

Scale Estimation for Monocular SLAM using Depth from Defocus

by

Tomoyuki Shiozaki

A thesis submitted in partial fulfilment of the
requirements for the degree of Master of Engineering (Research)

at the

Centre for Autonomous Systems
Faculty of Engineering and Information Technology
University of Technology Sydney

March 2018

Certificate of Original Authorship

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signed:

Date:

Scale Estimation for Monocular SLAM using Depth from Defocus

by

Tomoyuki Shiozaki

A thesis submitted in partial fulfilment of the requirements for the
degree of Master of Engineering (Research)

Abstract

An autonomous robot must map its environment and estimate its egomotion to perform effectively. Monocular simultaneous localization and mapping (SLAM) can generate maps of the robot's environment, except for the absolute scale. Alternatives based on stereo or RGB-D camera based SLAM systems can obtain the metric scale but have disadvantages in terms of the cost, size and power requirements. This thesis is focused on the development of an absolute metric scale monocular SLAM system for autonomous robots. A depth from defocus (DfD) technique that relies on image blur is used to estimate the metric scale. However, existing methods for DfD suffer from ambiguities caused by texture, motion blur, and the location of the focal plane. The novelty of this research is combining DfD with camera motion to resolve estimation errors caused by these ambiguities and compute a reliable measure of metric scale. Monocular SLAM algorithms are also prone to scale drift, where the scale gradually changes while mapping. It is demonstrated that integrating DfD into monocular SLAM eliminates scale drift and results in accurate metric scale maps.

Acknowledgements

This work would not have been accomplished without the support and encouragement of many others around me. In the following lines, I would like to take this opportunity to show my appreciation to those people who have helped me in the realization of this thesis.

Especially, my deepest gratitude and appreciation goes to my primary supervisor, Professor Gamini Dissanayake for accepting me as a Master student and for providing the opportunity to work on this topic. He has been supervising me with his immense patience, motivation, enthusiasm, and expertise. Thanks for the countless hours of thought-provoking discussions.

I am deeply grateful to my alternate supervisor Doctor Ravindra Ranasinghe for his continuous support which was indispensable for the accomplishment of this thesis.

I would like to express my greatest appreciation to Professor Tomonari Furukawa for giving me the initial idea of performing this study and for his helpful advice on this research.

I would like to offer my special thank to the funding received through Canon Inc. and Canon Australia Pty. Ltd. to undertake my Master's degree. I would never have been able to reach where I am today, without their support.

My gratitude also goes to Mr. John Hazelton for proofreading my thesis and providing me with insightful comments and suggestions.

I would like to thank Mr. Kuranage Asok Aravinda Perera and Mr. Clyde Webster for reviewing my thesis and providing valuable advice. I have had the support and encouragement of them.

I am appreciative to all my colleagues at Centre for Autonomous Systems (CAS), I find it to have been an exciting opportunity for me to work with such an intelligent and motivated group of people.

Finally, and most importantly, I would like to thank my wife Hiromi and my two little children, Shogo and Mizuho. Their support, encouragement, patience, and love were undeniably the bedrock upon which the past two years of my life in Sydney have been built.

Contents

Declaration of Authorship	i
Abstract	ii
Acknowledgements	iii
List of Figures	vi
List of Tables	xi
Nomenclature	xiv
1 Introduction	1
1.1 Background	1
1.2 Aim, Objective, and Significance	3
1.3 Research Method	3
1.3.1 Objective 1: Monocular Scale Estimation	3
1.3.2 Objective 2: Scale drift-free monocular SLAM system	4
1.4 Contribution	5
1.5 Publications Related to this Thesis	5
1.6 Thesis Outline	6
2 Literature Review	7
2.1 Monocular Depth Estimation	7
2.2 Depth from Defocus	10
2.2.1 Blur Texture Ambiguity	13
2.2.2 Impact of Motion Blur	14
2.2.3 Focal Plane Ambiguity	15
2.3 Monocular SLAM	16
2.3.1 Scale drift in monocular SLAM	17
2.3.2 Keyframe-based Optimization	18
2.3.3 ORB SLAM	19
2.4 Conclusion	20

