Tina Memo No. 2017-001
Presented at ISCV 2016.

Quantitative Performance Optimisation for Corner and Edge based Robotic Vision Systems: A Monte-Carlo Simulation.

Jingduo Tian, Neil Thacker, and Alexandru Stancu.

Last updated
17 / 11 / 2016

Imaging Science and Biomedical Engineering Division,
Medical School, University of Manchester,
Stopford Building, Oxford Road,
Manchester, M13 9PT.

Abstract

Corner and edge based robotic vision systems have achieved enormous success in various applications. To quantify and thereby improve the system performance, the standard method is to conduct cross comparisons using benchmark datasets. Such datasets, however, are usually generated for validating specific vision algorithms (e.g. monocular SLAM [1] and stereo odometry [2]). In addition, they are not capable of evaluating robotic systems which require visual feedback signals for motion control (e.g. visual servoing [3]). To develop a more generalised framework to evaluate ordinary corner and edge based robotic vision systems, we propose a novel Monte-Carlo simulation which contains various real-world geometric uncertainty sources. An edge-based global localisation algorithm is evaluated and optimised using the proposed simulation via a large scale Monte-Carlo analysis. During a long-term optimisation, the system performance is improved by around 230 times, while preserving high robustness towards all the simulated uncertainty sources.

1 Introduction

Robotic vision systems with the capabilities of visual perception, reasoning and associated high level control have attracted great research interests for decades. A number of state-of-the-art studies have achieved enormous success in various vision applications, such as visual SLAM [1] [4], autonomous localisation [5] [6] [7], long-term navigation [8] and pose estimation [2]. Meanwhile, many studies have reached real-time efficiency [1] [2], robustness on time variation [8], vast-scale [5], all-weather conditions [6] and plug-n-play flexibility [2].

The core of majority robotic vision systems is visual tracking and matching between image evidence, as it provides extrinsic reference signals from the environment, which compensate or rectify the intrinsic measurement error (e.g. odometry). To avoid computational burden while preserving image distinctiveness, image features are used in visual tracking and matching [4] [5] [9] [10]. Recently, salient point features such as SIFT [10] [11] and SURF [12] are widely used. Meanwhile, corners and edges are also successfully utilised [4] [5] [6] [9], providing efficient and invariant feature distinctiveness [13] [14].

To quantify the performance of robotic vision systems and optimise the parameter settings, the standard method is to conduct cross comparisons using benchmark datasets [15] [16]. The benchmark datasets, however, are usually pre-acquired image sequences for validating specific vision algorithms. Consequently, they are unsuitable for evaluating robotic systems which require visual feedback signals for motion control (e.g. visual servoing [3]).

This paper focuses on quantitative performance evaluation and optimisation of generic corner and edge based robotic vision systems. To develop a generalised framework, we propose a novel Monte-Carlo simulation which contains various real-world geometric uncertainties. The uncertainties (detailed in Section 3.2) are implemented as plug-in functions to satisfy different algorithm specifications. Notably, many of these uncertainties are not considered in the related work [17] [18] [19] or mainstream computer vision simulators (e.g. ROS and OpenCV).

We use the proposed simulation to evaluate and optimise an edge-based global localisation algorithm via a large scale Monte-Carlo analysis. The merit of Monte-Carlo analysis is to provide reliable statistics for performance quantification. This allows direct comparison between algorithm structures, to guide further system optimisations. In addition, Monte-Carlo analysis requires a vast number of experiments to be conducted using various control parameter settings, which may not be achievable in real-world trials. During a long-term optimisation, the system performance is improved by around 230 times, while preserving high robustness towards all the simulated uncertainty sources. The novelty of this work is the adoption of Monte-Carlo analysis in the optimisation of a robotic vision system, using the proposed simulation environment.

The rest of this paper is arranged as follows: Section 2 provides a literature review on the robotic vision simulations, identifying the uncertainty sources they consider. Section 3 details the construction of our simulation environment while explaining the derivation of each simulated uncertainty source. The optimisation of an edge-based global localisation algorithm is reported in Section 4. Finally, Section 5 concludes the proposed work and the future plan.

