International Journal of Computer Vision
https://doi.org/10.1007/s11263-018-1093-3

Efficiently Annotating Object Images with Absolute Size Information Using Mobile Devices

Martin Hofmann[1] · **Marco Seeland**[1] · **Patrick Mäder**[1]

Received: 28 July 2017 / Accepted: 18 April 2018

Abstract
The projection of a real world scenery to a planar image sensor inherits the loss of information about the 3D structure as well as the absolute dimensions of the scene. For image analysis and object classification tasks, however, absolute size information can make results more accurate. Today, the creation of size annotated image datasets is effort intensive and typically requires measurement equipment not available to public image contributors. In this paper, we propose an effective annotation method that utilizes the camera within smart mobile devices to capture the missing size information along with the image. The approach builds on the fact that with a camera, calibrated to a specific object distance, lengths can be measured in the object's plane. We use the camera's minimum focus distance as calibration distance and propose an adaptive feature matching process for precise computation of the scale change between two images facilitating measurements on larger object distances. Eventually, the measured object is segmented and its size information is annotated for later analysis. A user study showed that humans are able to retrieve the calibration distance with a low variance. The proposed approach facilitates a measurement accuracy comparable to manual measurement with a ruler and outperforms state-of-the-art methods in terms of accuracy and repeatability. Consequently, the proposed method allows in-situ size annotation of objects in images without the need for additional equipment or an artificial reference object in the scene.

Keywords Size annotation · Size measurement · In-situ size annotation · minimum focus distance · Absolute size · Mobile device

1 Introduction

Cameras allow us to create persistent copies of real-world objects. Unfortunately, the projection to a planar image sensor inherits a loss of major information about the 3D structure of the scene and especially its absolute dimensions. However, knowing an object's absolute dimensions can be beneficial for manifold image analysis tasks, like categorization, measurement, and classification. Especially for fine-grained classification problems, such as plant species identification (Wittich et al. 2018), size information can improve accuracy significantly. Today, size and depth information can only be captured along with image data by either utilizing specific equipment, e.g., stereo cameras or laser range sensors; or by adding a reference object of known dimensions, such as a ruler, to the scene. Reference objects are a frequently utilized approach, especially in standardized environments. However, to facilitate precise measurements, the reference object needs to be positioned in the same image plane as the object of interest and needs to be aligned in parallel to the object. This makes the approach often imprecise and cumbersome, especially in the field (Rzanny et al. 2017). Furthermore, a depicted reference object along with the actual scene can negatively impact image processing and analysis steps, e.g., image classification (Wäldchen and Mäder 2018).

In this paper, we present an efficient process allowing to precisely annotate objects in images with their absolute size information without utilizing additional hardware or placing reference objects in the scene. The entire process runs on commodity smart mobile devices. The approach is built on the fact that with a camera calibrated to a specific object dis-

Communicated by V. Lepetit.

✉ Martin Hofmann
martin.hofmann@tu-ilmenau.de

Marco Seeland
marco.seeland@tu-ilmenau.de

Patrick Mäder
patrick.maeder@tu-ilmenau.de

[1] Technische Universität Ilmenau, Ilmenau, Germany

Published online: 24 May 2018

tance, lengths can be measured within this specific object's plane. We use the camera's minimum focus distance as calibration distance and show that it can be accurately retrieved by humans. Facilitating measurements on larger object distances, we contribute an adaptive feature matching process allowing for precise computation of the scale change between two images. In summary, our contributions are:

- An efficient method for annotating object sizes while acquiring images with commodity mobile devices.
- An adaptive feature matching process precisely computing scale change between two images.

This paper is structured as follows. In Sect. 2, we give an overview on related work and review the methods relevant for our process. In Sect. 3, we introduce our approach and discuss its steps in detail. Section 4 discusses experiments to compare different configurations of the proposed algorithms for accuracy. In Sect. 5 we report on a user experiment to study the measurement error of the proposed approach in a measurement situation and in comparison to other approaches. In Sect. 6, we discuss our results and in Sect. 7 we conclude and outline future research.

2 Related Work

The size of an object of interest within an image can be measured by calculating the distance between points defining the external dimensions of the very object in the image plane. However, the outcome of this measurement will be in units of pixels. There exists no method for immediate object size measurements from images acquired by a monocular camera. Methods that are extensible in this regard span geometrical scene understanding, scene reconstruction, and visual odometry. Below, we review those methods that are in principle capable of measuring metric object sizes from one or a series of images. Furthermore, we review and compare methods relevant for the individual steps of our proposed approach.

2.1 Determining Metric Scale

In general, a reference scale is required to convert between pixel and metric dimensions measured within images. This reference scale can be measured actively or derived passively (Criminisi et al. 1999). Active measurement uses additional sensors, e.g., ultrasonic, laser range finders, or structured light, to acquire the reference each time a measurement is taken. We consider active devices too cumbersome for the envisioned object annotation scenario since the required additional sensors are not yet part of commodity mobile devices. A viable solution are passive references that are either directly or indirectly derived from a taken image.

Direct references are objects of known size that are placed in a captured scene. If the distance between camera and reference object is the same as the distance to the object of interest, the scaling between pixel and metric size can be computed based on the known dimensions of the reference object. If the reference object and object to be measured are not placed in the same image pane, geometrical scene understanding is required. Criminisi et al. (2000) developed a series of algorithms using projective geometry to acquire geometrical scene understanding for such cases. In detail, geometric cues such as planarity of points and parallelism of lines and planes are used as scene constraints for inferring a geometrical scene understanding and allowing to measure object sizes from a single image containing a reference object. We consider direct references not practical as they require additional user interaction in placing and annotating a reference object of known dimensions in every image. More convenient are indirect references regarding the distance between the object to be measured and the camera. These can be derived from accurate knowledge about camera parameters (Criminisi et al. 1999). For example, Mustafah et al. (2012) propose object depth and size computation from the disparity of a calibrated stereo camera with known alignment. Depth-From-Focus (DFF) analyzes an image sequence taken by altering the depth of field and estimating depth from the best focused points in each image (Nayar and Nakagawa 1994). However, the approach requires a large set of images acquired with precisely controlled camera parameters and computing sharpness across the sequence is computationally expensive and is reported to take several seconds even on GP-GPUs (Moeller et al. 2015).

In contrast, Depth-From-Defocus (DFD) estimates the amount of defocus between two or more images of a static scene (Pentland 1987). An early variant of DFD is the spatial-domain convolution-deconvolution transform (STM) (Subbarao and Surya 1994), which uses two images of a static scene taken at the same camera location, but with varying camera parameters, such as lens position, focal length, and aperture diameter. The camera needs to be calibrated experimentally and the photographer needs control over camera parameters and location for the approach to be applicable. The applicability of the STM approach is highly dependent on the scene's texture. Watanabe and Nayar (1998) improved DFD approaches by applying rational filters that are invariant to texture and lighting but require telocentric optics. Kuhl et al. (2006) evaluate the characteristic blur scheme of a specifically calibrated optic to deliver RGB-D information of a scene. Levin et al. (2007) demonstrated that a specially coded aperture delivers a distinct blur pattern that can then be transformed into depth information. More recent improvements were achieved by using video evaluation (Kim et al. 2016) that contains different focal levels. Lin et al. (2013) estimated depth from a single defocused image and pro-

posed to segment an image by level of defocus applying a sequence of aperture-shape filters and then smoothed the depth map based on a boundary-weighted belief propagation algorithm. Despite increased accuracy and faster execution, the approach requires several hundred seconds on conventional CPUs even for small images with < 0.15 Mpx. In general, DFD methods either lack accuracy or depend on highly specific hardware, such as telocentric optics and a coded aperture. A major problem of every DFD approach is that its accuracy decreases with increasing distance to the scene, since the unfocus blur range depends on this distance.

Researchers also propose machine learning approaches for estimating depth within images. Eigen and Fergus (2015) used a convolutional neural network to predict depth, surface normals, and semantic labels from single images. The authors report an absolute relative difference of 15.8% with respect to the ground truth depth for the indoor scene dataset NYUDepth. Uhrig et al. (2016) used a fully convolutional neural network trained on semantic labels and depth imagery for improved instance segmentation in urban street scenes. For 5% of the segmented and correctly classified instances of the Cityscapes dataset and 14% of the KITTI dataset, their network predicted depth with $\leq 25\%$ error. While machine learning approaches deliver increasingly promising results in terms of accuracy, they require a sufficient amount of scene-specific training data and are then restricted to comparable scenes.

In contrast to image scene understanding and depth estimation, Photogrammetry subsumes methods that recover positions of surface points from 2D image sequences in order to reconstruct 3D scenes (Luhmann et al. 2006). Seminal work in this area are Structure from Motion (SfM) (Koenderink and Doorn 1991) and Simultaneous Localization and Mapping (SLAM) (Smith and Cheeseman 1986). Both approaches compute the 3D structure of a scene from point observations (images) and intrinsic camera parameters. SfM reconstructs scenes from sets of images with arbitrary order. The complete graph describing a scene is optimized simultaneously. SfM algorithms were successfully scaled to work on very large problems, e.g., reconstructing cities from thousands of disordered images captured using different cameras (Agarwal 2009). SLAM on the other hand can be seen as real-time variant of SfM that is capable of localizing and tracking a camera's position and orientation and is of special importance for applications in robotic mapping and navigation (Nitzan 1985). SLAM relies on an ordered sequence of images acquired using a fixed camera setup (Cadena et al. 2016). Given time and computational resource constraints, only a limited number of keyframes can effectively be utilized. Research therefore often focuses on the precise reduction of the number of necessary keyframes (Cadena et al. 2016).

SfM and SLAM allow to calculate 3D coordinates of sparse as well as dense points in scenes and to localize the camera along with its pose within the reconstructed scene. Given such a 3D model, absolute lengths could theoretically be measured therein. This requires to use a reference object of known size or, in case of SLAM, to track the trajectory in absolute coordinates. Initially, SLAM approaches required a calibrated stereo camera setup with additional sensors like laser range-finder, inertial measurement units (IMUs), or sonar (Thrun 2002; Fuentes-Pacheco et al. 2015). However, more recently visual SLAM (vSLAM) approaches have been demonstrated to work with a single monocular camera (Karlsson et al. 2005; Davison et al. 2007). Parallel tracking and mapping (PTAM) by Klein and Murray (2007) is one of the first real-time applications of vSLAM. Loop closing, i.e., recognizing previously visited location visually or by sensors and updating coordinate estimations of the point cloud accordingly, was found beneficial for increasing the accuracy of scene reconstruction and self-localization (Williams et al. 2009). Mur-Artal et al. (2015) propose ORB-SLAM, a real-time monocular vSLAM approach combining best-practices, i.e., effective methods for keyframe selection, feature matching, point triangulation, per frame camera localization and re-localization in combination with scale-aware loop-closing and covisibility information. They used ORB features simultaneously for tracking, mapping, re-localization and loop-closing, enabling real-time operation. ORB-SLAM2 improves the effectiveness of the approach and makes it also suitable for RGB-D and stereo cameras (Mur-Artal and Tards 2017). Whereas the absolute scale of the map cannot be resolved by monocular vSLAM methods, ORB-SLAM2 in conjunction with a stereo or RGB-D camera allows for absolute scaling of the map. However, scale calibration using a reference object is still required for monocular cameras. Absolute scaling can also be achieved by fusion of visual and inertial measurements (Robertson et al. 2013; Cadena et al. 2016). This visual-inertial SLAM (viSLAM) was shown to yield improved performance in terms of accuracy, resolving the scale ambiguity and avoiding scale drift. For example, Leutenegger et al. (2015) reported $< 2\%$ translation error on the ETH Bicycle Trajectory and Main Building dataset using custom-built visual-inertial sensor hardware. Today, viSLAM is the sole technology behind commercial augmented reality frameworks, such as Google's ARCore (Google Inc. 2017) and Apple's ARKit (Apple Inc. 2017). While being a substantial improvement over previous approaches, viSLAM suffers from two major drawbacks: (1) a camera must have a clear light-of-sight under well-lit environments at any time, and (2) the continuous processing of video data implies high power consumption and quickly drains the battery of mobile devices.

