Monocular 3D Metric Scale Reconstruction using Depth from Defocus and Image Velocity

Tomoyuki Shiozaki and Gamini Dissanayake

Abstract— **This paper presents a novel approach to metric scale reconstruction of a three-dimensional (3D) scene using a monocular camera. Using a sequence of images from a monocular camera with a fixed focus lens, metric distance to a set of features in the environment is estimated from image blur due to defocus. The blur texture ambiguity which causes scale errors in depth from defocus is corrected in an EKF framework that exploits image velocity measurements. We show in real experiments that our method converges to a metric scale, accurate, sparse depth map and 3D camera poses with images from a monocular camera. Therefore, the proposed approach has the potential to enhance robot navigation algorithms that rely on monocular cameras.**

I. INTRODUCTION

A mobile robot must be able to map its environment and estimate its egomotion to be able to perform many tasks. Information from a monocular camera, visual odometry (VO) [1], visual simultaneous localization and mapping (V-SLAM) [2], or structure from motion (SfM) [3], can be used to generate this information accurate up to a scale. Typically, stereo cameras [4] or RGB-D cameras [5] are necessary to generate three-dimensional (3D) metric scale reconstruction. Although both stereo setups and RGB-D cameras are now widely available and becoming compact, the fact remains that the ability to use a monocular camera is still attractive, particularly in robotic applications, due to the small size and the versatility.

The typical monocular approaches to estimate scale are depth from focus (DfF) and depth from defocus (DfD) [6]. DfF requires many images of the same scene with different focus setting, thus is not suitable for mobile robots. On the other hand, DfD relies on the amount of defocus blur which depends on the distance to the object [7]. It has been demonstrated that the defocus blur can be estimated even from a single image [8], [9]. Therefore, the use of DfD has the potential to reduce the complexity of monocular VO and V-SLAM algorithms, and enhance their output by producing metric scale data. However, DfD from a single image has a fundamental problem: the blur texture ambiguity [8]. This means that from a single image it is not straightforward to distinguish between blur caused by defocus and that caused by texture. Although the coded aperture method [10] or active lighting [11] can resolve the blur texture ambiguity, these methods require modifications to the camera or additional illumination and therefore compromise the main advantage of monocular cameras; their versatility.

In this paper, we propose a method to correct the scale error of DfD due to the blur texture ambiguity and estimate the metric distance to a set of features of a static scene in an extended Kalman filter (EKF) framework. Our method uses a sequence of images taken by a moving monocular camera using a lens with a fixed focal length and a finite aperture. We show that the blur texture ambiguity is mainly caused by regions with low contrast. Also, we demonstrate that the scale error caused by the low contrast can be estimated from changes in defocus blur and image velocity induced by camera motion. The main contributions of this paper are as follows:

- Derivation of an equation to correct the scale error caused by blur texture ambiguity
- EKF framework for metric 3D reconstruction of an environment
- Experimental demonstration of the proposed method in combination with SfM using a conventional camera

We note here that the proposed approach does not require any additional sensors or camera modifications. Therefore it retains all the advantages of using a monocular camera and has the potential to enhance the level of information typically gathered through monocular VO or V-SLAM, particularly for robot navigation.

This paper is organized as follows. Section II provides a review of related works on DfD. Section III demonstrates how to estimate the metric scale. In this section, the DfD method is introduced, and the scale error caused by low contrast texture is formulated. Then, the EKF approach based on the relationship between changes in defocus blur and image velocity induced by camera motion is proposed. In section IV, experimental results are presented. The first experiment is to illustrate properties of the proposed algorithm. The second experiment is to demonstrate the application of the proposed method in combination with SfM. Section V discusses the strengths and limitations of the proposed method. Section VI concludes the paper.

II. RELATED WORK

Conventional depth estimation methods from a monocular camera require multiple images with changes to camera settings such as aperture and focal length to obtain different defocus blur [12], [13], [14]. Taking images with different camera settings is complex and requires solving the matching problem [6] and therefore not particularly attractive in many applications. Pentland [12] pointed out that defocus can be

T. Shiozaki and G. Dissanayake are with the Centre for Autonomous Systems, Faculty of Engineering and IT, University of Technology Sydney (UTS), NSW 7500 2007, Australia `shiozaki.tomoyuki@student.uts.edu.au` `gamini.dissanayake@uts.edu.au`

extracted at edge locations on a single image. Elder [15] used the derivatives of the input images to find the edge locations and their defocus blur. Zhuo and Sim [8] proposed a method based on the Gaussian gradient ratio that is more robust to image noise than those available in the literature.