2 Related Work

Simulation based design has been the cornerstone of engineering for a long time. For sufficiently complex systems such as robotic vision systems, simulation provides the most efficient way to investigate performance and design choices. However, a number of simulations are only capable of evaluating specific robotic designs, such as stereo-based fruit positioning [17], legged football robots [20], colour-based object positioning [21] and coloured model

recognition [22]. Other work [19] [23] utilise pre-acquired or synthesised image sequences, which cannot generate visual feedback signals for robotic motion simulation.

Visual perturbation is another important factor to be considered in robotic vision simulations, as they emulate the real-world uncertainties to challenge the system robustness in more realistic scenario. However, some simulations are designed as noise-free environments [21] [22] [24] which contradict the real-world conditions. In [20], elementary image variations are considered on the RGB acquisition of a web cam. Several camera uncertainties are simulated on rendered 3D objects in [18], providing a more realistically rendered environment. Another study [19] uses thermal noise, vignetting effect and chromatic aberration to perturb synthetic image pairs for algorithm evaluations. In addition, camera calibration error and stereo matching error are simulated in [17] to test a vision-based manipulator positioning system.

In contrast to the related work, the proposed simulation is able to evaluate generic corner and edge based robotic vision systems, providing visual feedback signals for motion control. It also contains a variety of visual uncertainty sources, some of which have never been proposed in the literature or off-the-shelf robotic simulation software. This simulation further enables a large scale Monte-Carlo analysis for the quantitative optimisation of robotic vision systems.

3 Simulation Environment and Uncertainty Sources

The construction of a 3D simulation environment is detailed in this section. Multiple visual uncertainty sources are modelled and implemented as plug-in functions in order to produce realistic data variation. The purpose of such a simulation is to emulate real-world geometric appearances to evaluate generic corner and edge based robotic vision systems via a large scale Monte-Carlo analysis. Therefore, wire-frame object representations are used in order to avoid the computational burden of 3D rendering. Whilst this paper presents a manually constructed wire-frame environment, in principle this could be constructed automatically using visual SLAM or visual odometry algorithms (e.g. [1] [2]).

The API that we use is the TINA vision system [25] which has been developed for three decades, providing a wide range of functionalities, including image handling, feature detection, GUI development and data transmission. It also provides an integrated set of high-level analysis techniques for machine vision, such as on-line camera calibration, 2D object recognition and 3D object localisation.

3.1 3D Wire-Frame Representation

The proposed simulation environment is similar to a CAD model of a working space (see Fig. 1), which has the volume of around 20m $\times$ 15m $\times$ 2.5m. By acquiring images from the working space and applying *Canny* detection, we identify the visible edges from the real data. The configuration of the *Canny* detector is specifically defined to obtain the most reliable edges while suppressing the image noise. These visible edges are then considered as wire-frame representations of real-world objects and are manually measured to an accuracy of one centimetre. The wire-frame models are defined as the ground truth, providing geometric benchmark for algorithm evaluation. In this simulation, a total of 7000+ geometric features are measured and modelled. Multiple highly similar scene components (e.g. door frames) are also created as templates and used where possible, to test the robustness against worst case geometric ambiguities.

To automatically eliminate occluded edges, a view-dependency file is created in which a sequence of view-points are pre-defined as reference points. Around each reference point, edges that are expected to be visible are displayed during the robot motion. This mechanism allows us to adequately approximate the visible geometry at any location.

3.2 Visual Uncertainty Sources

Due to data quality limitations, processing errors are inevitably caused by individual vision algorithms. Understanding these errors benefits system design and helps to avoid possible failures in real-world implementations. Therefore, we investigate a variety of visual uncertainty sources which have significant effect on corner and edge based vision algorithms. The magnitudes of each uncertainty sources are not static, their values are determined by the specification of different environment and algorithm settings. The default values are obtained by experimentation and algorithm analysis from previous studies. These values may vary during the evaluation of a system design, in order to comprehensively test the system robustness under different environment and algorithm specifications.