2.2 Local Feature Matching

One contribution of our proposed approach relies on accurately computing the scale change between two images of the same object taken at different camera distances. For computing this scale change, we utilize local features co-occurring in both images. Feature matching, i.e., finding corresponding features in a sequence of images, is a well known computer vision problem. Plenty of methods exist for detection of stable features as well as for robust extraction of discriminative descriptors. We are specifically interested in stable features, i.e., blob- and corner-like image regions that can be re-identified despite changes in image scale, orientation, lighting geometry, and noise, while possessing enough discriminative power for feature matching (Tuytelaars and Mikolajczyk 2008; Seeland et al. 2017). Since our process evaluates their relative locations, precise and repeatable localization of corresponding features in the image plane is most important. Aanæs et al. (2011) systematically investigated this property purely on geometrical constraints and for a wide set of state-of-the-art feature detectors on a comprehensive dataset of complex non-planar scenes. They found the Harris corner detector (Harris and Stephens 1988) to achieve the highest recall rates (ratio of feature matches to total amount of features) during changes in the horizontal viewing angle and lighting geometry. Since the Harris corner detector is a fixed scale detector, the authors observed a large drop in performance for scale change factors exceeding 1.4. For such scenarios, the scale-invariant Difference-of-Gaussians (DoG) (Lowe 2004) and the Hessian-based detectors (Mikolajczyk and Schmid 2004) outperformed the Harris detector and were found to yield stable local features despite overall lower recall rates compared to the Harris detector (Aanæs et al. 2011). Moreels and Perona (2006) investigated the performance of various feature detectors and descriptors in matching 3D object features across 144 different viewpoints and three lighting conditions. They found a combination of a Hessian based detector with Scale-invariant Feature Transform (SIFT) descriptor to be most robust on a database of 100 different 3D objects. Reviewing performance evaluation studies of feature detectors and descriptors, also Li and Allinson (2008) concluded that SIFT is generally accepted as a most robust and discriminative descriptor. Several extensions and algorithmic improvements of SIFT were proposed over the past years. The most notable contributions are PCA-SIFT (Ke and Sukthankar 2004), RootSIFT (Arandjelovic and Zisserman 2012), and DSP-SIFT (Dong and Soatto 2015). Given the success of neural networks in object recognition and detection, neural networks were also used for learning local descriptors. Schönberger et al. (2017) performed extensive comparison of learned and hand-crafted local features with respect to their feature matching performance for image-based reconstruction. They found hand-crafted feature descriptors, namely DSP-SIFT and RootSIFT (Bursuc et al. 2015), to outperform learned features in terms of precision, recall, and match ratio. Piasco et al. (2018) surveyed methods for visual based localization, including local features, and concluded hand-crafted descriptors like SIFT to perform best in most scenarios. In addition, hand-crafted descriptors are still faster to compute, despite running on the CPU (Schönberger et al. 2017). Current implementations of learned descriptors require GPU hardware and are currently not practical for resource limited devices such as mobile phones. Based on these studies and observations, we consider the advanced DSP-SIFT in combination with its DoG detector (Dong and Soatto 2015) as one possible technique for feature matching. As the Harris corner and the Hessian-based SURF detector were found to yield more stable image regions compared to DoG, we also study their matching performance in combination with the RootSIFT descriptor (further denoted as SIFT) in our study.

3 The CamMeter Approach

We propose an approach for annotating images with size information consisting of two major processes. The calibration process, illustrated in Fig. 1, aims at determining the pixel size coefficient of a camera at a user-reproducible calibration distance. This is typically a one-time process per user and device. The actual measurement process, illustrated in Fig. 2, consists of retrieving the calibration distance on the object of interest, computing the relative scale change to facilitate measurement of larger objects requiring more distant imaging, and finally annotating the object with its size information.

3.1 Calibration Process

To allow for size measurements, the camera of the mobile device must be calibrated once. During this calibration, the

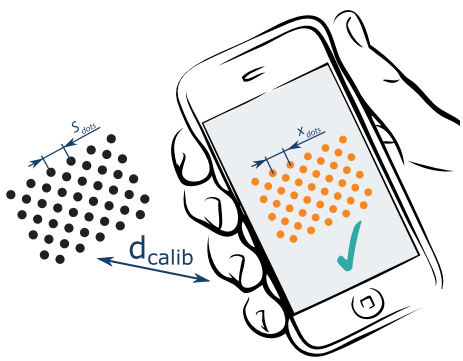

Fig. 1 Calibration process of the proposed approach. Calibration is performed by taking an image of a reference dot pattern at the shortest focusable object distance

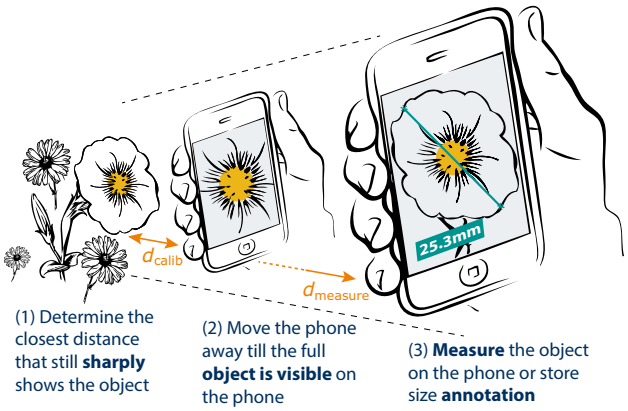

Fig. 2 The CamMeter measurement process

pixel scaling coefficient ξ_{calib}, i.e., the metric size of an image pixel in the object plane, is determined. Therefore, a user takes calibration images of a reference dot pattern with known size and geometry. These images are then automatically evaluated. The dots of the pattern are detected with a blob detector and filtered based on their roundness and convexity to make sure the pattern is detected correctly. The averaged pixel distances between neighboring dots' centroids $\bar{x}_{\text{dots}}$ are then used to compute a camera's pixel scaling coefficient as

$$\xi_{\text{calib}} = \frac{s_{\text{dots}}}{\bar{x}_{\text{dots}}} \tag{1}$$

where s_{dots} is the known physical distance between dots in the printed version of the pattern. For being able to utilize ξ_{calib} within the measurement process, it is required that the user takes an initial image of the object of interest at the same distance d_{calib} that the camera has been calibrated at. Hence, it is required that d_{calib} is well defined and can be found in a reproducible and convenient manner by the user. We propose using a camera's minimum focus distance, i.e., the shortest possible distance between object and camera where the object can be mapped sharply.

A user can easily identify this distance by moving the camera slowly away from the object until the object becomes focusable by the camera's autofocus function for the first time. To improve measurement accuracy, the calibration process is repeated by the user several times until the statistical variance in ξ_{calib} becomes low, i.e., $< 5\%$ of ξ_{calib}. Thereby, we assure that the user has learned to reproducibly find this specific distance by the end of the calibration process. A user does not need to find the exact minimum focus distance of a given camera. Instead, she or he should be able to take images of the calibration pattern as well as of the objects of interest at the same distance. Finding the user specific minimum focus distance is a matter of individual perception and we evaluated the ability to reproducibly find this individual distance in a user study.

3.2 Measurement Process

After determination of ξ_{calib} a user may perform size measurements within images taken at the d_{calib} distance. However, larger objects cannot be mapped to the camera at their full extent if the image is taken at d_{calib} (see Fig. 2), but require an increased distance d_{measure} to the object. Hence, a second image at distance $d_{\text{measure}} > d_{\text{calib}}$ is taken, this time capturing the whole object of interest (cp. Fig. 2). ξ_{calib} is then transformed to ξ_{measure} by computing the change in scale between the two images taken at d_{calib} and d_{measure}. The scale change is derived from scale-invariant features co-occurring in both images as described below.

3.2.1 Scale-Invariant Feature Detection and Matching

Considering two images of an object from the same perspective but different distances, we utilize the pattern of local features co-occurring in both images for computing the change in scale. To make this approach applicable, the local features primarily need to be invariant with respect to changes in scale, but also to slight distortions in perspective and illumination while possessing sufficient discriminative power to allow for unambiguous matching. As detailed in Sect. 2, DSP-SIFT (Dong and Soatto 2015) as well as a combination of the Hessian-based detector (Mikolajczyk and Schmid 2004) and the SIFT descriptors (Arandjelovic and Zisserman 2012) meet these requirements. For our implementation, we chose the Hessian-based Speeded-up Robust Features (SURF) detector by Bay et al. (2008). This detector achieves a five-time speed-up over DoG (Tuytelaars and Mikolajczyk 2008) by using a fast approximation of the Hessian matrix based on a set of box-type filters (Bay et al. 2008). Given the high recall rates of the Harris corner detector (Aanæs et al. 2011), we also included the Harris-SIFT in our study and experimentally evaluated the accuracy of these detector-descriptor combinations under different image perturbations.

The two sets of local features F_{calib} and F_{measure} are then matched against each other, i.e., for every feature in F_{calib} a corresponding feature in F_{measure} is searched. A corresponding feature from F_{measure} is considered identified if its associated descriptor is most similar to the query descriptor taken from F_{calib}. Figure 3 shows examples of matching features. We use the Euclidean distance as similarity measure, following a L1 normalization of each SIFT descriptor, and a square-rooting of its elements. This algebraic extension termed RootSIFT allows for improved retrieval performance compared to the original SIFT descriptors (Arandjelovic and Zisserman 2012).