However, to be effective, single image DfD methods require strategies to resolve ambiguities due to focal plane, motion blur, and blur texture. The focal plane ambiguity results from the fact that DfD from a single image cannot differentiate on which side of the focal plane the objects are placed [8]. Kumar et al. [16] demonstrated that chromatic aberration provides an effective indicator to solve the focal plane ambiguity. Second, the motion blur influences the defocus estimation. However, motion blur is also a useful depth cue. Paramanand and Rajagopalan [17] proposed a method to recover the 3D structure from both motion blur and defocus blur with camera motion in an Unscented Kalman Filter (UKF) framework. Third, the blur texture ambiguity is still a challenging problem. Srikakulapu et al. [18] proposed a method to correct the depth map by using texture information such as edge sharpness, spot energy, and contrast. However, this approach cannot estimate the metric scale. As addressing the blur texture ambiguity is the main objective of this paper, we do not address the focal plane ambiguity and motion blur. We assume that all observed objects exist on one side of the focal plane and the camera motion is sufficiently slow.

The work most related to our paper is by Wöhler et al. [7]. They combined DfD with SfM and estimated the metric distance with reasonable accuracy. In [7], the scale error caused by the blur texture ambiguity is termed due to "image content". The main drawback of this method is the assumption that each of observed features in the scene is in focus at somewhere in a sequence of images. The method proposed in this paper relaxes this condition and is able to estimate the metric scale even when the features concerned are never in focus.

III. THREE DIMENSIONAL METRIC RECONSTRUCTION

In this section, the proposed methodology for 3D metric reconstruction is described. First, the DfD approach based on a formula derived using the thin lens model is introduced. The point spread function is approximated with a Gaussian to model the amount of defocus blur in a given image at edge locations. Second, ambiguity caused by low contrast texture is formulated. Finally, using the relationship between changes in defocus blur and image velocity induced by camera motion, an EKF based approach to resolving this ambiguity is proposed.

A. Depth from Defocus

Image formation based on the thin lens model is shown in Fig. 1 [12]. All rays from a point located at the in-focus distance d_f converge to a single point on the image plane placed at the distance b_f from the lens. On the other hand, rays from an object located at any other distance d converge to a point on a plane located at a distance $b_f + b_\delta$ from the lens and therefore will be out of focus when viewed at the image plane. Rays from such an object will make a blurred circle on the image plane. This is known as the circle of confusion (CoC). The diameter of this circle is given by

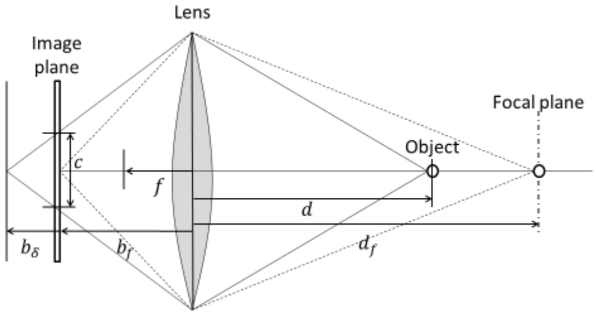

Fig. 1. Thin lens model. Origin is the lens center. b_f is the distance to the image plane. d_f is the distance to the focal plane. Size of c depends on the object distance d. When the image plane is placed at $b_f + b_\delta$, the object is best focused.

$$c = \frac{|d - d_f|}{d} \frac{f^2}{N(d_f - f)}, \tag{1}$$

where f is focal length and N is f-number of the camera [8]. It is seen that larger $|d - d_f|$ is, the larger the CoC. To get a large amount of defocus blur, a long focal length and a large aperture are required as the f-number is $N = f/A$ where A is the aperture diameter of the lens.

The size of c can be approximated by σ of the Gaussian-shaped point spread function (PSF) $G(\sigma)$ as

$$I_i = G(\sigma) * I_{f_i}, \tag{2}$$

where $*$ means convolution, I_i is a small region of interest (ROI) around a feature i, and I_{f_i} is the ROI around the same feature when it is best focused [7]. c can be expressed as $c = \gamma\sigma$ with a camera-specific value of γ [19]. We use the method proposed by Zhuo and Sim [8] to estimate σ from an image.