3	Monocular Metric Scale Estimation	22
3.1	Depth from Defocus	22
3.2	Blur Texture Ambiguity	28
3.3	Extended Kalman Filter	30
3.4	Experimental Evaluation	34
3.4.1	Experiment 1: Properties of the proposed method	34
3.4.2	Experiment 2: 3D metric scale estimation in a cluttered environment	38
3.5	Discussion	41
3.6	Conclusion	44
4	Scale Drift-free Monocular SLAM	46
4.1	Scale Drift Elimination Strategy	46
4.2	Eliminating the Impact of Motion Blur	48
4.3	Scale Optimization	51
4.3.1	Optimization	51
4.3.2	Initial Guess	52
4.3.3	Feature Point Selection for Optimization	53
4.4	Experimental Evaluation	56
4.4.1	Experiment 1: Eliminating scale drift in a corridor environment	57
4.4.1.1	Dataset	57
4.4.1.2	Scale estimation result	58
4.4.2	Experiment 2: Demonstration using a small camera	62
4.4.2.1	Dataset	62
4.4.2.2	Scale estimation result	65
4.5	Discussion	66
4.6	Conclusion	70
5	Conclusion	71
5.1	Summary of Contributions	72
5.2	Discussion of Limitations and Future Work	74

List of Figures

2.1	Illustration of Structure from Motion. The red point shows a corresponding point of a scene observed from different camera positions.	8
2.2	Illustration of Motion Blur. (a) is a focused image captured by a stationary camera and (b) is a blurred image captured by a moving camera. The size of the motion blur is one of the monocular depth cues.	8
2.3	Illustration of Active Stereo. In this example, a stripe pattern is projected onto the surface of the object. The effective measuring range depends on the light source power.	8
2.4	Illustration of geometric constraints. Under the assumption that the ground is flat, the known fixed camera height above the ground plane (H_c) allows calculation of the metric scale in a SfM or SLAM system. The size of an object such as a vehicle running in front of the camera (H_v) is also used to recover the metric scale.	9
2.5	Illustration of photometric stereo. Illumination from different directions makes different shading onto the surface of an object. The changes in the intensities on the images makes it possible to compute the 3D shape of the object.	10
2.6	Illustration of Depth from Defocus. (a) shows the image formation of the thin lens model and (b) shows Depth-Defocus curve. The defocus blur amount depends on the distance between the object and the focal plane as shown in (b).	11
2.7	Illustration of image convolution. $*$ means the convolution operator.	12
2.8	Demonstration of the defocus map estimation method proposed by [6]. (a) is the input image, (b) is the sparse defocus map at edge locations of (a), and (c) is the full defocus map generated by propagating the defocus blur amount at edge locations of (b) to the entire image. In (b) and (c), the grayscale indicates the amount of defocus blur.	12
2.9	Illustration of the blur texture ambiguity, adapted from [6]. (a) is the input image and (b) is the full defocus map. In the white boxed region, a wrong defocus estimation occurred due to the texture of the flower.	13
2.10	Illustration of the difference between defocus blur (a) and motion blur (b). Although the defocus blur is a non-directional blur, the motion blur is in the same direction as the camera or the object motion.	14