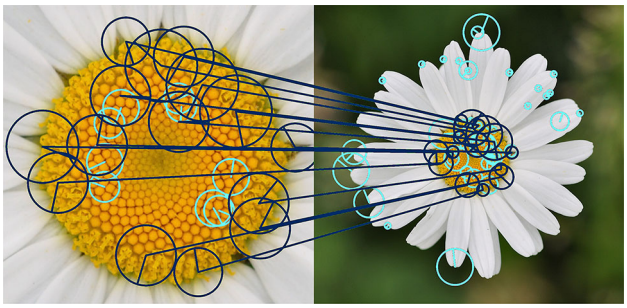

Fig. 3 Example of local features co-occurring in two related images of the same object. Green circles denote matched features and lines connect counterparts in both images, whereas red circles denote unique features not matched in the other image. Please note that only a fraction of the total number of features is displayed to improve visibility (Color figure online)

The number of detected features depends on an image's resolution and its content. Large images typically produce several thousand features, thereby significantly affecting the time necessary for descriptor extraction and for matching on a mobile device. At the same time, a sufficient number of matching features is required for accurate determination of the scale change. To investigate this tradeoff, we conducted a series of experiments evaluating the error of the scale change computation upon systematically altered image resolutions.

The set of matching features F_{matches} is filtered in three stages to reject invalid matches. First, F_{matches} is filtered based on Lowe's ratio test (Lowe 2004). In detail, the ratio of the Euclidean distances in descriptor space of the first and the second-best match per feature must be below $\text{RT}_1 = 0.80$ to be classified as valid match (Lowe 2004). Second, the geometry of the remaining features is verified. By affine transformation, a homography is computed for projecting the positions of F_{calib} onto the positions of F_{measure}. A RANSAC algorithm is used to filter outliers not matching the homography. Third, remaining invalid matches are filtered using a ratio test on the remaining valid features with the threshold $\text{RT}_3 = 0.775$. The value of $\text{RT}_3 = 0.775$ has been identified by experiment (cp. Sect. 4.2.1).

3.2.2 Scale Change Computation

The set of valid matches F_{matches} is now used to compute the scale change between the images taken at d_{calib} and d_{measure}. We compute a homography of the image pair and decompose it to extract the change in scale. We consider this method our baseline and refer to it as *SURF-SIFT Homography Decomposition*.

As an advanced approach, we introduce our method *S*cale *Cha*nge from *F*eature *D*istance Ratios (SChaFD). First, we identify correlations in F_{matches} and use them for sorting F_{measure} according to the order in F_{calib}. Next, we compute pairwise Euclidean distances d_{ij} for the positions of the n valid features in F_{matches} in the image plane. The results are distance matrices D per image in the format

$$D = \begin{pmatrix} d_{0,0} & \cdots & d_{0,n} \\ \vdots & \ddots & \vdots \\ d_{n,0} & \cdots & d_{n,n} \end{pmatrix}. \tag{2}$$

The distance ratio matrix S can now be computed by Hadamard division of the distances matrices D_{calib} and D_{measure} as

$$S = D_{\text{calib}} \oslash D_{\text{measure}}. \tag{3}$$

As S stores pairwise distance ratios s_{ij}, a multitude of relative scale values are computed, i.e., one entry for every pair of corresponding features in F_{matches}. Given a planar object, perfectly aligned in parallel to both image planes, the distribution of s_{ij} ideally follows a Dirac-function. However, mainly two factors induce broadening of the distribution: (1) most objects are 3D or exhibit non-planar structures, and (2) the images are not acquired on the same optical axis, resulting in translation and perspective transformation. Therefore, we extract the median of s_{ij} as combined relative scale change $s_c = \tilde{s}_{ij}$. Combining DSP-SIFT, SURF-SIFT as well as Harris-SIFT with the proposed algorithm, we study in detail the methods *DSP-SIFT-SChaFD*, *SURF-SIFT-SChaFD*, and *Harris-SIFT-SChaFD* respectively.

Whereas the SURF and the DoG detectors are scale-invariant by design, the Harris corner detector is a fixed scale detector showing a drop in performance for scale changes larger than 1.4 magnification (Aanæs et al. 2011). Using the Harris-SIFT combination, we perform the steps (1) feature detection and matching, and (2) scale change computation in an iterative manner. In each iteration, the photograph taken at d_{calib} is resized based on the computed scale change to match the physical resolution of the target image at d_{measure}. If no matches are found, the image at d_{calib} is downsized to fit 80% of its previous size. The idea is based on the findings of Aanæs et al. (2011) that studied the Harris detector in-depth and found that it reaches its best recall of detected features among two images when their scale differs between 0 and 30%. We end the downsizing process once matches are found or when the image is downscaled to 4% of its initial size. We refer to this method as *iterative Harris-SIFT-SChaFD*. The Harris corner detector has been found to achieve higher recall rates and to be computationally less expensive than DSP-SIFT and the SURF-SIFT combination (Aanæs et al. 2011) and therefore promises an efficient iterative computation. However, in order to validate this previous finding regarding our research problem, we also study the *iterative DSP-SIFT-SChaFD* method.

3.2.3 Object Segmentation and Annotation

Eventually, the measured object is segmented from the color image and an annotation XML file in PASCAL VOC format is being created (Everingham et al. 2010). The bounding box is depicted based on the corners of the smallest rectangle enclosing the convex hull of the object mask. The diameter in millimeters is stored along with the bounding box. The GrabCut (GC) algorithm (Rother et al. 2004) is used for interactive object segmentation. GC is based on iterated graph cuts (IGC), an algorithm evaluated as most accurate and time-effective for interactive image segmentation (McGuinness and OConnor 2010; Peng et al. 2013). GC provides three improvements over IGC: (1) GC segments color images using Gaussian Mixture Models instead of grayscale histograms; (2) GC iterates between foreground-background estimation and parameter learning, replacing the one-shot min-cut estimation of IGC; and (3) GC allows for incomplete labeling, eventually reducing the amount of user interaction required for accurate segmentation (Rother et al. 2004). To speed up the segmentation process on a mobile device, the image at d_{measure} is resized to a maximum of 600 px at the largest side while maintaining the aspect ratio. The original GC is initialized by a rectangle loosely placed around the object of interest by the user. However, we use the convex hull of the positions of the SChaFD features to compute a saliency map as prior for initializing the segmentation. The user is then allowed to iteratively refine the mask by adding scribbles marking either foreground or background. The binary mask image depicting only the area of the object of interest is then resized to the original image size followed by boundary smoothing.

3.2.4 Mobile Application

We have implemented the proposed approach for mobile devices that either run Android or iOS. Feature detection and descriptor extraction is performed using OpenCV, an open-source computer vision library (Bradski 2000). If no pixel scaling coefficient is found on the device, the user has to perform the calibration process as detailed in Sect. 3.1. The user is then guided trough the steps of our process (cp. Fig. 2). For all images at minimum focus distance, the focus of the camera is automatically set to the shortest distance and kept locked. Autofocus is enabled for all other images. A magnification of the area around a user-defined point-of-interest, augmented into the camera stream, helps the user in determining the sharpness of the point of interest. Scale change computation is done using the iterative Harris-SIFT-SChaFD approach.

3.3 Measurement Limitations

Our approach relies on feature matching in images with different scales. Local features can only be detected if the object of interest possesses sufficient texture and structure. A blank surface of homogeneous color, e.g., a white wall, will not yield sufficient features allowing for scale change computation. Highly self-similar objects, i.e., objects with lots of repetitions in their structure and texture, likely cause confusion during feature matching. Furthermore, the object itself has to be static, i.e., it must not change or move during acquisition of the image pairs. There is also an upper limit in scale change that can be computed from images with a finite resolution. SIFT features require local patches of at least 4×4 px. Geometrical verification through homography estimation requires at least four local features, but nine features result in a much more robust solution against outliers. Assuming a non-overlapping 3×3 arrangement of feature points, the minimum required image size is 12 px at either side. Our process resizes the image taken at minimum focus distance iteratively in order to match the scale of the more distant image. Assuming an up-to-date 12 MP smartphone camera capturing images with a resolution of 3000 px at the smaller side, a scale change of up to 250 between two images would theoretically be supported. These cameras typically allow capturing objects of up to 40 mm in size at minimum focus distance. Hence, the theoretical upper limit for object sizes measurable by our approach is 10 m. The practically measurable size, however, is more likely to be 5 m due to arbitrary feature padding and distribution as well as space around the object within the image plane. This limitation could be mitigated by taking intermediate images while moving away from minimum focus distance. This approach either requires more user interactions for taking additional images while moving away from the minimum focus distance or a SLAM-like solution, which is known to be computationally expensive and slow.

4 Process Experiments

To investigate the scale computation properties of the proposed algorithm, we performed a series of controlled experiments on publicly available datasets. We specifically investigated the effect of different image pair perturbations, i.e., distinct scale changes and angular displacement as well as the overall scene planarity on the accuracy of the methods introduced in Sect. 3.2. As measure for the algorithms performance we evaluated the median relative error (*MRE*) on the respective scenes. This measure is not the algorithm's

average error but a measure to compare different settings by quality. We chose this measure because of the non-normally distributed error and to suppress the influence of low and high outliers.

Before investigating the accuracy under such perturbations, we performed initial experiments aiming for optimization of our process. First, we studied how the filtering stages (see Sect. 3.2) affect the accuracy of our method. Second, we examined the effect of image resolution. On one hand, reducing the image resolution results in increased relative sharpness and reduced noise. On the other hand, image details are lost and coarse structures become more prominent. Furthermore, image resolution affects the reliability and accuracy of the measurement process in terms of image pairs at different scales.

4.1 Experimental Setup

Since, we aimed to investigate the impact of absolute scale change, angular displacement and scene planarity on the quality of the computed scale change, we needed a controlled setting (or dataset) that keeps other factors influencing the measurement accuracy constant. Furthermore, we needed ground truth information accompanying this dataset, e.g., in terms of camera coordinates or homography matrices. We found two datasets meeting these requirements: the *Zoom Sequences*[1] published by Mikolajczyk and Schmid (2004) and the *Point Feature Dataset*[2] recorded by Aanæs et al. (2010).

The *Zoom Sequences* consist of six scenes with high planarity, such as paintings, as well as scenes far away from the camera, such as mountains. Each scene is represented by 4–21 images with highly parallel image planes covering scale changes in the range from 1 to 5.5. For this dataset, we computed the scale change between the image at the closest distance and the remaining images at increasing distance. Scale changes were computed with the SURF-SIFT Homography Decomposition, the DSP-SIFT-SChaFD, the SURF-SIFT-SChaFD, and the Harris-SIFT-SChaFD approach. In addition, DSP-SIFT-SChaFD and Harris-SIFT-SChaFD were also utilized in the proposed iterative procedure (see Sect. 3.2). For every image pair, the *MRE* of the scale change was calculated by comparing the algorithms' results against the respective ground truth scale change computed from the homography delivered along with the dataset.

The *Zoom Sequences* cover large changes in scale but show diminishing and rather uncontrolled displacement from the optical axis. In contrast, the *Point Feature Dataset* allows us to examine the influence of nonplanar scenes with well defined angular displacement. This dataset consists of roughly 135 k images from 60 highly nonplanar scenes, each recorded from 119 positions and under 19 different illumination conditions. The position of each frame was precisely controlled by mounting the camera on an industrial robot. The images of the *Point Feature Dataset* came with a width of 800 px. Camera calibration matrices for the 119 camera positions were supplied with the dataset. Based on these matrices we reconstructed the original camera coordinates in 3D space (cp. Fig. 4) and used them as ground truth for computing the *MRE* of our scale computation process.