Wöhler et al. [7] proposed the following function to relate d and σ:

$$\begin{aligned} \sigma = D(d) &= \frac{1}{\phi_1} \exp(-\frac{1}{\phi_2}(b_\delta(d))^2) + \phi_3, \\ b_\delta(d) &= \frac{df}{d - f} - b_f, \end{aligned} \tag{3}$$

where ϕ_1, ϕ_2, and ϕ_3 are the calibration parameters. For a given camera setup, these parameters together with b_f and f can be estimated using a calibration process which is performed by measuring values of σ at the corners of the black-and-white checkerboard while changing its distance. Solving Eq. (3) yields the metric distance d from the measured σ. However, Wöhler et al. [7] pointed out that the measuring σ does not work well on features other than black-and-white corners due to errors caused by blur texture ambiguity of the input image. This ambiguity is due to many factors such as soft shadows, brightness and color of the object, and the

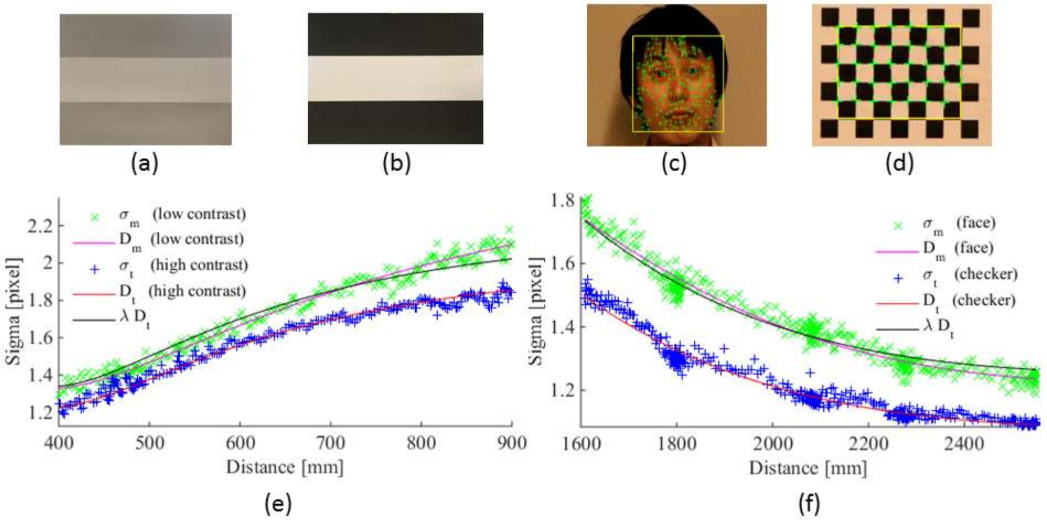

Fig. 2. Demonstration of Eq. (4). (a) is a low contrast edge pattern with 50% and 75% gray levels. (b) is a high contrast binary edge pattern. (c) is a face and (d) is a checkerboard. × and + show the measured σ. Red and magenta lines show the approximations of measured σ using Eq. (3). Black line shows the effect of correction using Eq. (4). (e) was taken with $N = 3.5$ and $d_f = 400$mm. (f) was taken with $N = 5.0$ and $d_f = 2800$mm. Table I shows all the other parameters.

TABLE I

CALIBRATION PARAMETERS

case	ϕ_1	ϕ_2	ϕ_3	b_f[mm]	f[mm]
case1	-1.13	0.275	2.13	20.4	19.4
case2	-0.555	1.42	2.88	47.7	46.9

Parameters of D_t. Case1 and case2 are for Fig. 2 (e) and (f), respectively.

illumination. Our experiments demonstrated that one of the main causes is the difference of the contrast between the ROIs. Also, it was observed that this error could be expressed empirically by the following equation:

$$\sigma_m = \lambda \sigma_t = \lambda D(d), \tag{4}$$

where σ_m is the measured σ at a low contrast edge and σ_t is the true σ measured at a high contrast edge without texture ambiguity, and λ describes the correction factor for the extent of texture blur. This is illustrated in Fig. 2. Fig. 2(a) and (b) are low contrast and high contrast edge patterns, respectively. In Fig. 2(e), σ_m is measured from the low contrast edge, σ_t is measured from the high contrast edge, approximations D_m and D_t are based on Eq. (3), and λD_t is based on Eq. (4).

