less than a threshold τ:

$$\|p_0 - p\| < \tau. \tag{3}$$

We set $\tau = 1.6$ m (2-3 human steps).
The number of correct matches n_c is estimated as

$$n_c = \sum_{i=1}^{n_m} [\|p_{0i} - p_i\| < \tau]. \tag{4}$$

The matching rate R is then calculated as the ratio of the number of correct matches n_c and the total number of images n:

$$R = \frac{n_c}{n} \times 100\%. \tag{5}$$

The matching error E is the average value of all positioning errors of the correct matches:

$$E = \frac{\sum\limits_{i=1}^{n_m} \|p_{0i} - p_i\| \left(\|p_{0i} - p_i\| < \tau \right)}{n_c}. \tag{6}$$

The weighted error E_w is computed as

$$E_w = wE, \tag{7}$$

where w is the corresponding weighting coefficient of a certain model.

For each j-th model in a test case (a model before and a model after outlier removal) the coefficient w_j is calculated as follows:

$$w_j = 1 - \frac{R_j - \min\{R_1, R_2\}}{100\%}. \tag{8}$$

In fact, the ICP alignment of a test model to the reference model might contain an error up to 1 m. Thus, the absolute values of localization measurements might not be precise. As the same alignment within a test case is used for all cases, the correct estimate of relative error is possible. For measuring the impact of outlier detection on the quality of localization, the final quality indicator is

$$\Delta E_{w0} = E_{w0} - E_w, \tag{9}$$

where E_{w0} is the weighed localization error associated with the model before outlier removal, and E_w is the corresponding weighed error in localization using the model after outlier removal applied.

Student's t-test is applied to the entire sample of positioning errors to see whether the changes in positioning performance are significant or not.

2.4.3. Model alignment

The correctness of model alignment was estimated on the Maastrict Downtown dataset. Transformation parameters were automatically estimated for each model in the dataset. Next, based on visual inspection, the alignments were classified as correct and incorrect. For all incorrect alignments, manual adjustments were proposed, if possible.

Manually adjusted models were considered as the ground truth, and the alignment error for each parameter (translation, rotation and scaling) was estimated as the difference between automatically estimated values and ground truth values.

Additionally, an error of initial alignment with respect to GPS positions was estimated in the same way (for the models with correct alignment, automatically estimated parameters were taken as the ground truth).

3. Results

3.1. Localization and outlier removal

On average, the proposed method reduced the models' sizes about 10.2%. According to visual inspection, the proposed method is able to reduce noise while preserving model structure (Fig. 4). Comparing to the original models containing sparse outliers, the outcomes of outlier removal methods look clean. Some wall fragments containing only a few feature points might be missing but the basic structure is always preserved. Outlier removal itself is performed in linear time. For a model consisting of about 10,000 points, it takes less than 1 s (Fig. 7).

The positioning experiment has shown that reduction of outliers indeed leads to improvement in matching time and has a positive impact on model's size comparing to the performance associated with a model before outlier removal. The benefits in matching time $\Delta T_{m0j} = 8.8\%$ and storage requirements $\Delta S_{m0j} = 10.1\%$ are proportional to the amount of points P_r removed from the model.

The probability to locate an image with a precision of 1.6 m was 80% and 74% for the models before and after outlier removal respectively. Using this threshold, the absolute error values on average were 0.53 m and 0.51 m respectively. Taking into account the matching rate, the relative weighted localization error ΔE_{w0} indicated the loss of positioning quality of 1 cm. Student's t-test conducted on the entire sample of positioning errors classified this difference as not significant ($p = 0.288$).

3.2. Model alignment

Out of 17 models, it was possible to classify the automatic alignment of 3 models as correct. The alignment succeeded in the cases having either a small initial alignment error (translation 3.24 m, rotation 7.3°, and scaling 0.07 in the best case), or having a clearly defined model structure (e.g., points on both sides of the road, corners, intersections are present in the model) in spite of a relatively big initial alignment error (translation 32.5 m, and scaling factor 0.53 in the worst case).

For two of the models, which alignment was classified as incorrect, a manual adjustment was possible. In both cases, the initial translation errors were relatively big (36.2 m and 38.95 m compared to estimated ground truth), the structures were not descriptive enough to provide enough cues for the alignment, and the OSM outlines had a repetitive structure, what additionally constrained the alignment procedure.