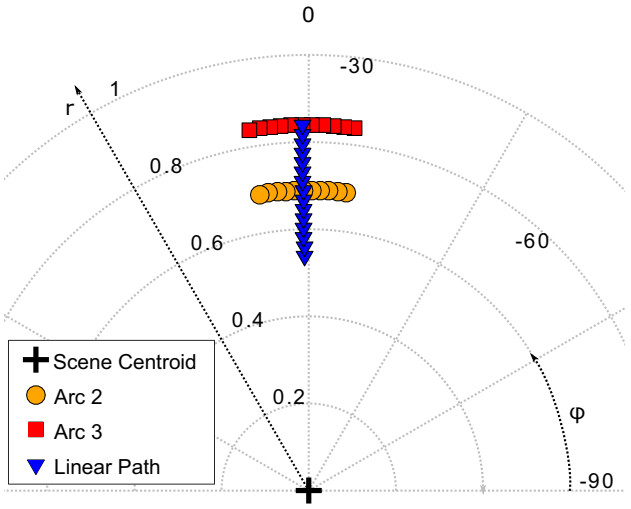

Fig. 4 Camera positions for the images of the *Point Feature Dataset* in polar coordinates ($r[m]$ and $\phi[°]$. The notation follows the original notation of Aanæs et al. (2010)

To investigate the behavior under distinct angular displacement of up to $\pm\phi = 7°$, we computed the scale change between the images acquired at the arcs and one image acquired at the central position on the linear path (cp. Fig. 4). This procedure was repeated for all central positions on the linear path at the distances $d_{\text{Linear Path}} \leq d_{\text{Arc}}$. This resulted in a grid of 10×7 data points for 'Arc 2' (ten angular displacements at seven scale changes) and 10×14 data points for 'Arc 3' (ten angular displacements at 14 scale changes). For every point on these grids, we computed the *MRE* of the scale change using the iterative Harris-SIFT-SChaFD across all 60 scenes and all 19 illuminations conditions of the *Point Feature Dataset*.

4.2 Results

The subsections below present evaluation results on the parameterization of the proposed feature filtering, on the impact of image resolution on scale change computation, on the accuracy for increasing scale changes, and on the influence of non-planarity and angular displacement on scale change computation accuracy.

[1] https://thoth.inrialpes.fr/people/mikolajczyk/Database/zoom.html.

[2] http://roboimagedata.compute.dtu.dk/?page_id=24.

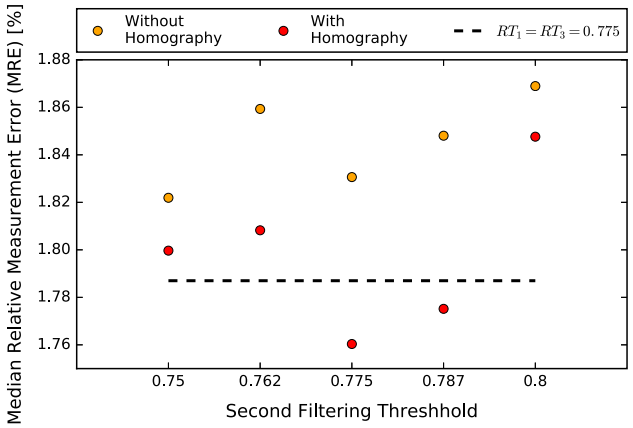

Fig. 5 The influence of the third filtering stage on the *MRE* of the scale changes computed on the *Zoom Sequences* using iterative Harris-SIFT-SChaFD. Orange (red) circles display results with (without) homography filtering. The dark red line represents $RT_1 = 0.775$ (Color figure online)

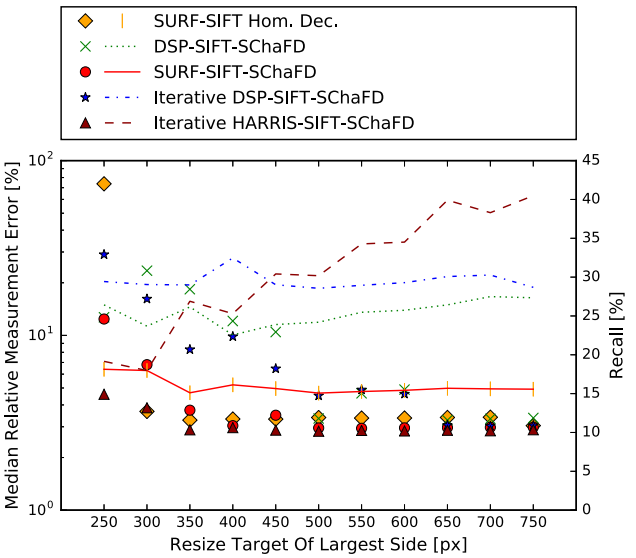

Fig. 6 The effect of image resolution on the *MRE* for the five scale change computation methods SURF-SIFT Homography Decomposition, (iterative) DSP-SIFT-SChaFD, SURF-SIFT-SChaFD, and iterative Harris-SIFT-SChaFD. The lines show the average recall per method

4.2.1 Optimizing Feature Filtering

We investigated how the filtering stages (see Sect. 3.2) affect the accuracy of the proposed scale change computation method. In addition to Lowe's observations (Lowe 2004), we found empirically that filtering detected features with a homography in the second stage and an additional ratio threshold RT_3 in a third stage can improve accuracy of scale change computation by removing weak matches. Therefore, we conducted an experiment on the *Zoom Sequences* dataset and varied the introduced ratio threshold RT_3 of the third stage from 0.75 to 0.8 in steps of 0.05 with and without filtering by homography (see Fig. 5). We did not consider threshold values lower than $RT_3 < 0.75$ since (Lowe 2004) demonstrated that it would eliminate more correct matches than false ones. Furthermore, at $RT_3 \geq 0.8$ no additional filtering would occur since such matches are already removed upon the first filtering stage with $RT_1 = 0.8$. We found the lowest *MRE* of 1.76% at $RT_3 = 0.775$, which we therefore use as default threshold value for the next experiments. We found that directly setting $RT_1 = 0.775$ rather than introducing the additional RT_3 is disadvantageous (cp. $RT_1 = RT_3 = 0.775$ in Fig. 5). We also found that homography filtering resulted in the lowest *MRE* and applied it by default for the next experiments. Since the features depend on the dataset, none of the chosen parameters might be optimal for all scenes. However, the *Zoom Sequences* contain a variation of different scenes and we found our results to be consistent across these different scenes.

4.2.2 The Impact of Image Resolution

We studied the impact of image resolution on the scale change computation error. Each image of the *Zoom Sequences* was downscaled by bilinear interpolation to resolutions from 250 to 750 px on the widest side with 50 px increments. Images originally smaller than the target size were not enlarged, i.e., images of the scenes "Asterix" and "VanGogh" were not evaluated for $x > 500$ px. The majority of image pairs yield scale changes smaller than three. In order to equally aggregate results without bias towards such moderate scale changes, we aggregated the image pairs into scale change bins of width 1 and computed the *MRE* per bin. Figure 6 shows median *MRE* in relation to image resolution. We found that the iterative Harris-SIFT-SChaFD outperforms all other methods at image sizes from 250 to 750 px in terms of *MRE*, with the best result of 2.82% achieved when downscaling images to 500 px. Results of the iterative Harris-SIFT-SChaFD method vary only by 1.7% in terms of *MRE* for resolutions between 250 and 750 px demonstrating its robustness against varying image resolutions. At the maximum resolution of 750 px, all methods achieve close results with the iterative Harris-SIFT-SCHaFD performing slightly better than the others. Especially, DSP-SIFT features result in worse *MRE* for image sizes < 500 px, which we attribute to less stable features detected by DoG compared to the Harris and the Hessian-based SURF detector used by the other approaches (Moreels and Perona 2006; Aanæs et al. 2011). Applying DSP-SIFT iteratively improves *MRE*. We also studied the recall of the feature matching process and found the iterative Harris-SIFT-SChaFD method to yield higher recall of retrieved keypoints compared to SURF-SIFT, DSP-SIFT and iterative DSP-SIFT (cp. Fig. 6). Therefore, we decided to use the iterative Harris-SIFT-SChaFD for further experiments.

4.2.3 Non-planar Scenes and Angular Displacement

Our proposed approach typically requires a user to take one image at short distance and another at larger distance capturing the object at its full extent. In practice, both images will not exactly be on the same optical axis causing the image pairs to be subject to translation, rotation as well as perspective distortion. We conducted another experiment to study the influence of these factors on scale computing accuracy. Given the findings in the previous experiment, we rescaled images to a maximum of 350 px at their widest side. Features were filtered in three stages, i.e., with $RT_3 = 0.775$ and the iterative Harris-SIFT-SChaFD was used for computing scale changes.

Figure 7 shows that the *MRE* increases with increasing angular displacement and scale change. However, the impact of angular displacement decreases with increasing scale. Comparing the *MRE* at the greatest angular displacements of Arc 2 and Arc 3, the *MRE* doubles at Arc 2 while it only increases by 20% in Arc 3. The reason could be a hidden correlation with the drop of recall rate which is smaller for Arc 3 than for Arc 2 at the greatest angular displacement as found by Aanæs et al. (2011). For Arc 3, the highest *MRE* of 2.65% is observed at an angular displacement of 1° (cp. Fig. 7b). The results for Arc 2 show a comparable behavior, i.e., a local maximum in the *MRE* at only slight angular displacements. This observation contradicts a correlation with recall rate described for this dataset by Aanæs et al. (2011). We consider nonlinear changes in perspective to be the main cause for the observed effect. However, the interesting fact for this study is that for all scale changes in Fig. 7 the *MRE* ultimately increases upon increasing angular displacement. Despite the angular displacement and scale changes, the largest observed *MRE* is $\approx 2.5\%$, which is comparable to the *Zoom Sequences* dataset and very low given the complex 3D scene structure and distinct illumination conditions of the Point Feature dataset. We decided to perform user studies to evaluate the actual measurement accuracy in a practical setting.

5 User Experiments

The following user experiments were designed for (1) evaluating reproducibility of finding the SFOD distance; for (2) evaluating the overall accuracy of our approach under real measurement conditions and in comparison to alternative manual methods; and for (3) evaluating the measurement accuracy of our approach in comparison to other automated solutions. Given the time required for measuring a large number of objects with multiple methods, we decided to perform two consecutive experiments. The first experiment evaluates reproducibility of finding the SFOD and compares the proposed approach with manual measurement methods. The second experiment evaluates the approach in comparison to other automated applications capable of measuring object size.

5.1 Experimental Setup

For the experiment, we randomly selected twenty participants that had not seen or used CamMeter before. These participants were students and employees of the Technische Universität Ilmenau and were between 23 and 35 years old. Ten out of the twenty participants had eye problems and needed to wear glasses with $-$ 6.5 to 1.5 diopters, a fact that we considered potentially relevant for the applicability of the approach.