As can be seen in Fig. 2(e), λD_t is close to D_m. This means that a constant λ can approximate the extent of texture blur of σ_m. This is because the gradients at edge locations are used to estimate the amount of defocus blur in [8], [9], and [15]. When the contrast of the edge location is low, the gradient becomes low. This behavior caused by low contrast texture is independent of the distance between the camera and the object. Therefore, λ remains constant independent of d. It can also be seen that the same is true in a more complex scene. Fig. 2(f) shows the results from the images of a face (Fig. 2(c)) and the checkerboard (Fig. 2(d)). The σ_m and σ_t are measured at the features detected by KLT tracker [20], [21]. λ approximates the extent of texture blur of σ_m. The median of measured values of σ at each distance was used in the experiments. The illumination and camera parameters such as the shutter speed, the aperture size, and the sensitivity were remain unchanged during the experiments.

B. Extended Kalman Filter

The experiments presented above demonstrate that the relationship between measured σ_m of a point and the distance d can be expressed using Eqs. (3) and (4). This section presents an EKF framework for estimating the scale based on these relationships.

We begin by defining the scale factor Λ and image velocity v_i, where v_i is the projection of the 3D relative velocity to a point onto the image plane with unit focal length $f = 1$. Λ can be used to obtain the geometry of the scene in metric scale using

$$d_i = \Lambda u_i, \tag{5}$$

where d_i is the metric distance to each point of a scene, u_i is its up to a scale counterpart. Both u_i and v_i can be obtained using a sequence of images and one of the many algorithms available in the literature, for example, [22], assuming that the observed object is stationary. Subscript i is used to denote the i-th point. Given that image velocity $v_i = \frac{\dot{u}_i}{u_i}$, the time derivative of Eq. (5) can be expressed as

$$\dot{d}_i = \Lambda \dot{u}_i = \Lambda u_i v_i. \tag{6}$$

Taking the time derivative of Eq. (4) and using Eqs. (5) and (6):

$$\begin{aligned} \dot{\sigma}_{m,i} &= \lambda_i \frac{d}{dt} D(\Lambda, u_i, v_i) \\ &= \frac{2\lambda_i b_{\delta,i} \Lambda u_i v_i}{\phi_1 \phi_2} \left(\frac{f}{\Lambda u_i - f}\right)^2 \exp\left(-\frac{1}{\phi_2} b_{\delta,i}^2\right), \end{aligned} \tag{7}$$

where $b_{\delta,i} = \frac{\Lambda u_i f}{\Lambda u_i - f} - b_f$. In the following, we describe the use of an EKF to estimate Λ and λ_i, which are constants.

The state vector of the EKF is as follows:

$$X = [\Lambda \ \lambda_i \ \sigma_{m,i}]^T, \tag{8}$$

where $i = 1 \ldots N$ and N is the number of observed points. The process equations governing the evolution of the state vector are

$$\begin{aligned} \Lambda_{k+1} &= \Lambda_k + \varepsilon_{\Lambda,k}, \\ \lambda_{i,k+1} &= \lambda_{i,k} + \varepsilon_{\lambda,i,k}, \\ \sigma_{m,i,k+1} &= \sigma_{m,i,k} + \lambda_{i,k} \frac{d}{dt} D(\Lambda_k, u_{i,k}, v_{i,k}) \ \Delta t + \varepsilon_{\sigma,i,k} \Delta t, \end{aligned} \tag{9}$$

where Δt is defined as $\Delta t = t_{k+1} - t_k$ and $\varepsilon(k) = [\varepsilon_{\Lambda,k} \ \varepsilon_{\lambda,i,k} \ \varepsilon_{\sigma,i,k}]^T$ represents the process noise. Note that Eq. (7) is used to compute $\sigma_{m,i,k+1}$.

The observations of the defocus blur $\sigma_{m,i}$ are obtained at edge locations using the method proposed by Zhuo and Sim [8]. Therefore, the observation vector is $Z = [\sigma_{m,i}]^T$. The observation equations then become

$$\hat{\sigma}_{m,i,k} = \sigma_{m,i,k} + \eta_{i,k}, \tag{10}$$

where $\eta(k) = [\eta_{i,k}]^T$ is the observation noise vector. Furthermore, the constraint defined by Eq. (4) always needs to be satisfied. In the EKF framework, equality constraints can be imposed using the projection method [23]. These constraints are rewritten as

$$c[X(k)] = \frac{\lambda_{i,k} D(\Lambda_k, u_{i,k})}{\sigma_{m,i,k}} - 1 + \zeta_{i,k} = 0, \tag{11}$$

where $\zeta(k) = [\zeta_{i,k}]^T$ is the noise vector added to account for the possible extent of constraint violations.