2.11	Illustration of focal plane ambiguity. The objects across the focal plane shown as red and yellow dots in (a) have the same amount of defocus blur as shown in (b) where the dot colors correspond to the colors of objects in (a).	15
2.12	Illustration of scale drift. (a) shows the feature location and camera trajectory estimates before loop-closure. The blue solid-line is the trajectory estimate, the brown dot-line is the ground truth, and the blue dots are feature location estimates. Scale drift causes the mapping error. (b) shows the feature location and camera trajectory estimates after loop-closure. The orange solid-line is the trajectory estimate, the orange dots are feature location estimates. Although the loop-closure reduces the effect of scale drift, the scale error in different local regions still remains in the map, which means the scale factors Λ_1 , Λ_2 , and Λ_3 , which are ideally the same value, become different values.	17
2.13	Illustration of the difference between filter-based and optimization-based SLAM systems, adapted from [52]. (a) and (b) show the filter-based and the optimization-based systems, respectively. The orange lines show the data connection between camera poses and feature locations used for the estimation. The blue lines show the tight data connection between feature location estimates. The camera poses shown with dashed-line are not used for the estimation. Note that the features still have correlations with each other as the result of the marginalization of the intermittent keyframes, although not shown in (b).	18
3.1	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. The 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.	23
3.2	The illustration of image convolution. (a) shows the 1D case and (b) shows the 2D case. In (a), the blue line is a sharp edge, the orange line is the Gaussian PSF, and the green line is the blurred edge due to the image convolution.	24
3.3	The overview of the blur estimation method proposed by [6]. The green lines show the blurred edges due to the defocus. The red lines show the reblurred edges by a known Gaussian PSF. The black dash lines show the edge locations. The ratio of the gradient magnitude between the blurred edge and the reblurred edge becomes maximum at the edge location and it is used to calculate the value of σ	25
3.4	Calibration chart (a) and Depth-Defocus Curve (b). The σ is measured at the binary edge pattern, and the depth is measured from the known size of the checkerboard shown in (a).	27

3.5	Demonstration of Eq. (3.11). (a) is a low contrast edge pattern with 50% and 75% gray levels. (b) is a high contrast binary edge pattern. (d) is a face and (e) is a checkerboard. In (c), the green \times shows σ_m measured at the low contrast edge, the blue $+$ shows σ measured at the high contrast edge, the red line is the approximation of σ based on Eq. (3.10), and the black line is the approximation of σ_m based on Eq. (3.11). In (f), the green \times is σ_m measured on the face, and the blue $+$ is σ measured on the checkerboard.	29
3.6	Illustration of the metric scale (a) and the image velocity (b)	31
3.7	The chart used in Experiment 1, where (i) is the checkerboard used to compute the true metric scale, and (j) and (k) have the same edge patterns as (b) and (a) described in Fig. 3.5, respectively.	35
3.8	CANON EOS 650D (EOS Kiss X6i in Japan) camera with the EF-S 18-135mm f/3.5-5.6 IS STM lens used in the experiments.	35
3.9	The estimates of Λ (a), λ^i (b), σ_m^i (c), and the metric distance d^i (d) in Experiment 1. The blue lines show the estimates, the red lines show the ground truth, and the black line shows the measurement.	37
3.10	The desk environment used in Experiment 2. The red $+$'s indicated by arrows with letters 'l' and 'm' are two of the feature points where σ_m^i are measured. The green $+$'s show the other feature points used for the scale estimation. The checkerboard was placed to compute the true scale.	39
3.11	The estimates of Λ (a), λ^i (b), σ_m^i (c), and the metric distance d^i (d) at the point indicated in 'm' of Fig. 3.10. The blue lines show the estimates, the red lines show the ground truth, and the black line shows the measurement.	40
3.12	The camera poses and 3D point map reconstructed to the metric scale. The red line shows the camera trajectory reconstructed by the estimated scale. The green line shows the ground truth. The red dots show the estimated 3D locations of observed feature points. The green dots show the 3D locations of feature points on the black-and-white corners of the checkerboard.	41
3.13	The estimates of λ^i (a), σ_m^i (b), and the metric distance d^i (c) at the point indicated in 'l' of Fig. 3.10. The blue lines show the estimates, the red lines show the ground truth, and the black line shows the measurement.	43
3.14	Illustration of a method to avoid focal plan ambiguity by using the camera motion. The orange dots show the defocus blur amounts of feature points located on the near side of the focal plane. The blue dot shows the defocus blur amount of a feature point located on the far side of the focal plane. The direction of defocus blur change induced by a camera motion is a possible indicator to resolve the focal plane ambiguity problem.	44
4.1	Illustration of the scale difference caused by the scale drift in local regions of the map. The blue dots show the map points generated by monocular SLAM algorithm.	47
4.2	Illustration of Eq. (4.4). The blue dots show the corresponding feature points between the successive images, the orange lines show the edges at the feature points. The red ellipse shows the size of motion blur. ϕ is an internal angle formed by vectors \mathbf{b} and \mathbf{u} . For simplicity, it is assumed that $T_e = T_f$ in this figure.	49