Our experiment had four independent variables chosen to assess whether the proposed CamMeter approach facilitates

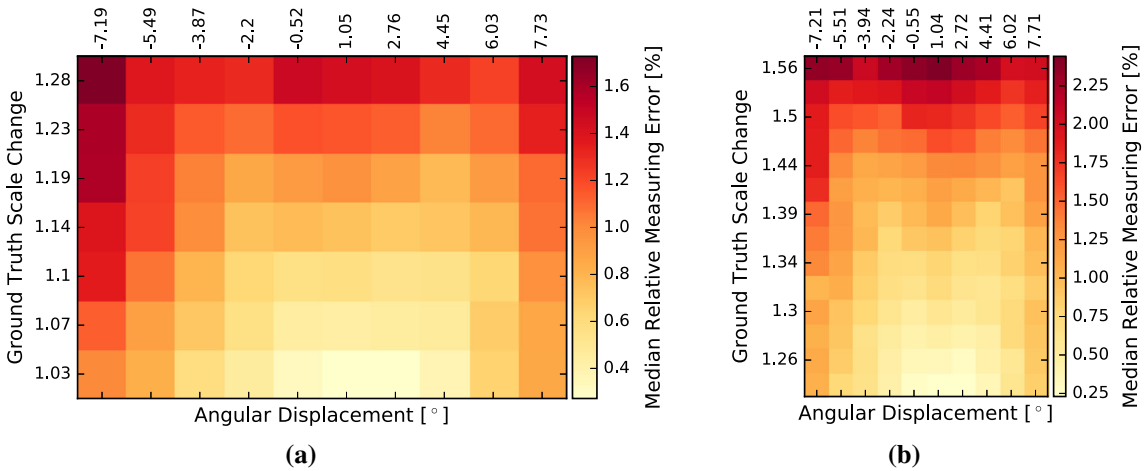

Fig. 7 *MRE* of the scale change computed using the iterative Harris-SIFT-SChaFD for different angular displacements and scale changes. The median was calculated for images on **a** Arc 2 and **b** Arc 3 across all 60 scenes and all 19 illumination conditions of the *Point Feature Dataset*

measurements that are comparable to alternative methods used for acquiring object size information. These variables are: the measurement method, the measured object, whether the participant had a visual disorder, and the participant themselves.

We selected three *measurement methods* for the first experiment: (1) using a traditional ruler (Manual), (2) holding a reference object of known dimensions at the measurement plane into the image (Reference), and (3) using the proposed approach in the configuration determined in the previous section (CamMeter). We selected another three *measurement methods* for the second experiment: (1) using the state-of-the-art vSLAM approach (ORB-SLAM2) (Mur-Artal and Tards 2017), (2) using a viSLAM approach realized with Apple's ARKit (Tape Measure), and (3) again using the proposed approach in the configuration determined in the previous section (CamMeter). We selected the two automated solutions since they have the same preconditions as our approach, i.e., using a monocular camera and acquiring indirect reference with sensors available in commodity mobile devices, and may be utilized for the same purpose with some adaptation. ORB-SLAM2 is a state-of-the-art vSLAM approach (Mur-Artal and Tards 2017) with publicly available source code executable on iOS (ygx2011 2017). We extended the available implementation in two aspects: (1) to support a scale calibration analogous to our approach, and (2) to allow a user to select two measurement points on the camera stream and to return the Euclidean distance between them. Apple's ARKit as well as Google's ARCore are libraries that utilize latest viSLAM approaches. For selecting a prominent and accepted application based on those libraries, we queried the Apple App Store with the search string "('Photo' OR 'Camera' OR 'Tape' OR 'AR') AND ('Meter' OR 'Measure' OR 'Ruler')". Out of the 553 returned apps, we kept those explicitly mentioning 'AR', 'ARKit', or 'Augmented Reality' in their description. We identified Tape Measure (AppStoreId 1271546805) as the currently most popular app based on normalized average rating (four stars) and number of ratings (2891).

We selected twelve *objects to be measured* having different dimensions and showing a rich variation in the properties: planarity, gloss, and self-similarity (cp. Figs. 8, 9 and Table 1). The selected objects are non-deformable and have static shape, allowing for reproducible size measurements. The coin is small enough to be measured without the necessity for scale change computation during CamMeter measurements. The round tangerine makes it hard to find the right measuring plane. The sole has a complex self-similar texture potentially complicating the local feature matching and the scale change computation. The artificial flowers have a high divergence in appearance. The dahlia is non-planar and has a cluttered surface with some degree of self-similarity. The sunflower has an irregular shape, high self-similarity, and its petals are in another plane than the stamens. The PCB exposes a low self-similarity, is partly non-planar due to the elements on the board, has clear edges, and a glossy surface. The second set of six objects resembles object typically occurring in urban scenes and that are frequently part of benchmark datasets, e.g., the MIT-CSAIL dataset (Torralba et al. 2004) and the RGB-D object dataset (Lai et al. 2011). The bottle has a glossy surface with high transparency, the only exception being its label. The can has a cylindrical shape and a shiny label. The door is highly planar, but large in size

Table 1 Measured objects (first and second set) rated for planarity, gloss of the surface, and self-similarity

Measured object *o*	Planarity	Gloss	Self-similarity
Two-Euro *coin*	☆☆☆	☆☆	☆
Artificial *tangerine*	☆	☆☆	☆☆☆
Shoe *sole*	☆☆☆	☆☆	☆☆
Artificial *dahlia*	☆	☆	☆☆
Artificial *sunflower*	☆☆	☆	☆☆
Printed circuit board (*PCB*)	☆☆	☆☆☆	☆
Room *door*	☆☆☆	☆☆	☆
Cereal *box*	☆☆☆	☆☆	☆
Lemonade *bottle*	☆	☆☆☆	☆☆
Soccer *ball*	☆	☆☆	☆☆
Can of soup	☆	☆	☆
Computer *keyboard*	☆☆	☆	☆☆☆

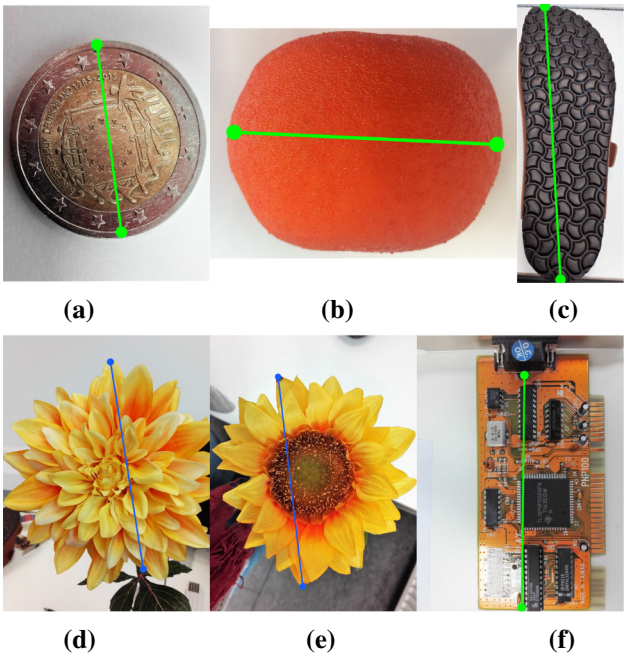

Fig. 8 First set of objects with two measuring marks each, in between which, participants had to measure the distance with the three methods: Manual, Reference, and CamMeter. **a** Coin, **b** tangerine, **c** sole, **d** dahlia, **e** sunflower and **f** PCB

Fig. 9 Second set of objects with two measuring marks each, in between which, participants had to measure distance with the three methods: ORB-SLAM2, Tape Measure, and CamMeter. **a** Door, **b** box, **c** bottle, **d** ball, **e** can and **f** Keyboard

and feature-less except for an image of postcard dimensions in the middle. The keyboard is highly self-similar. The cereal box is planar and has a glossy, but well textured surface. The ball is considered to be among the most complicated objects given its spherical shape, glossy surface, and high self-similarity.

We considered participants' *visual disorder* as potentially affecting the results and therefore recorded whether they were wearing glasses or not.

We studied the following dependent variables for the different phases of the experiment. To asses participants' ability in retrieving the minimum focus distance, we measured ξ_{calib} in (mm/px) (see Sect. 3.1). To asses and compare the quality of measurements among all methods, we compute the relative measurement error Δs as

$$\Delta s = \frac{\|s_{\text{measure}} - s_{\text{GT}}\|}{s_{\text{GT}}} \tag{4}$$

with s_{measure} being the measured distance between the marked points and s_{GT} being the correct distance as determined under lab conditions. To compare ease of use and preference across the different methods, we asked participants for their opinion on a 5-step Likert scale.

We applied a four step experimental procedure. Initially, participants were *briefed* through a written tutorial on the goal of the experiment, on how to perform measurements with the respective three methods, and on the experimental steps to perform. In between measurements they were guided by an experimenter that followed a pre-scripted procedure. To suppress uncontrolled influences, all runs of the studies were performed in an office environment with darkened windows. The actual experiments had two phases: in the *first phase*, each participant had to calibrate the same smartphone (first ten participants: Huawei P8 lite, second ten participants: Apple iPhone SE) to its minimum focus distance ten times. In between these ten calibration runs, participants were asked to put the smartphone on the table in order to ensure that the process was repeated entirely each time. In the *second phase* the participant had to use the calibrated smartphone to carry out measurements on one set of objects using three out of the six measurement methods introduced above (2×18 measurements in total). Each measurement object was marked with two colored points indicating the distance to be measured. Eventually, participants filled a *questionnaire* collecting their experiences in finding the correct measurement plane per object and in handling of the different smartphone applications.

5.2 Statistical Analyses

All measured variables were first characterized by descriptive statistics, i.e., min, max, mean, median, and standard deviation. Prior to analyses, we examined data points for normality using Quantile–Quantile plots, one-sample Kolmogorov–Smirnov tests, and Shapiro–Wilk tests. We found all data points to be normally distributed making an analysis of variances (ANOVA) the appropriate means for investigation. One-way ANOVA was performed in prior to a pair-wise post-hoc comparison. The precondition of equality in variances was examined through Levene's test and showed samples to actually have homogeneous variances. The Holm-Bonferriori method was used as post-hoc test. Accordingly, we performed a multifactorial ANOVA analysis to explore the influence of the independent variables: measurement method, measured object, visual disorder, and participant.