We assume that the noises $\varepsilon(k)$, $\eta(k)$ and $\zeta(k)$ are all Gaussian, temporally uncorrelated and zero-mean

$$E[\varepsilon(k)] = E[\eta(k)] = E[\zeta(k)] = 0, \forall k \tag{12}$$

with corresponding covariance

$$\begin{aligned} E[\varepsilon(k)\varepsilon(k)^T] &= Q, \\ E[\eta(k)\eta(k)^T] &= R, \\ E[\zeta(k)\zeta(k)^T] &= Rc. \end{aligned} \tag{13}$$

Equations used to implement the EKF are given in the Appendix.

IV. EXPERIMENTAL EVALUATIONS

A. Experiment 1: Properties of the proposed EKF

The objective of this experiment is to evaluate the ability of the EKF shown in subsection III-B to estimate Λ and λ_i. In this experiment, the same edge patterns and camera settings used to obtain Fig. 2(e) were used. The chart is shown in Fig. 3(a). A sequence of images with 640 × 480 pixels resolution at 30fps was taken by the CANON EOS 650D with the EF-S 18-135mm f/3.5-5.6 IS STM lens. Initially, the camera was positioned to face the chart at a distance $d = 1000$ mm. The camera was moved at an approximately constant speed of 55 mm/sec along the optical axis until $d = 400$ mm. During the experiment, values of σ were measured at the edge locations on (k) of Fig. 3(a) and the median of them was used as $\sigma_{m,i}$. σ_t was measured at the edge locations on (j) of Fig. 3(a) in the same way. The true scale was calculated using a known size of the checkerboard shown in (i) of Fig. 3(a). u_i and v_i were calculated from changes in the size of the checkerboard in the image sequence. Therefore, in this experiment, measured v_i was accurate except for some small amount of noise.

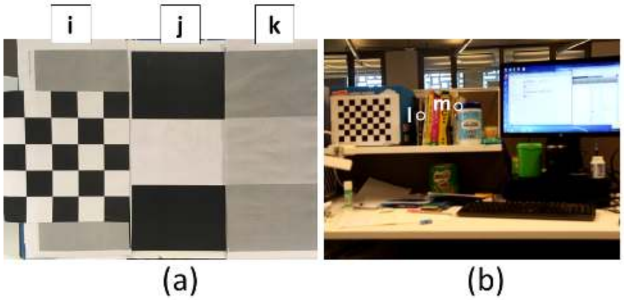

Fig. 3. Experimental environments. (a) shows the chart used in experiment 1, where (i) is the checkerboard used to get the true metric scale, and (j) and (k) have the same edge patterns as (b) and (a) described in Fig. 2, respectively. (b) shows the scene used for experiment 2. (l) and (m) are two of the feature points where $\sigma_{m,i}$ are measured.

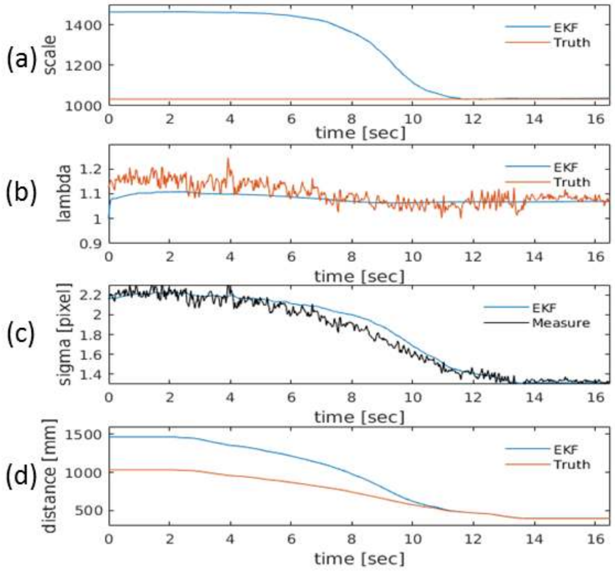

Fig. 4. The estimates of Λ (a), λ_i (b), $\sigma_{m,i}$ (c), and the metric distance d_i (d) in experiment 1. The blue lines show the results from the EKF, the red lines show the ground truth, and the black line shows the measurement.