Image Noise

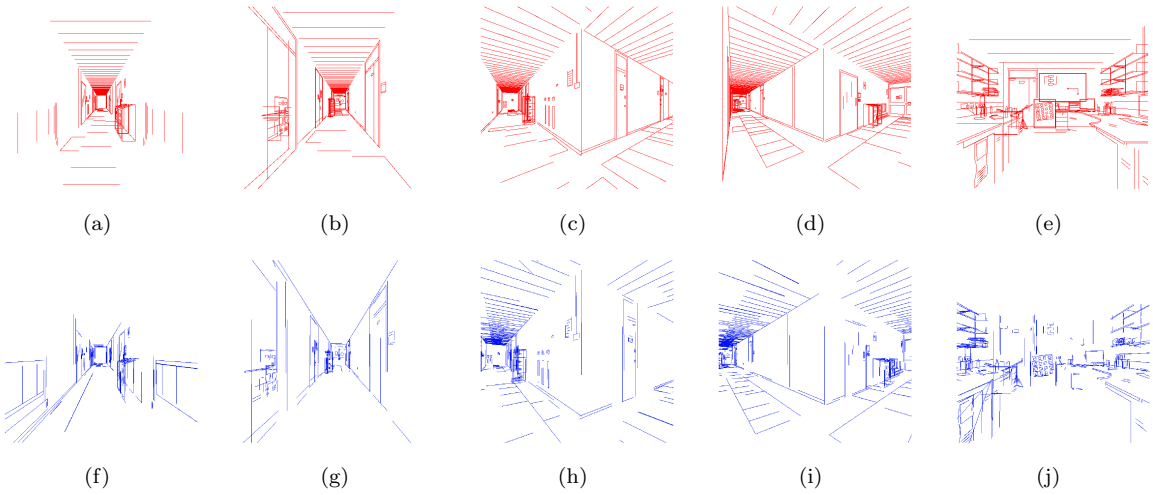

Figure 1: *The original environment (a - e) and the perturbed environment (f - j).*

Grey scale image noise affects vision algorithms, including edge detection and subsequent geometrical approximation algorithms. These effects are approximated by the following lateral edge shifting and geometric approximation error.

Lateral Edge Shifting and Edge Detection Loss

Edges are usually represented by the discontinuities of image brightness, therefore they are highly susceptible to illumination. Due to blurring or being lack of contrast with the background, genuine edges are not always detectable. A part of an extended edge feature can either be totally undetected or be detected with a biased orientation and location. When the detected edge locations are compared with the object model predictions, some systematic problems become apparent. Under distant illumination conditions (e.g. *Lambertian* reflectance models), a planar surface with homogeneous texture is assumed to have the same grey-level. Therefore, illumination changes apply systematically along extended boundaries of the surfaces, resulting in lateral edge shifting [26].

Given an image, edge detection loss is defined as the percentage of genuine edges that cannot be repeatedly detected by an edge detector. A quantitative verification using a power-law model proposed in [27] is utilised to quantify the edge detection loss of the *Canny* edge detector. This approach manually creates 16 wire-frame models of a selection of referenced man-made objects constructed from a variety of materials. These wire-frame models are optimally projected onto their corresponding image edges, and the pixel alignments between model contours and image edges are quantitatively verified.

The verification scores indicate the portion of artificially defined object contours that are detectable by *Canny*. Due to subjective definition of contour locations, the verification scores are inevitably deviated from their true values. By performing a linear interpolation on the data reported in [26], the bias is expected to be largely reduced. The result provides a quantitative approximation of the average *Canny* edge detection loss (11.25% with a range from 0% to 31.25%). In our simulation, the uncertainty of edge detection loss randomly removes ψ_e (by default 11.25%, from 0% to 31.25%) of the length on each visible features. The uncertainty of lateral edge shifting then drifts each feature laterally within a range ψ_l (by default 3 $\pm$1.5 pixels), as suggested in [26].

Geometric Approximation Error

Geometric approximation fits distinctive image pixels into polygonised features, thereby allowing high-level feature analysis. The error on geometric approximation is the residual between a fitted geometric feature and its corresponding image pixels. A model that estimates the error distribution of line fitting on *Canny* edges has been proposed and validated in [28]. Circular *Gaussian* uncertainty regions are defined at the end points of a fitted line, representing the error on geometric approximation. The standard deviation of this *Gaussian* region is determined by line fitting threshold, which is used to define the break point of a line when fitting to the image pixels. In this work, the threshold is set to 0.15 pixels by default. Therefore, we simulate circular *Gaussian* noise regions with the standard deviation of ψ_g (by default 0.15 pixels $\pm$0.05) on each of the line end points.

Corner Detection Loss and Corner Match Error

Corner detection loss is the proportion of corners that cannot be re-detected in image 2 given they have been

detected in image 1, due to the change on local image re-projection. Corner match error is the amount of incorrect corner matches that are accepted as correct matches, due to ambiguities on image patches around competing matching candidates.