For another two models, only rough estimation of the ground truth parameters was possible. One of those models had a clearly defined structure and a relatively small translation error (5.32 m on rough estimation). It turned out that the model was built incorrectly on the reconstruction stage. The second model was not descriptive enough, representing only straight walls and not providing enough constraints for the estimation of scaling parameter.

It was impossible to estimate the ground truth parameters for five other incorrectly aligned models. These models were typically too small and non-descriptive, containing only short walls or wall segments. The GPS readings of these models were inconsistent within the walks, providing completely different alternatives for initial alignment. Moreover, two of those models were recorded while walking through a passage building. That would typically lead to a GPS signal loss.

The alignment procedure could not be applied in the remaining five cases. The models were too small and sparse to identify their footprints (the biggest number

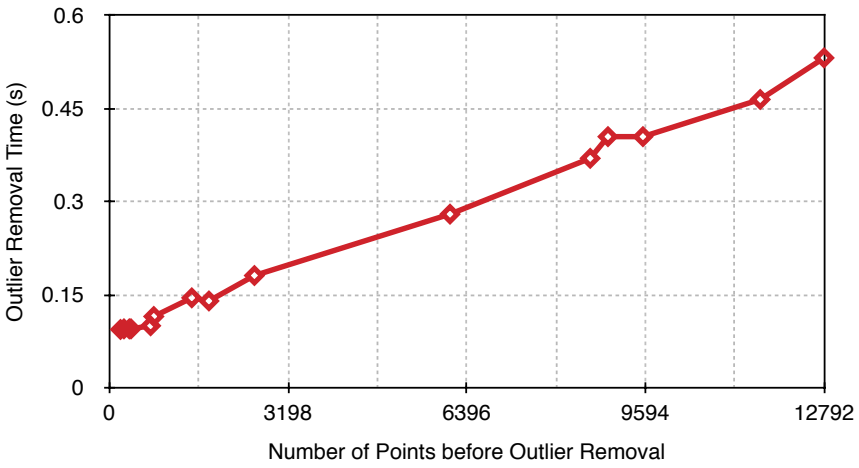


Figure 7. Runtime performance of outlier removal. Outlier removal was applied to 15 different-sized models from the dataset.

of points was 388 and the smallest 43), and the number of cameras (the biggest number was 14 and the smallest 3) were not sufficient to estimate similarity transformations.

4. Discussion

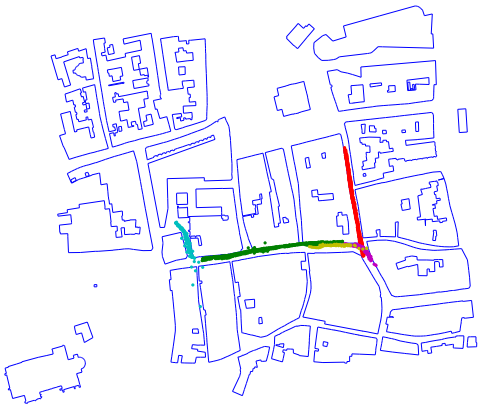
The results have shown that image-based localization achieves a significantly higher positioning precision than the one reached by modern consumer-level GPS sensors. Average error of localization is 0.51 m. This value additionally accumulates an error gained in the process of alignment to the reference model, which cannot be extracted from the final result. Comparing the achieved results to the average GPS error of 34 m, we consider the positioning with an error of 0.51 m (less than 1 human step) as reliable.

The relative weighted localization error indicated the loss of positioning quality of 1 cm due to outlier removal. Such a small loss is acceptable for insignificant considering benefits in terms of matching time and space requirements for the task of city-scale navigation. The Student's t-test conducted on the entire sample of positioning errors confirms the conjecture classifying those losses as outlier removal has a positive impact on image-based navigation. While outlier removal as deployed in this work has the potential to remove non-static parts of the environment located away from buildings, additional care has to be taken regarding changes in the facades. Changes in color (due to painting) can be eliminated through the right choice of feature descriptors, large billboards or commercial signs posted directly onto the walls remain a challenge. One option is the incremental build of the models over time using newly collected images from the VIP navigating along the routes. If keypoints on building facades are not seen over a certain amount of time, these could be eliminated from the model. Similarly, if a new keypoint is found "behind" another keypoint along the optical axis, the older keypoint can be removed as it is no longer obstructing the new one.

An additional method that can also reduce the numbers of matches needed, and thereby further decreasing the time used for localization, is a pruning of the search space. This will be achieved by reducing the points to an area within a certain range around the most likely position (e.g., based on prior position and trajectory). A careful evaluation will be needed to investigate the tradeoff between positioning accuracy and matching time. An iterative approach with a growing region around the estimated position could also be possible, as the most expensive calculation is the matching process. Using the direction from which a point can be seen could also be used to further reduce the number of eligible points.