4.3	Demonstration of Eq. (4.4). (a) shows the chart with a tilted binary edge pattern and a checkerboard. The chart was positioned to face the camera at a distance of two meters and moved from side to side with the velocity shown in (c). In (b), the blue dash line, the red solid line, and the green dot-dash line show σ_{mb}^i , σ_m^i , and σ_b^i . As expected, σ_m^i is nearly constant. The exposure time was 8 ms and the frame period was 33 ms.	50
4.4	Demonstration of Eq. (4.10). (a) and (b) show the charts with a low-contrast edge and a binary edge, respectively. In (c), the blue \times and the red $+$ show $\sigma_m^{i,j}$ measured on (a) and (b), respectively. In (d), the cyan \times and the magenta $+$ show the edge strength evaluated by the index $mg^{i,j}$ measured on (a) and (b), and the blue \times and the red $+$ show the edge strength evaluated by the proposed index $smg^{i,j}$ measured on (a) and (b), respectively.	54
4.5	The camera and lens used in Experiment 1. The field of view is about 37-degree width.	57
4.6	The rear camera on iPhone SE used in Experiment 2.	57
4.7	In (a) and (b), the green lines show the camera trajectory, and the blue dots show the point cloud of feature points generated by ORB-SLAM. The scale was reconstructed using the mean value of the scales computed using checkerboard patterns and shown in Table 4.1. Some turns of the trajectory used to capture (b) were sharper than the trajectory shown in (a).	59
4.8	The box plot showing the absolute errors between the estimated keyframe positions and the ground truth in the local regions CB (a), C2 (b), C3(c), and C4(d). The box lengths indicate the interquartile range (first to third quartiles). The line in the center of the boxes indicates the median value. The whiskers down to the minimum and up to the maximum.	62
4.9	$z^{i,j}$ vs $\sigma_m^{i,j}$ in local regions C3 (a) and C4 (c), and the examples of keyframes in C3 (b) and C4 (d). In (a) and (c), the cyan o's show all feature points, the blue x's show the feature points selected for the initial guess. Each blue line connects the same feature for different keyframes, which is selected for the optimization. The magenta, orange, and green lines show the approximations by $\sigma^{i,j} = D(z^{i,j})$ as results of the initial guess, the optimization, and the truth. In (b) and (d), the green x's show the feature points selected for the initial guess, and the red *'s show the feature points selected for the optimization. To be fair, feature points on the checkerboards were excluded for the optimizations.	63
4.10	$z^{i,j}$ vs $\sigma_m^{i,j}$ in local regions (CA(a), CB(c), C1(e), C2(g)) and the examples of keyframes (CA(b), CB(d), C1(f), C2(h)). In (a), (c), (e), and (g), the cyan o's show all feature points, the blue x's show the feature points selected for the initial guess. Each blue line connects the same feature for different keyframes, which is selected for the optimization. The magenta, orange, and green lines show the approximations by $\sigma^{i,j} = D(z^{i,j})$ as results of the initial guess, the optimization, and the truth. In (b), (d), (f), and (h), the green x's show the feature points selected for the initial guess, and the red *'s show the feature points selected for the optimization. To be fair, feature points on the checkerboards were excluded for the optimizations.	64