Fig. 4(a), (b), and (c) show the estimates of Λ, λ_i, and $\sigma_{m,i}$. The true value for λ_i was calculated from the true scale with Eq. (4). It can be seen that $\sigma_{m,i}$ gradually changes as expected and Λ converges as more and more measurements are obtained. Fig. 4(d) shows the estimated metric distance. After convergence, the final distance error between the camera and the chart is only 1.6 mm. These results illustrate that the proposed method can correctly estimate the metric scale.

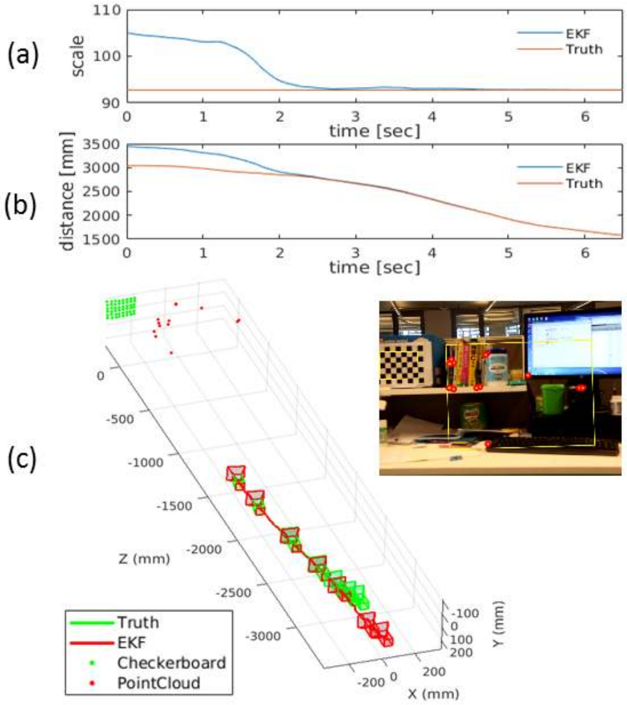

Fig. 5. (a) and (b) show the estimates of Λ and d_i to the point indicated in (m) of Fig. 3(b). The blue lines show the results from the EKF and the red lines show the ground truth. (c) is the camera poses and 3D sparse depth map reconstructed to the metric scale. The red line shows the trajectory of the camera with EKF. The green line shows the true trajectory of the camera.

TABLE II

PARAMETERS FOR EXPERIMENT 2

ϕ_1	ϕ_2	ϕ_3	b_f[mm]	f[mm]	N	d_f [mm]
-1.14	0.086	1.53	32.3	32.2	4.0	5000

B. Experiment 2: 3D metric scale reconstruction in a cluttered environment

This experiment is aimed at demonstrating that the proposed algorithm can estimate Λ and λ_i even in a cluttered environment. A set of feature points around a desk in Fig. 3(b) was observed by the same camera with Experiment 1. The parameters used are shown in Table II. Initially, the camera was set facing to the desk at the distance of approximately 3000 mm. It was then moved at an approximately constant speed (around 230 mm/sec) until about 1500 mm. u_i and v_i were measured using the SfM algorithm with bundle adjustment [24] as implemented in Matlab®. The camera egomotion and the 3D sparse depth map obtained from the SfM algorithm were then rescaled with the metric scale estimated using the proposed EKF. In this experiment, the values of σ measured at feature points detected by KLT tracker were used as $\sigma_{m,i}$. As in the case with subsection IV-A, the true scale was calculated with the checkerboard with a known size shown in Fig. 3(b).

Fig. 5(a) shows the estimate of Λ. Fig. 5(b) shows the metric distance d_i to the point indicated in (m) of Fig. 3(b). Fig. 5(c) shows the camera poses and 3D sparse depth map with the estimated scale. Note that the large error in camera position at the beginning is expected as the EKF takes time to converge. The root-mean-square error of the final distances from the camera to the reconstructed 3D points is only 0.32mm under the assumption that the 3D sparse depth map obtained from the SfM algorithm is true. Results from this experiment demonstrate that the proposed method combined with SfM can generate 3D camera poses and sparse depth map to metric scale with only a monocular camera even in a cluttered environment.