As the evaluation of model alignment has shown, some models are initially reconstructed wrong, therefore no alignment can be created using the implemented framework. Such models require non-rigid transforms. Incorporation of outlier removal into the bundle adjustment could be one option. Iteratively applying outlier removal after each new image might decrease the number of erroneously reconstructed models. Since images cannot be more than 2 m away from each other, bundle adjustment can be further improved by a priori knowledge. Detecting the images which are

(a) Footprints of different point clouds encoded in color



(b) Camera path based on aligned models

(c) Camera positions based on camera-GPS

Figure 8. Comparison of aligned models.

not congruent with the movement of the person can help here.

Repetitive environments as often found in rural areas or indoors pose an additional challenge. However, the proposed system aims to improve GPS-based positioning in urban areas. GPS can be used in rural areas where the precision is sufficient. We cannot recommend the proposed technique however for indoor environments. Tests within a public building (hospital) did not result in a useable 3D model. Additional landmarks applied to walls or the floor could help to improve the model creation for public buildings but would require a massive investment and is therefore not desireable.

In general, even though it was possible to align only a small number of models, combined together they cover a sufficient portion of the recordings (Fig. 8(a)). The combined camera path (Fig. 8(b)) subjectively appears more precise than the initial GPS path (Fig. 8(c)). Incremental model growth might fill in the gaps in the path and resolve the alignment ambiguity in some cases so that manual adjustments will not be necessary. Another reason of model alignment failures is lacking information about the structure (e.g., the models representing single walls without additional corners and crossroads). Required changes in the acquisition will be also addressed in the future work.

Furthermore, the alignment with OSM is flat and does not consider geographical height. That might cause problems for alignment in the environments with sharp height changes. Alignment to digital elevation models (DEMs, data.geocomm.com/dem/) would solve this problem. Information provided by DEMs might also add an additional constraint for better estimation of the scaling parameter. Additional incorporation of altitude data on the acquisition stage would be required to make this improvement possible.

However, automatic model alignment, or in general the creation of one coherent model, remains the most challenging part of the reconstruction [1,13,22]. The problem is amplified by the need for a user-friendly image-acquisition and consumer hardware. Future research should take this pitfall into account.

The major bottleneck of the navigation is the time spent on the communication between mobile device and the server. 3G and 4G technologies on average achieve the upload speed of 1.6 mbit/sec and 12.4 mbit/sec respectively. Our experiments have shown that the current system requires a bandwidth of approximately 12 mbit/sec, which can be achieved with 4G broadband speed. We intend to shift computational complexity towards the smartphone and, thus, reduce

the required bandwidth in order to support the usage of 3G technology. A comparison experiment involving the study of different feature extraction methods (SIFT, SURF, ORB, BRISK) performed on a mobile device has shown that the required bandwidth can be reduced by a factor of 40.

5. Conclusion

The IMAGO project aims at improving the independence and social mobility of VIP. The proposed system yields a high positioning accuracy and is thus usable for pedestrian navigation. Additionally, the recording of routes and navigation can be performed on consumer-grade hardware (i.e. smartphones) to allow for crowdsourcing and easy dissemination of the navigation software. The combination with an intelligent white cane guides VIP unobtrusive.

It is necessary to investigate user's reaction on the system's performance in terms of tolerance for waiting time and positioning error. This will be addressed in a future user study involving VIP end-users as well as seeing persons. The study will have to prove navigational efficiency as well as "soft" factors such as usability, compliance and acceptance.

Additionally, transfer of complete images through mobile networks is creating a bottleneck. To improve performance, future work will focus on offloading more and more computational work to the smartphone. As a first step, the feature extraction could be performed at the user side and only a small subset of features could be transferred, thereby reducing the amount of data to be transmitted and transmission time. This would however require different features as SIFT features are computationally expensive and don't necessarily reduce transmission size. The final stage of the navigation software should be run completely on the smartphone to allow offline navigation once a 3D route model is transferred from a database.

6. Acknowledgements

The project is funded by the European Union, Framework 7, Ambient Assisted Living joint programme, project number 16SV5846.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in a day. In *Proceedings of the 12th International Conference on Computer Vision*, ICCV'09, pages 72–79, Washington, DC, USA, 2009. IEEE Computer Society.