4.11	The map and camera poses reconstructed by the estimated scale in C2 (a), C3 (b), and C4 (c). The blue lines show the trajectory generated by ORB-SLAM. The red lines show the trajectory corrected by the estimated scales. The green lines show ground truth obtained from the checkerboard detection algorithm. The point clouds indicated by arrows are the map points on the corresponding checkerboards.	65
4.12	The map and camera poses generated by using iPhone SE. (a) shows the office environment. (b) is the map and the camera trajectory reconstructed by iPhone SE. In (b), the green line is the trajectory and blue dots are the map points generated by ORB-SLAM.	66
4.13	$z^{i,j}$ vs $\sigma_m^{i,j}$ in local regions CI (a) and CII (c), and the example of keyframes in CI (b) and CII(d). In (a) and (c), the cyan o's show all feature points, the blue x's show the feature points selected for the initial guess. Each blue line connects the same feature for different keyframes, which is selected for the optimization. The magenta, orange, and green lines show the approximations by $\sigma^{i,j} = D(z^{i,j})$ as results of the initial guess, the optimization, and the truth. In (b) and (d), the green x's show the feature points selected for the initial guess, and the red *'s show the feature points selected for the optimization. To be fair, feature points on the checkerboards were excluded for the optimizations.	67
4.14	The box plot showing the absolute errors between the estimated keyframe positions and the ground truth in the local region CII.	68

List of Tables

3.1	Calibration parameters	30
3.2	Parameters for Experiment 2	39
4.1	Scale and Scale Drift in Experiment 1	58
4.2	Parameters used in DfD for Experiment 1	60
4.3	Threshold values used in the optimization for Experiment 1	60
4.4	Error in Scale Estimate in Each Area (%) in Experiment 1	60
4.5	RMSE of keyframe positions (mm) in Experiment 1	61
4.6	Error in Scale Estimate in Each Area (%) in Experiment 2	66
4.7	RMSE of keyframe positions by iPhone SE (mm) in Experiment 2	66
4.8	Parameters used in DfD for Experiment 2	68
4.9	Threshold values used in the optimization for Experiment 2	68

Acronyms & Abbreviations

1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
BRIEF	Binary Robust Independent Elementary Features
CAS	Centre for Autonomous Systems
CoC	Circle of Confusion
CPU	Central Processing Unit
DfD	Depth from Defocus
DfF	Depth from Focus
EKF	Extended Kalman Filter
FAST	Features from Accelerated Segment Test
IMU	Inertial Measurement Unit
KLT	Kanade-Lucas-Tomasi
PSF	Point Spread Function
PTAM	Parallel Tracking and Mapping
ORB	Oriented FAST and Rotated BRIEF
RGB-D	Red, Green, Blue, and Depth

RMSE	Root mean square error
ROI	Region of Interest
SfM	Structure from Motion
SLAM	Simultaneous Localization and Mapping
UKF	Unscented Kalman Filter
UTS	University of Technology Sydney

Nomenclature

General Notations

A	Aperture diameter of a lens
A_m	Amplitude of a step function
\square^*	Unconstrained \square in the two-step projection method
B	Unknown offset
\mathbf{b}	Motion blur vector
b_δ	Distance from the image plane to a virtual plane where the rays from an out-of-focus point converge
b_f	Distance from the lens center to the image plane along the optical axis
c	Diameter of the circle of confusion
$\mathbf{c}[\cdot]$	Constraint function for equality state constraints
d	Distance from the lens center to a point along the optical axis
$D(\cdot)$	Depth-Defocus function
d_f	Distance from the lens center to a focal plane along the optical axis
Δt	Length of time between discrete steps
e_{thl}, e_{thh}	Threshold values to select features with strong edges
$\boldsymbol{\epsilon}_k = [\epsilon_{\Lambda_k} \ \epsilon_{\lambda_k^i} \ \epsilon_{\sigma_k^i}]^T$	Process noise vector of EKF
$\boldsymbol{\eta}_k = [\eta_k^i]^T$	Observation noise vector of EKF
\mathbf{F}	State transition matrix
f	Focal length of a lens
$G(\cdot, \cdot)$	Gaussian-shaped point spread function