5.3 Results

Figures 10 and 11 aggregates participants' results across ten times calibrating the corresponding phone to d_{calib} following the procedure and using the calibration pattern described in Sect. 3.1. The correct dot centroid distance for the used pattern was $s_{\text{dots}} = 5.88$ mm. Per calibration run, we computed the average Euclidean dot centroid distance $\bar{x}_{\text{dots}}$ between all pairs of neighboring dots and determined the pixel scaling coefficient ξ_{calib}. For the Huawei P8 lite, the median of participant's results deviates between a minimum of 0.0294 mm/px for participant 8 and a maximum of 0.0320 mm/px for par-

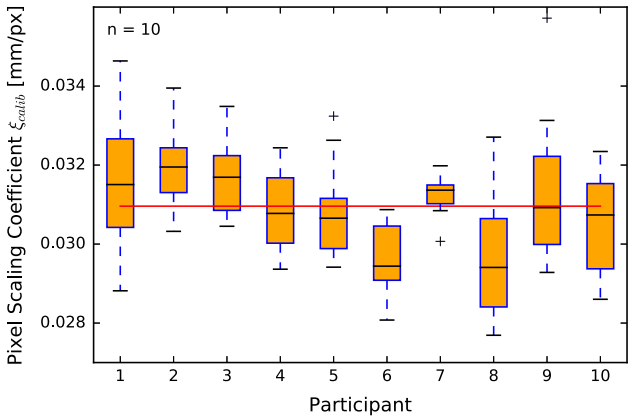

Fig. 10 Distribution of pixel scaling coefficient ξ_{calib} per participant based on ten repetitions of the calibration step. The red line refers to the ground truth value of $\xi_{\text{GT}} = 0.031$ for the utilized Huawei P8 lite (Color figure online)

ticipant 2, translating into minimum focus distances between 58.3 and 62.9 mm. For the iPhone SE, the median of the results deviates between 0.0422 and 0.0509 mm/px, translating into minimum focus distances between 75.2 and 90.8 mm. These values show that a distinction between blurry and sharp is indeed individual per participant. More relevant than the absolute calibration distance, however, is a user's ability in repeatedly retrieving her or his own distance measured as the standard deviation within the ten measurements per participant. We found this standard deviation in ξ_{calib} to be between 1.61% for participant 7 and 5.61% for participant 9 for the P8 lite and to be between 1.74% for participant 12 and 5.36% for participant 11 for the iPhone SE. This translates into an average variance among the individual ξ_{calib}'s of 4.7% for the P8 lite and of 3.11% for the iPhone SE.

To evaluate measurement accuracy of the proposed approach in relation to the manual baseline methods, the first ten participants measured the first six objects with the approaches manual, reference, and CamMeter, thereby creating 180 data points. The next ten participants measuring the the second set of six objects with the three automated approaches ORB-SLAM2, Tape Measure, and CamMeter, creating another 180 data points. Table 2 shows descriptive statistics per object-method combination and across all measurements per method. Column 2 shows the number of data points aggregated per row and column 3 shows the "correct" length to be measured per object (ground truth). The following three column pairs show mean and standard deviation of the absolute measured distances ($s_{m,o}$) as well as relative measurement error determined as relation between the absolute measurement and the ground truth ($\Delta s_{m,o}$) per object-method combination and across all data points per methods.

Additionally, Figs. 12 and 13 provide a visual overview of results as box-and-whisker plot aggregating solely the relative measurement errors $\Delta s_{m,o}$ per measured object-method combination. Manual measurement results are as reported by participants, Reference measurements are derived by referencing the distance between the measurement marks on the object to the known length on the reference object, ORB-SLAM2 and Tape Measure results are as reported by the respective app, while CamMeter results were computed as described in Sect. 3 and configured as described in Sect. 4. We used participants' individual pixel scaling coefficient ξ_{calib} determined in the first phase of the experiment to compute results.

The multifactorial ANOVA analysis revealed that measurement method m significantly impacts the relative measurement error Δs, while the measured object o, participants' visual disorder v and their identity p did not significantly influence results (cp. Table 3). Table 2 shows the results of individual statistical comparisons between the measurement methods as superscript letters 'a' and 'b' next to the individual relative measurement error $\Delta s_{m,o}$. The same letter at two values within one row indicates non-significant differences among them, while different letters indicate significant differences. For the objects Coin, Dahlia, and Sunflower, the measurement errors of the three methods do not significantly differ from one another. For the Dahlia and the Tangerine, the higher measurement error of the Reference method seems to stem from participants having difficulties in finding the correct measurement plane to hold the reference object at. The Sole was more accurately measured with the ruler (Manual method) due to its plane and flat surface. The PCB was more difficult to measure with the ruler since the components on the circuit board made it physically more difficult to align it. We also examined the relative measurement error across all objects per method Δs_m and found the Reference method to perform significantly worse than the manual and the CamMeter method. On average, CamMeter delivers slightly worse results than a manual measurement with a ruler, but these differences are non-significant suggesting that CamMeter facilitates measurement results that are indeed comparable to manual measurement. Contrasting the automated methods (lower part of Table 2), CamMeter delivered the smallest measurement error for the Door, the Box, the Can, and the Keyboard. For the Bottle, ORB-SLAM2 achieved slightly smaller measurement error compared to CamMeter. Tape Measure and ORB-SLAM2 delivered more accurate results for the Ball compared to CamMeter. Finally, we investigated whether a visual disorder, operationalized as a participant wearing glasses or not, influences measurement accuracy. Table 4 shows that we found no significant difference in the relative measurement error between participants with and without glasses.

A qualitative analysis of participants' responses to the questionnaire revealed that handling the reference object simultaneously to the smartphone was very challenging for

Table 2 Absolute measured distances $s_{m,o}$ and relative measurement errors $\Delta s_{m,o}$ per object and measurement method as mean ± standard deviation

Measured object o	n	Ground Truth	Measurement method m					
			Manual		Reference		CamMeter	
		s_{GT} (cm)	$s_{m,o}$ (cm)	$\Delta s_{m,o}$ (%)	$s_{m,o}$ (cm)	$\Delta s_{m,o}$ (%)	$s_{m,o}$ (cm)	$\Delta s_{m,o}$ (%)
Coin	30	2.58	2.48 ± 0.13	$3.98^{a} \pm 4.77$	2.51 ± 0.07	$3.22^{a} \pm 1.34$	2.58 ± 0.09	$2.82^{a} \pm 1.61$
Tangerine	30	5.71	5.97 ± 0.44	$7.21^{ab} \pm 5.00$	5.61 ± 0.84	$11.46^{b} \pm 8.74$	5.79 ± 0.22	$3.13^{a} \pm 2.50$
Sole	30	29.38	29.46 ± 0.12	$0.33^{a} \pm 0.35$	29.24 ± 1.20	$2.73^{b} \pm 2.80$	28.57 ± 1.18	$4.15^{b} \pm 2.26$
Dahlia	30	20.73	21.68 ± 0.78	$4.83^{a} \pm 3.46$	21.29 ± 4.93	$14.16^{a} \pm 18.76$	19.94 ± 1.02	$4.86^{a} \pm 3.71$
Sunflower	30	11.84	11.80 ± 0.38	$2.37^{a} \pm 2.01$	11.38 ± 0.99	$6.39^{a} \pm 6.40$	12.10 ± 0.49	$4.13^{a} \pm 1.95$
PCB	30	11.78	11.87 ± 0.14	$0.78^{a} \pm 1.13$	11.46 ± 0.52	$3.20^{b} \pm 4.06$	11.97 ± 0.62	$4.66^{ab} \pm 2.23$
All Δs_m [%]	180			$3.25^{a} \pm 3.97$		$6.86^{b} \pm 9.78$		$3.96^{a} \pm 2.48$
Measured object o	n	Ground Truth	Measurement method m					
			ORB-SLAM2		Tape Measure		CamMeter	
		s_{GT} (cm)	$s_{m,o}$ (cm)	$\Delta s_{m,o}$ (%)	$s_{m,o}$ (cm)	$\Delta s_{m,o}$ (%)	$s_{m,o}$ (cm)	$\Delta s_{m,o}$ (%)
Door	30	221.65	262.89 ± 81.63	$20.20^{ab} \pm 35.88$	227.12 ± 110.95	$37.44^{a} \pm 30.91$	211.75 ± 12.11	$5.92^{b} \pm 3.62$
Box	30	22.00	24.56 ± 4.59	$12.07^{a} \pm 20.60$	25.55 ± 9.76	$29.96^{a} \pm 35.52$	21.33 ± 1.14	$5.02^{a} \pm 2.99$
Bottle	30	25.68	26.37 ± 1.74	$6.10^{a} \pm 3.54$	21.82 ± 5.76	$21.13^{b} \pm 16.12$	24.99 ± 2.55	$6.53^{a} \pm 7.67$
Ball	30	10.73	10.81 ± 0.81	$5.77^{a} \pm 4.48$	10.47 ± 0.57	$4.48^{a} \pm 3.45$	10.57 ± 1.39	$9.30^{a} \pm 8.59$
Can	30	11.82	12.17 ± 2.11	$11.79^{a} \pm 13.19$	13.22 ± 8.88	$34.32^{a} \pm 67.05$	11.76 ± 0.66	$4.55^{a} \pm 2.94$
Keyboard	30	44.74	41.36 ± 14.80	$22.12^{a} \pm 24.79$	47.01 ± 5.68	$6.68^{ab} \pm 11.86$	43.99 ± 1.80	$3.44^{b} \pm 2.47$
All Δs_m [%]	180			$13.01^{ab} \pm 20.65$		$22.33^{a} \pm 35.43$		$5.88^{b} \pm 5.42$

Different letters indicate significant differences between the measurement methods

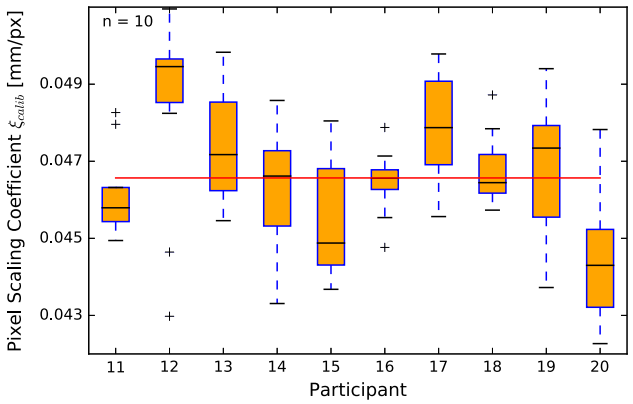

Fig. 11 Distribution of pixel scaling coefficient ξ_{calib} per participant based on ten repetitions of the calibration step. The red line refers to the ground truth value of $\xi_{GT} = 0.0466$ for the utilized Apple iPhone SE (Color figure online)

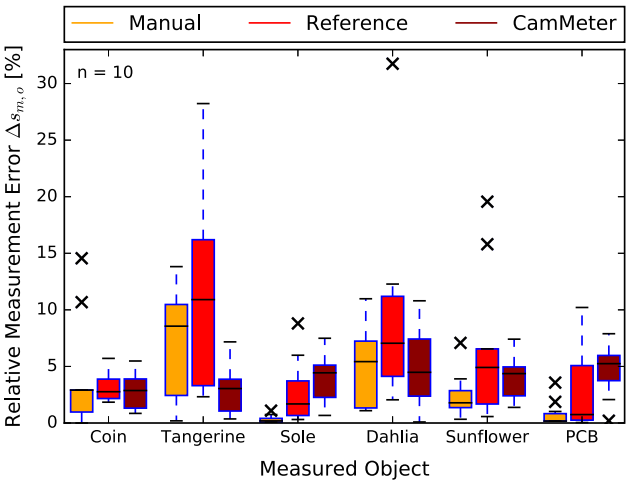

Fig. 12 Distribution of relative measurement error $\Delta s_{m,o}$ for the first set of objects and methods

the majority of participants (Reference method). Participants felt range finding slightly more difficulty using CamMeter compared to the Reference method. No participant reported difficulties in handling the ruler (Manual method). Users reported ORB-SLAM2 to be time-consuming and tedious due to the low framerate (roughly three fps) and since fast movements effect the loss of point features and abortion of the measurement. Though ORB-SLAM2 allows for efficient re-localization, a measurement still needs to be repeated, which participants reported disappointing. The usability of CamMeter and Tape Measure were considered equal and considerably better than that of the ORB-SLAM2 approach. Several participants recognized the large measurement error produced by Tape Measure, especially for vertical distances, e.g., when measuring the door.