In this work, the adopted corner matching algorithm [29] is based on the *Harris* and *Stephens* corner detector [30]. As reported in [29], the typical repeatability of detecting the same corner in both images is around 85% (this varies for algorithm configurations and datasets). In the matching procedure, a region in image 2 is calculated using stereo geometry to find the matching candidates given the corner in image 1. As reported in [29], 1% of the accepted corner matches are incorrect. In the simulation, corner features are manually defined inside the 3D virtual environment. The uncertainty of corner detection loss randomly drops ψ_{cd} (15% $\pm$10%) corners in both images. The uncertainty of corner match error then randomly drifts ψ_{cm} (1% $\pm$1%) corners inside circular regions with a 10 pixel radius to simulate mismatches.

Stereo Match Error

For an image pair, stereo matching is performed along epi-polar constraints within a disparity range, matching geometric features with the largest correspondence. Due to huge ambiguities, however, the features that lie along epi-polar lines cannot be properly matched. Some algorithms [31] therefore inherently ignore these features before conducting the match. Stereo match error is the amount of incorrect matches that are accepted by a stereo matching process, and the elimination of those features parallel to epi-polar lines.

In our simulation, a 'stretch correlation' algorithm [31] is used for stereo matching between geometric features. According to a quantitative analysis in [31], the typical mismatch rate is around 1% with a default disparity range of 20 pixels on each side. This algorithm automatically eliminates all the features which are $\pm 5^{o}$ parallel to their intersected epi-polar lines. For simplicity, the simulation system removes all the features that are $\pm 5^{o}$ parallel to a horizontal line. Then, ψ_s (1% varying from 0% to 10%) of the remaining features are shifted randomly within the disparity range of 20 pixels in both images.

Camera Calibration Error

Camera calibration is used to estimate the intrinsic and extrinsic parameters of camera models from image evidence. We consider an automatic stereo calibration approach [32] that uses matched stereo corner pairs from generic images. Using real data, we derive a covariance matrix with respect to the calibrated parameters, and use it to specify the parameter error correlations.

A stereo model is defined as $S = F(f, a_x, a_y, o_x, o_y, k, R, T)$, with focal length f, aspect ratio (a_x, a_y), optical centre (o_x, o_y), radial distortion coefficient k, rotation matrix R and translation matrix T. A unified least-square cost function is then followed as,

$$\chi_t{}^2 = (a - a_t)^T {C_a}^{-1} (a - a_t) + \sum_i (y_i - \phi_i(a_t))^T {W_i}^{-1} (y_i - \phi_i(a_t)) \quad (1)$$

with a representing the true stereo parameters and a_t denoting the parameter estimations at the t^{th} iteration; $\chi_t{}^2$ represents the cumulative residual between the true parameters and the estimations; y_i is defined as the image projections of a matched corner pair where $\phi_i(a_t)$ is the projection estimation given an epi-polar model ϕ_i with respect to an estimated parameter set a_t. The covariance matrix C_a is obtained from the Minimum Variance Bound (MVB). By conducting singular value decomposition (SVD) on C_a, the correlations and inverse variances of different parameter errors are derived. Using these, the predicted error distributions on individual parameters are calculated and simulated. The calibration uncertainty magnitude is defined as a variable ψ_c (by default 1, from 0.1 to 10).

4 System Optimisation via A Monte-Carlo Analysis

The Monte-Carlo analysis provides quantitative performance assessment of a system design, by repeating specified experiments under finite uncertainty sources. The accuracy of such analysis follows a binomial observation error model, which normally requires over 900 samples in order to achieve a 0.5% confidence interval. The assessment proves evidence to guide and verify the design improvement, resulting in an optimised system specification.

In this section, we apply a large scale Monte-Carlo analysis throughout the long-term optimisation of an edge-based global localisation algorithm [33]. The optimisation approaches include algorithm improvement, parameter tuning and system structure upgrade. The proposed simulation environment and the associated uncertainty sources are utilised via the TINA API software [25]. The simulations are conducted under Cent-OS Linux 7 system, with an Intel Xeon(R) E5-2630 v2 CPU working at 2.60GHz and 32 GB ROM.