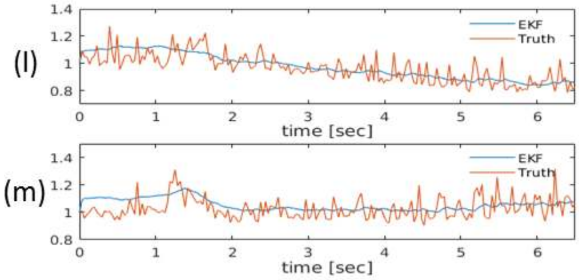

Fig. 6. (l) and (m) show the estimates of λ_i at the points indicated in (l) and (m) of Fig. 3(b). The blue lines show the results from the EKF and the red lines show the ground truth. λ_i of (l) decreases continuously due to the changes in texture, although λ_i of (m) is almost constant. The fluctuations seen from 1 to 2 seconds are due to motion blur.

V. DISCUSSION

Assumption that λ_i is constant

The proposed method relies on the assumption that λ_i is constant, which does not hold if there are significant changes in texture and illumination through the image sequence. For example, Fig. 6(l) shows the estimate of λ_i at the point indicated in (l) of Fig. 3(b). It is clear that λ_i of (l) decreases continuously due to the changes in texture. The point of (l) is positioned at the spine of the book. The spine appears as a single edge when the camera is at a distance. However, it reveals rich texture due to the letters present on it when the camera is nearby. The amount of defocus blur cannot be estimated correctly in this case as gradient the calculated is not correct when there are many discontinuities in the ROI I_i shown in Eq. (2). Our assumption that λ_i is constant is no longer correct these situations. However, the use of additive noise for the possible extent of constraint violations in EKF relaxes the constraint that λ_i is constant. Therefore, as seen from multiple results in section IV, the EKF can estimate the metric scale correctly despite the fact that some of λ_i change with the camera motion.

Size of the lens

The range over which the proposed method applies depends on the focal length f, and the aperture size A. In small cameras such as those present in mobile phones, defocus blur is not present at points beyond relatively short distances from the lens. Fig. 7 shows the 3D map and camera poses reconstructed by the rear camera on iPhone SE. Although this result demonstrates that the proposed method is effective

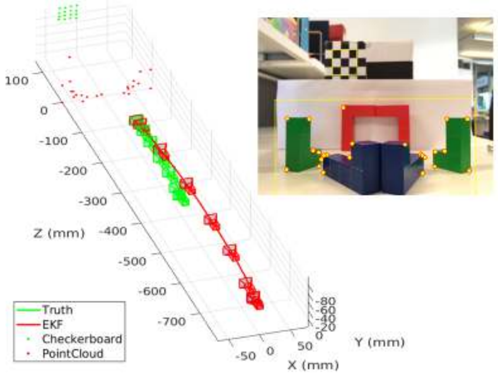

Fig. 7. The camera poses and 3D sparse depth map reconstructed by iPhone SE. The red line shows the trajectory of the camera with EKF. The green line shows the true trajectory of the camera.

even for a small camera on a mobile phone, the effective measuring range is only about 500 mm. Therefore, in typical robotic applications, it will be necessary to select a suitable lens to increase the effective range.

VI. CONCLUSION

An approach for metric scale reconstruction of 3D environments from a sequence of monocular images is demonstrated in this paper. It is shown that blur due to texture can be represented using a constant gain when estimating depth from defocus. An EKF framework that incorporates information from non-scaled distances and image velocity is shown to be able to resolve blur texture ambiguity and produce accurate metric reconstruction. Use of the proposed approach in more complex and large scale environments, and examining the possible positive impact of being able to estimate scale in conventional monocular SLAM algorithms will be the focus of future work.

APPENDIX

Julier and LaViola [23] proposed a two-step projection method to implement an EKF with nonlinear equality constraints. The procedure is as follows:

1) compute constrained covariance
$Sc(k) = Hc(k)P^{*}(k|k)Hc^{T}(k) + Rc(k)$
2) compute constrained gain
$Wc(k) = P^{*}(k|k)Hc^{T}(k)Sc^{-1}(k)$
3) apply first-step constraint for estimate
$X^{+}(k|k) = X^{*}(k|k) - Wc(k)(HcX^{*}(k|k) - dc(k))$
4) apply first-step constraint for state covariance
$P^{+}(k|k) = P^{*}(k|k) - Wc(k)Sc(k)Wc^{T}(k)$
5) update 1) and 2) with $P^{+}(k|k)$ instead of $P^{*}(k|k)$
6) apply second-step constraint for estimate
$X(k|k) = X^{+}(k|k) - Wc(k)(HcX^{+}(k|k) - dc(k))$
7) apply second-step constraint for state covariance
$P(k|k) = P^{+}(k|k) + (X(k|k) - X^{+}(k|k))$
$\times (X(k|k) - X^{+}(k|k))^{T}$

Here, $X^{*}(k|k)$ is the unconstrained estimate, $P^{*}(k|k)$ is the unconstrained state covariance. $Hc(k)$ and $dc(k)$ are related by $Hc(k)X(k) = dc(k)$, and are derived from Eq. (11).