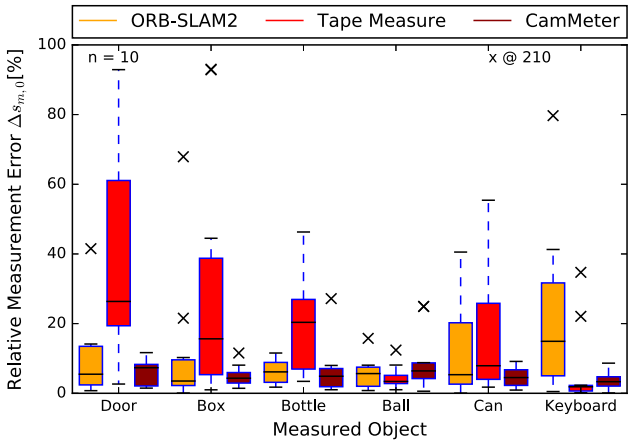

Fig. 13 Distribution of relative measurement error $\Delta s_{m,o}$ for the second set of objects and methods

6 Discussion

Our user study showed that the proposed CamMeter method facilitates measurements during image acquisition with a relative measurement error ($\Delta s_{\text{CamMeter}} = 3.96 \pm 2.48\%$) comparable to the manual measurement with a ruler ($\Delta s_{\text{Manual}} = 3.25 \pm 3.97\%$). At the same time the error is significantly lower compared to photographing a reference object in the measurement plane along with the object ($\Delta s_{\text{Reference}} = 6.86 \pm 9.78\%$). In addition to achieving better results than the Reference method, CamMeter has the benefit of not influencing the image scene with a reference object, which is relevant when training image classifiers. Participants responded in the questionnaire that the Reference method was substantially more laborious than the CamMeter method. Therefore, we consider the proposed method suitable for object size annotation in the field, especially, since there is no additional tool needed beyond a smartphone. In a second experiment, we compared CamMeter to other smartphone applications built on state-of-the-art vSLAM (ORB-SLAM2) and viSLAM (Tape Measure) approaches and exposed all three approaches to a more divers set of objects. We found that CamMeter on average facilitated measurements with higher accuracy and smaller standard deviation ($\Delta s_{\text{CamMeter}} = 5.88 \pm 5.42\%$) than the ORB-SLAM2 ($\Delta s_{\text{ORB-SLAM2}} = 13.01 \pm 20.65\%$) and the Tape Measure ($\Delta s_{\text{TapeMeasure}} = 22.33 \pm 35.42\%$) approaches, though, only the CamMeter and the Tape Measure results differ statistically significantly. We found no evidence that the user and her or his visual abilities do have a significant effect on the achieved measurement accuracy throughout both experiments. However, we found that especially non-planar objects with glossy surface, e.g., the Ball and the Bottle, as well as large texture-less objects, such as the Door, increase CamMeter's measurement error. CamMeter's largest measurement error was found for the Ball ($\Delta s_{CamMeter\bar{B}all} = 9.30 \pm 8.42\%$). Studying the problem in more depth, we found that in addition to its challenging object properties, user had to retrieve the minimum focus distance at a point that was further away from the camera than the closest distance to the object. Several users reported that they had trouble in understanding where to retrieve the minimum focus distance in this setting and we currently discuss how to guide the user in such settings. Contrary to our expectations, we observed no decrease in measurement accuracy for objects with a high degree of self-similarity, e.g., for the Sole and the Keyboard, and attribute this robustness to the effectiveness of geometrical verification within the CamMeter method. Compared to ORB-SLAM2 and Tape Measure, CamMeter yielded the smallest measurement error for the Door, the Box, the Keyboard, and the Can. More importantly, CamMeter showed the smallest overall deviation of measurement errors among the compared methods per experiment, demonstrating high measurement reproducibility.

The accuracy of the proposed approach depends mainly on two factors: (1) users' ability in repeatedly retrieving the minimum focus distance and (2) the accuracy in scale computation. We found that users were able to retrieve their individual calibration distance with low standard deviation, i.e., 4.7% for the Huawei P8 lite and 3.1% for the Apple iPhone SE. Comparing our approach to the other automated solutions, ORB-SLAM2 also requires calibration through direct or indirect reference per measurement and we implemented our proposed calibration process also for the ORB-SLAM2 app to make it suitable for measurements. In contrast, ARKit relies on precisely calibrated inertial sensors of the Apple hardware and does not require any further calibration step per measurement. Regarding scale change computation, we found the iterative Harris-SIFT-SChaFD configuration to be the optimal selection for our approach. An *MRE* of 2.82% for the *Zoom Sequences* dataset indicated the best configuration for scale measurement of plane scenes. Additionally, we tested our algorithm on the *Point Feature* dataset and found that the *MRE* increases with distance and angular displacement. However, the maximum *MRE* remained at 2.5% and the influence of angular displacement decreases with increasing scale. This behavior makes the algorithm robust against human inability in moving the mobile phone on straight lines in space. Our approach has limitations when applied to images of poor quality, i.e., low sharpness, high noise, or large angular displacement. However, due to direct feedback on the success of a measurement a user may easily retake the measurement while still being on the scene. We estimate CamMeter's practical upper object size limit to approximately 5 m. Since SLAM solutions do not need to match point features in every frame of the captured stream a user can move during measurement, limiting the maximum measurable object size only through the length of user's trajectory. However, the scale has been found to drift over longer mea-

Table 3 Multifactorial ANOVA analysis between the independent variables and the relative measurement error

Factor	Df	F-value	$Pr(>F)$	
Measurement method m	4	13.041	7.31^{-10}	***
Measured object o	11	1.497	0.131	$n.s.$
Visual disorder v	1	0.708	0.401	$n.s.$
Participant p	17	0.634	0.864	$n.s.$
Residuals	326	0.0305		

Significance codes for $Pr(>F)$: $\leq .001$ '***', ≤ 0.01 '**', ≤ 0.05 '*'

Table 4 Relative measurement error across all objects for participants wearing glasses and those without for the first (upper part) and the second (lower part) experiment

Visual disorder v	n	Measurement method m			
		Manual	Reference	CamMeter	All
		$\Delta s_{m,v}$ (%)	$\Delta s_{m,v}$ (%)	$\Delta s_{m,v}$ (%)	Δs_v (%)
No glasses	72	$3.64^a \pm 2.24$	$6.19^a \pm 6.77$	$3.65^a \pm 2.24$	4.21 ± 4.71
Glasses	108	$4.16^a \pm 2.63$	$7.30^a \pm 11.42$	$4.16^a \pm 2.63$	5.00 ± 7.34
		ORB-SLAM2	Tape measure	CamMeter	All
No glasses	108	$12.02^a \pm 20.29$	$19.97^a \pm 24.57$	$6.30^a \pm 6.59$	12.76 ± 19.44
Glasses	72	$14.49^a \pm 21.55$	$25.89^a \pm 47.75$	$5.03^a \pm 2.85$	15.14 ± 31.07

Differing superscript letters indicate significant differences between methods

surements, requiring loop-closing (Mur-Artal et al. 2015) or the utilization of additional sensors like in viSLAM. Studies showed that user movement often triggers re-localization of the camera, causing the point features to be slightly displaced. Several study participants reported this behavior as very disappointing. Study participants liked the usability of the ARKit-based Tape Measure. However, the viSLAM approach showed significantly higher measurement error than the other two approaches. A possible explanation being that the ARKit developers had to find a balance between user experience, i.e., higher frame rates and lower computational effort, and precision of the matching. Despite the lower accuracy, Tape Measure and similar apps are tied to latest Apple or Android devices with precisely calibrated sensors. In contrast, the proposed CamMeter approach is computationally more lightweight and independent from OS constraints.

Our user study is limited in the following aspects. We studied only twelve *different objects*, which are not representative for the variety of all possible objects to be measured. However, we rated the studied objects with regard to properties (planarity, gloss, and self-similarity) that we consider relevant for measurement accuracy and aimed in their selection for large variety of these properties. Our experiment took place under *defined conditions*, e.g., no wind, no direct sun light. It can be considered harder to achieve similar accuracy in a field study. However, all methods were evaluated in the same controlled environment and we hypothesize that the accuracy of all investigated methods would be negatively impacted under field conditions. We evaluated our method with *only two smartphones*, one rather low-end model, and one high-end, suggesting that other devices could possibly produce different results. To minimize *experimenter bias*, we instructed participants through a written tutorial and the experimenter followed a scripted procedure. Although, influence of the participants cannot entirely be prevented, we aimed to minimize it as far as possible.

7 Conclusion and Future Research

Major advances in machine learning approaches for classification tasks trigger a growing demand for high quality image datasets (Wäldchen et al. 2018). Often, images are acquired on mobile devices and additional scene information, such as size annotations, has proven relevant for improving classification results. We designed and implemented a semi-automated measurement approach (CamMeter) that executes on commodity smartphones and can be performed in addition to taking training images. CamMeter's measurement accuracy is dependent on properties of the scene, e.g., the planarity of the measured object as well as image sharpness and noise. However, through extensive empirical evaluation we found the resulting measurement error in an acceptable range. A user study showed that CamMeter's average size annotation error is comparable to that achieved by humans performing the same measurement manually with a ruler, and that it performs equally or better than other measurement

applications. We therefore consider CamMeter an effective semi-automated approach for acquiring object size annotations in-situ along with the image training data.

Future research could make use of smartphones' other sensors, e.g., the gyroscope to warn users about angular displacement, and could analyze other camera parameters if accessible, e.g., the auto-focus position or the time to focus to estimate and validate distances. We are also interested in the possibility of estimating coarse 3D shapes or depth maps from scale changes and back-projection of local distance information into image plane. In addition, further filtering could be applied with an approach like (Saxena et al. 2009) estimating if two key points are rather in the same distance or not.