γ	Camera specific constant to approximate c with σ
\mathbf{H}	Observation matrix
\mathbf{H}_c	Constrained matrix
$\hat{\square}$	Prediction value of \square
$I(\cdot, \cdot)$	Blurred image
$I_f(\cdot, \cdot)$	Sharp image
I_x	Gradient magnitude along x_e axis of $I(x_e, y_e)$
I_y	Gradient magnitude along y_e axis of $I(x_e, y_e)$
\square^i	i-th feature point
$\square^{i,j}$	i-th feature point seen from j-th keyframe
k	Time (discrete step)
Λ	Scale factor which defines the relationship between the metric map and the estimated geometry
λ	Texture correction factor for DfD
Λ_{ini}	Initial guess of Λ
λ_{ini}	Initial guess of λ
m	Number of keyframes
N	Number of observed points
N_c	F-number of a lens
ν	Innovation vector
\mathbf{P}	State covariance matrix
$\mathbf{p} = [x \ y \ z]^T$	Non-scaled feature location in camera coordinates
$\mathbf{p}_w = [x_w \ y_w \ z_w]^T$	Non-scaled feature location in world coordinates
ϕ_{\square}	Calibration parameters for Depth-Defocus function
\square^+	First-step constrained \square in the two-step projection method
\mathbf{Q}	Covariance matrix of the process noise
R	Gradient ratio of input image and reblurred image
\mathbf{R}	Covariance matrix of the observation noise
r	Index to evaluate the constancy of λ
\mathbf{R}_c	Covariance matrix of the noise or the extent of constraint violations

r_{thl}, r_{thh}	Threshold values to select features with constant λ
\mathbf{S}	Innovation covariance matrix
\mathbf{S}_c	Constrained innovation covariance matrix
σ	Standard deviation of Gaussian-shaped PSF
σ_b	σ for motion blur
σ_m	Measured σ for defocus blur
σ_{mb}	Composite σ of σ_m and σ_b
σ_r	Standard deviation of reblurred Gaussian-shaped PSF
smg	Index to evaluate edge strength
t	Time (continuous)
θ	Edge direction angle
$u(\cdot)$	Step function
$\mathbf{u} = [f_u \ f_v]^T$	Optical flow vector
v	Image velocity
\mathbf{W}	Filter gain vector
\mathbf{W}_c	Constrained filter gain vector
x_e	X-axis on an image where the edge is placed at $x_e = 0$
$\mathbf{x}_k = [\Lambda_k \ \lambda_k^i \ \sigma_{m,k}^i]^T$	State vector of EKF
y_e	Y-axis on an image where the edge is placed at $y_e = 0$
z	Non-scaled distance from the lens center to a point along the optical axis
$\mathbf{Z}_k = [z_k^i]^T$	Observation vector of EKF
z_{th}	Threshold value to select features nearby camera
$\boldsymbol{\zeta}_k = [\zeta_k^i]^T$	Noise vector for the extent of constraint violations

Operations

$\square * \square$	Convolution operator
$ \cdot $	Absolute value
$\sqrt{\cdot}$	Square root
$\exp(\cdot)$	Exponential
\square^2	Square of \square

∇	Gradient operator
$\ \cdot\ $	Norm
$\dot{\square}$	First derivative of \square
$E[\cdot]$	Expectation
$\Sigma(\cdot)$	Summation
$\operatorname{argmin}[\cdot]$	Argument of the minimum

State Transitions

$\square_{k-1 k-1}$	Previous state
$\square_{k k-1}$	Predicted current state
$\square_{k k}$	Updated current state

Transforms

\square^T	Transpose
\square^{-1}	Inverse
\boldsymbol{r}	Rotation matrix of keyframe pose
\boldsymbol{t}	Translation vector of keyframe pose
\boldsymbol{t}_t	Ground truth of \boldsymbol{t}
$\boldsymbol{T} = [\boldsymbol{r} \boldsymbol{t}]$	Transformation matrix of keyframe pose