REFERENCES

[1] D. Scaramuzza and F. Fraundorfer, “Visual Odometry, Part I: The First 30 Years and Fundamentals [Tutorial]”, IEEE Robotics & Automation Magazine, vol. 18, pp. 80-92, 2011.
[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: A Versatile and Accurate Monocular SLAM System,” IEEE Transactions on Robotics, vol. 31, pp. 1147-1163, 2015.
[3] D. Nister, “An efficient solution to the five-point relative pose problem,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, pp. 756-770, 2004.
[4] A. Howard, “Real-time stereo visual odometry for autonomous ground vehicles,” in 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3946-3952, 2008.
[5] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, “3-D Mapping with an RGB-D camera,” IEEE Transactions on Robotics, vol. 30, pp. 177-187, 2014.
[6] Y. Y. Schechner and N. Kiryati, “Depth from Defocus vs. Stereo: How Different Really Are They?,” International Journal of Computer Vision, vol. 39, pp. 141-162, 2000.
[7] C. Wöhler, P. d’Angelo, L. Krüger, A. Kuhl, and H.-M. Groß, “Monocular 3D scene reconstruction at absolute scale,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 64, pp. 529-540, 2009.
[8] S. Zhuo and T. Sim, “Defocus map estimation from a single image,” Pattern Recognition, vol. 44, pp. 1852-1858, 2011.
[9] S. Liu, F. Zhou, and Q. Liao, “Defocus Map Estimation From a Single Image Based on Two-Parameter Defocus Model,” IEEE Transactions on Image Processing, vol. 25, pp. 5943-5956, 2016.
[10] C. Zhou, L. Stephen, and S. Nayar, “Coded aperture pairs for depth from defocus,” in 2009 IEEE 12th International Conference on Computer Vision, pp. 325-332, 2009.
[11] M. J. Amin and N. A. Riza, “Active depth from defocus system using coherent illumination and a no moving parts camera,” Optics Communications, vol. 359, pp. 135-145, 2016.
[12] A. P. Pentland, “A New Sense for Depth of Field,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, pp. 523-531, 1987.
[13] P. Favaro and S. Soatto, “A geometric approach to shape from defocus,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, pp. 406-417, 2005.
[14] A. N. Rajagopalan and S. Chaudhuri, “Optimal selection of camera parameters for recovery of depth from defocused images,” in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 219-224, 1997.
[15] J. H. Elder and S. W. Zucker, “Local scale control for edge detection and blur estimation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, pp. 699-716, 1998.
[16] H. Kumar, S. Gupta, and K. S. Venkatesh, “Resolving focal plane ambiguity in depth map creation from defocus blur using chromatic aberration,” in 2015 10th International Conference on Information, Communications and Signal Processing, pp. 1-5, 2015.
[17] C. Paramanand and A. N. Rajagopalan, “Depth From Motion and Optical Blur With an Unscented Kalman Filter,” IEEE Transactions on Image Processing, vol. 21, pp. 2798-2811, 2012.
[18] V. Srikakulapu, H. Kumar, S. Gupta, and K. S. Venkatesh, “Depth estimation from single image using Defocus and Texture cues,” in 2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, pp. 1-4, 2015.
[19] C. Wöhler, 3D computer vision: efficient methods and applications: Springer Science & Business Media, 2012.
[20] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in 7th international joint conference on Artificial intelligence, pp. 674-679, 1981.
[21] J. Shi and C. Tomasi, “Good features to track,” in 1994 IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600, 1994.
[22] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, An invitation to 3-d vision: from images to geometric models. New York: Springer, 2003.
[23] S. J. Julier and J. J. LaViola, “On Kalman Filtering With Nonlinear Equality Constraints,” IEEE Transactions on Signal Processing, vol. 55, pp. 2774-2784, 2007.
[24] M. I. Lourakis and A. A. Argyros, “SBA: A software package for generic sparse bundle adjustment,” ACM Transactions on Mathematical Software (TOMS), vol. 36, p. 1-30, 2009.