Acknowledgements We would like to thank all participants of our user experiment for supporting our work. We are funded through a scholarship of the Friedrich Naumann Stiftung; the German Ministry of Education and Research (BMBF) Grants: 01LC1319A and 01LC1319B; the German Federal Ministry for the Environment, Nature Conservation, Building and Nuclear Safety (BMUB) Grant: 3514 685C19; and the Stiftung Naturschutz Thüringen (SNT) Grant: SNT-082-248-03/2014.

References

Aanæs, H., Dahl, A. L., & Perfanov, V. (2010). *A ground truth data set for two view image matching*. Technical report, DTU Informatics, Technical University of Denmark. http://roboimagedata.imm.dtu.dk/papers/technicalReport.pdf.

Aanæs, H., Dahl, A. L., & Steenstrup Pedersen, K. (2011). Interesting interest points. *International Journal of Computer Vision*, *97*(1), 18–35. https://doi.org/10.1007/s11263-011-0473-8.

Agarwal, S. (2009). R.: Building rome in a day. In *International conference on computer vision (ICCV)*.

Apple Inc. (2017). Arkit. https://developer.apple.com/arkit/.

Arandjelovic, R., & Zisserman, A. (2012). Three things everyone should know to improve object retrieval. In *2012 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2911–2918). https://doi.org/10.1109/CVPR.2012.6248018.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (surf). *Computer Vision and Image Understanding*, *110*(3), 346–359.

Bradski, G. (2000). The OpenCV library. *Dr Dobb's Journal of Software Tools*, *25*, 120–123.

Bursuc, A., Tolias, G., & Jégou, H. (2015). Kernel local descriptors with implicit rotation matching. In *Proceedings of the 5th ACM on international conference on multimedia retrieval* (pp. 595–598). ACM, New York, NY, USA, ICMR '15. https://doi.org/10.1145/2671188.2749379.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., et al. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332. https://doi.org/10.1109/TRO.2016.2624754.

Criminisi, A., Reid, I., & Zisserman, A. (1999). A plane measuring device. *Image and Vision Computing*, *17*(8), 625–634.

Criminisi, A., Reid, I., & Zisserman, A. (2000). Single view metrology. *International Journal of Computer Vision*, *40*(2), 123–148. https://doi.org/10.1023/A:1026598000963.

Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(6), 1052–1067.

Dong, J., & Soatto, S. (2015). Domain-size pooling in local descriptors: Dsp-sift. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5097–5106). https://doi.org/10.1109/CVPR.2015.7299145.

Eigen, D., & Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *2015 IEEE international conference on computer vision (ICCV)* (pp. 2650–2658). https://doi.org/10.1109/ICCV.2015.304.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.

Fuentes-Pacheco, J., Ruiz-Ascencio, J., & Rendón-Mancha, J. M. (2015). Visual simultaneous localization and mapping: A survey. *Artificial Intelligence Review*, *43*(1), 55–81.

Google Inc. (2017). Arcore. https://developers.google.com/ar/.

Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the alvey vision conference* (pp. 23.1–23.6). Alvety Vision Club. https://doi.org/10.5244/C.2.23.

Karlsson, N., di Bernardo, E., Ostrowski, J., Goncalves, L., Pirjanian, P., & Munich, M. E. (2005). The vslam algorithm for robust localization and mapping. In *Proceedings of the 2005 IEEE international conference on robotics and automation* (pp. 24–29). https://doi.org/10.1109/ROBOT.2005.1570091.

Ke, Y., & Sukthankar, R. (2004). Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004* (Vol. 2, pp. II–506–II–513). CVPR 2004. https://doi.org/10.1109/CVPR.2004.1315206.

Kim, H., Richardt, C., & Theobalt, C. (2016). Video depth-from-defocus. In *2016 fourth international conference on 3D vision (3DV)* (pp. 370–379). IEEE.

Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality* (pp. 225–234). https://doi.org/10.1109/ISMAR.2007.4538852.

Koenderink, J. J., & van Doorn, A. J. (1991). Affine structure from motion. *Journal of the Optical Society of America A*, *8*(2), 377–385. https://doi.org/10.1364/JOSAA.8.000377.

Kuhl, A., Wöhler, C., Krüger, L., d'Angelo, P., & Groß, H. M. (2006). *Monocular 3D scene reconstruction at absolute scales by combination of geometric and real-aperture methods* (pp. 607–616). Berlin, Heidelberg: Springer. https://doi.org/10.1007/11861898_61.

Lai, K., Bo, L., Ren, X., & Fox, D. (2011). A large-scale hierarchical multi-view rgb-d object dataset. In *2011 IEEE international conference on robotics and automation* (pp. 1817–1824). https://doi.org/10.1109/ICRA.2011.5980382.

Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., & Furgale, P. (2015). Keyframe-based visualinertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, *34*(3), 314–334. https://doi.org/10.1177/0278364914554813.

Levin, A., Fergus, R., Durand, F., & Freeman, W. T. (2007). Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics (TOG)*, *26*(3), 70.

Li, J., & Allinson, N. M. (2008). A comprehensive review of current local features for computer vision. *Neurocomputing*, *71*(1012), 17711787. https://doi.org/10.1016/j.neucom.2007.11.032.

Lin, J., Ji, X., Xu, W., & Dai, Q. (2013). Absolute depth estimation from a single defocused image. *IEEE Transactions on Image Processing*, *22*(11), 4545–4550. https://doi.org/10.1109/TIP.2013.2274389.

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

Luhmann, T., Robson, S., Kyle, S., & Harley, I. (2006). *Close range photogrammetry: Principles, methods and applications*. Dunbeath: Whittles.

McGuinness, K., & O'Connor, N. E. (2010). A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, *43*(2), 434–444. https://doi.org/10.1016/j.patcog.2009.03.008.

Mikolajczyk, K., & Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, *60*(1), 63–86. https://doi.org/10.1023/B:VISI.0000027790.02288.f2.

Moeller, M., Benning, M., Schnlieb, C., & Cremers, D. (2015). Variational depth from focus reconstruction. *IEEE Transactions on Image Processing*, *24*(12), 5369–5378. https://doi.org/10.1109/TIP.2015.2479469.

Moreels, P., & Perona, P. (2006). Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision*, *73*(3), 263–284. https://doi.org/10.1007/s11263-006-9967-1.

Mur-Artal, R., Montiel, J. M. M., & Tards, J. D. (2015). Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, *31*(5), 1147–1163. https://doi.org/10.1109/TRO.2015.2463671.

Mur-Artal, R., & Tards, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, *33*(5), 1255–1262. https://doi.org/10.1109/TRO.2017.2705103.

Mustafah, Y. M., Noor, R., Hasbi, H., & Azma, A. W. (2012). Stereo vision images processing for real-time object distance and size measurements. In *2012 international conference on computer and communication engineering (ICCCE)* (pp. 659–663). https://doi.org/10.1109/ICCCE.2012.6271270.

Nayar, S. K., & Nakagawa, Y. (1994). Shape from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*(8), 824–831. https://doi.org/10.1109/34.308479.

Nitzan, D. (1985). Development of intelligent robots: Achievements and issues. *IEEE Journal on Robotics and Automation*, *1*(1), 3–13.

Peng, B., Zhang, L., & Zhang, D. (2013). A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, *46*(3), 1020–1038. https://doi.org/10.1016/j.patcog.2012.09.015.

Pentland, A. P. (1987). A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, *9*(4), 523–531. https://doi.org/10.1109/TPAMI.1987.4767940.

Piasco, N., Sidib, D., Demonceaux, C., & Gouet-Brunet, V. (2018). A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition*, *74*, 90–109. https://doi.org/10.1016/j.patcog.2017.09.013.

Robertson, P., Frassl, M., Angermann, M., Doniec, M., Julian, B. J., Puyol, M. G., Khider, M., Lichtenstern, M., & Bruno, L. (2013). Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments. In *International conference on indoor positioning and indoor navigation* (pp. 1–10). https://doi.org/10.1109/IPIN.2013.6817910.

Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, *23*(3), 309–314. https://doi.org/10.1145/1015706.1015720.

Rzanny, M., Seeland, M., Wäldchen, J., & Mäder, P. (2017). Acquiring and preprocessing leaf images for automated plant identification: Understanding the tradeoff between effort and information gain. *Plant Methods*, *13*(1), 97. https://doi.org/10.1186/s13007-017-0245-8.

Saxena, A., Sun, M., & Ng, A. Y. (2009). Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(5), 824–840.

Schönberger, J. L., Hardmeier, H., Sattler, T., & Pollefeys, M. (2017). Comparative evaluation of hand-crafted and learned local features. In *Conference on computer vision and pattern recognition (CVPR)*

Seeland, M., Rzanny, M., Alaqraa, N., Wäldchen, J., & Mäder, P. (2017). Plant species classification using flower imagesa comparative study of local feature representations. *PLoS ONE*, *12*(2), e0170,629.

Smith, R. C., & Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, *5*(4), 56–68.

Subbarao, M., & Surya, G. (1994). Depth from defocus: A spatial domain approach. *International Journal of Computer Vision*, *13*(3), 271–294. https://doi.org/10.1007/BF02028349.

Thrun, S., et al. (2002). Robotic mapping: A survey. *Exploring Artificial Intelligence in the New Millennium*, *1*, 1–35.

Torralba, A., Murphy, K. P., & Freeman, W. T. (2004). Sharing features: Efficient boosting procedures for multiclass object detection. In *Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004* (Vol. 2, pp. II–762–II–769). CVPR 2004. https://doi.org/10.1109/CVPR.2004.1315241.

Tuytelaars, T., & Mikolajczyk, K. (2008). Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, *3*(3), 177–280. https://doi.org/10.1561/0600000017.

Uhrig, J., Cordts, M., Franke, U., & Brox, T. (2016). *Pixel-level encoding and depth layering for instance-level semantic labeling* (pp. 14–25). Cham: Springer. https://doi.org/10.1007/978-3-319-45886-1_2.

Wäldchen, J., & Mäder, P. (2018). Plant species identification using computer vision techniques: A systematic literature review. *Archives of Computational Methods in Engineering*, *25*(2), 507–543. https://doi.org/10.1007/s11831-016-9206-z.

Wäldchen, J., Rzanny, M., Seeland, M., & Mäder, P. (2018). Automated plant species identificationtrends and future directions. *PLoS Computational Biology*, *14*(4), e1005,993.

Watanabe, M., & Nayar, S. K. (1998). Rational filters for passive depth from defocus. *International Journal of Computer Vision*, *27*(3), 203–225. https://doi.org/10.1023/A:1007905828438.

Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I., & Tards, J. (2009). A comparison of loop closing techniques in monocular slam. *Robotics and Autonomous Systems*, *57*(12), 1188–1197. https://doi.org/10.1016/j.robot.2009.06.010.

Wittich, H. C., Seeland, M., Wäldchen, J., Rzanny, M., & Mäder, P. (2018). Recommending plant taxa for supporting on-site species identification. *BMC Bioinformatics, 19*. https://doi.org/10.1186/s12859-018-2201-7

ygx2011. (2017). Orb slam2 ios. https://github.com/ygx2011/ORB_SLAM2-IOS.