[2] A. Aladrén, G. López-Nicolás, L. Puig, and J. J. Guerrero. Navigation assistance for the visually impaired using rgb-d sensor with range expansion. *IEEE Systems Journal*, PP(99): 1–11, 2014.

[3] J. Bitsch, P. Smith, N. Viol, and W. K. Footpath: Accurate map-based indoor navigation using smartphones. In *Proceedings of the 2011 International Conference on Indoor Positioning and Indoor Navigation*, 2011.

[4] A. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052–1067, June 2007. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1049.

[5] F. Dellaert, S. M. Seitz, C. E. Thorpe, and S. Thrun. Structure from motion without correspondence. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 557–564, 2000.

[6] R. W. Emerson and D. Sauerburger. Detecting approaching vehicles at streets with no traffic control. *Journal of Visual Impairment and Blindness*, 102(12):747–760, 2010.

[7] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer. The user as a sensor: Navigating users with visual impairments in indoor spaces using tactile landmarks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 425–432, New York, NY, USA, 2012. ACM.

[8] V. Filipe, F. Fernandes, H. Fernandes, A. Sousa, H. Paredes, and J. Barroso. Blind navigation support system based on microsoft kinect. In *4th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion, DSAI 2012*, volume 14, pages 94–101. Elsevier, 2012.

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

[10] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 368–381, Berlin, Heidelberg, 2010. Springer-Verlag.

[11] F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, February 1969.

[12] H. Huang, M. Schmidt, and G. Gartner, editors. *Spatial Knowledge Acquisition in the Context of GPS-Based Pedestrian Navigation*. Springer-Verlag Berlin Heidelberg, 2012.

[13] A. Irschara, C. Zach, M. Klopschitz, and H. Bischof. Large-scale, dense city reconstruction from user-contributed photos. *Computer Vision and Image Understanding*, 116(1):2–14, Jan. 2012. ISSN 1077–3142.

[14] N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. Munich. The vslam algorithm for robust localization and mapping. In *Robotics and Automation*, pages 24–29, 2005.

[15] S. Kef, J. Hox, and H. Habekothé. Social networks of visually impaired and blind adolescents. structure and effect on well-

being. *Social Networks*, 22(1):73–91, 2000.

[16] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, pages 392–403, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[17] H. Leppäkoski, J. Collin, and J. Takala. Pedestrian navigation based on inertial sensors, indoor map, and wlan signals. *Journal of Signal Processing Systems*, 71(3):287–296, 2013.

[18] N. Li and B. Becerik-Gerber. Performance-based evaluation of rfid-based indoor location sensing solutions for the built environment. *Advanced Engineering Informatics*, 25(3):535–546, 2011.

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2): 91–110, 2004.

[20] M. Modsching, R. Kramer, and K. ten Hagen. Field trial on GPS accuracy in a medium size city: The influence of built-up. In *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*, pages 209–218, Hannover, Germany, 2006. IEEE.

[21] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *Proceedings of the International Conference on Computer Vision*, pages 667–674, 2011.

[22] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *Proceedings of the ACM SIGGRAPH 2006*, pages 835–846, New York, NY, USA, 2006. ACM.

[23] C. Strecha, T. Pylvaenaeinen, and P. Fua. Dynamic and scalable large scale image reconstruction. In *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 406–413, June 2010.

[24] B. Streckel and R. Koch. Lens model selection for visual tracking. In W. Kropatsch, R. Sablatnig, and A. Hanbury, editors, *Pattern Recognition*, volume 3663 of *Lecture Notes in Computer Science*, pages 41–48. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-28703-2.

[25] S. Thrun and J. J. Leonard. *Springer Handbook of Robotics*, chapter Simultaneous Localization and Mapping, pages 871–889. Springer-Verlag Berlin Heidelberg, 2008.

[26] O. Untzelmann, T. Sattler, S. Middelberg, and L. Kobbelt. A scalable collaborative online system for city reconstruction. In *Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 644–651, Dec 2013.

[27] K. M. Varpe and M. P. Wankhade. Visually impaired assistive system. *International Journal of Computer Applications*, 77 (16):5–10, September 2013.

[28] World Health Organization. Visual impairment and blindness. Fact Sheet 282, World Health Organization, 2014.

[29] C. Wu. Towards linear-time incremental structure from motion. In *Proceedings of the 2013 International Conference on 3D Vision*, 3DV '13, pages 127–134, Washington, DC, USA, 2013. IEEE Computer Society.

[30] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, October 1994.