The global localisation algorithm relies upon a pre-acquired topological map which encodes edge-based environment representation using Pairwise Geometric Histograms (PGH) [34]. The topological map is obtained by a robot travelling along a path, capturing and encoding edge-based patterns at various locations. Given a sampled scene at an arbitrary location, the global localisation algorithm estimates its most likely position by conducting PGH matchings throughout the topological map. Each PGH matching is ranked using a *Bhattacharyya* distance [35] indicating the confidence of a position estimation. All the estimations are refined using a filtering scheme until only one remains, leaving the corresponding topological node as the location of the robot.

In the simulation, the performance of this algorithm is defined as localisation accuracy and computational time. The localisation accuracy is represented by the percentage of correct location estimations among all the attempts. Specifically, we define an estimation as 'success' if it is within 1 meter from the true position. The computational time includes the time consumption from the start of a localisation attempt until the end. All the computational time in this paper are the average values of multiple simulation trails under the same conditions.

During a 4-month system optimisation, 77 sets of totally 74,110 simulations have been conducted and used for Monte-Carlo analysis, with each set of simulations contains approximately 900 samples to achieve a small observation error. The localisation accuracy and computational time regarding a selection of experimental trials is summarised in Fig. 2. The performance improvement is due to algorithm improvement, parameter tuning and system structure upgrade. It shows that, from the initial version (28.44%, 1356.35s) to the optimised algorithm (92.82%, 17.62s), the performance is improved around 230 times. Meanwhile, the robustness towards all the simulated uncertainty sources is also maintained.

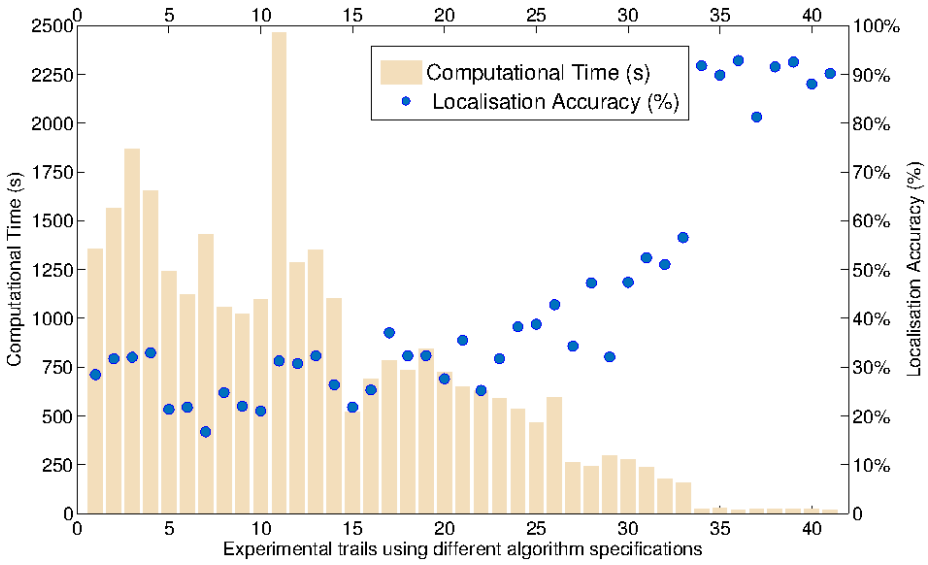

Figure 2: *The algorithm performance over a long-term optimisation.*

Figs. 4 (a) (b) (c) further demonstrate a parameter tuning process of the optimised algorithm using Monte-Carlo analysis, where the best PGH resolution and the threshold value are determined. Fig. 4 (d) shows an evaluation on the effect of individual uncertainty source towards the algorithm performance, to guide a deep investigation on specific algorithm components.

5 Conclusion

This paper presents a Monte-Carlo simulation which is suitable for quantitative performance optimisation of corner and edge based robotic vision algorithms. The merit of this work is the modelling and implementation of real-world geometric uncertainties in a novel wire-frame environment, which allows repeated experiments to be conducted and thereby supports a large scale Monte-Carlo optimisation. The validity of this simulation is demonstrated by a long-term optimisation upon an edge-based robotic vision algorithm, resulting in a performance improvement of around 230 times while reserving robustness towards all the uncertainties. The use of Monte-Carlo analysis in quantitative optimisation of a robotic vision system contributes the leading novelty of this work.

The future work include a joint performance validation on the optimised localisation algorithm using real image data and the simulation environment. We aim to use datasets containing sufficient number of images to develop

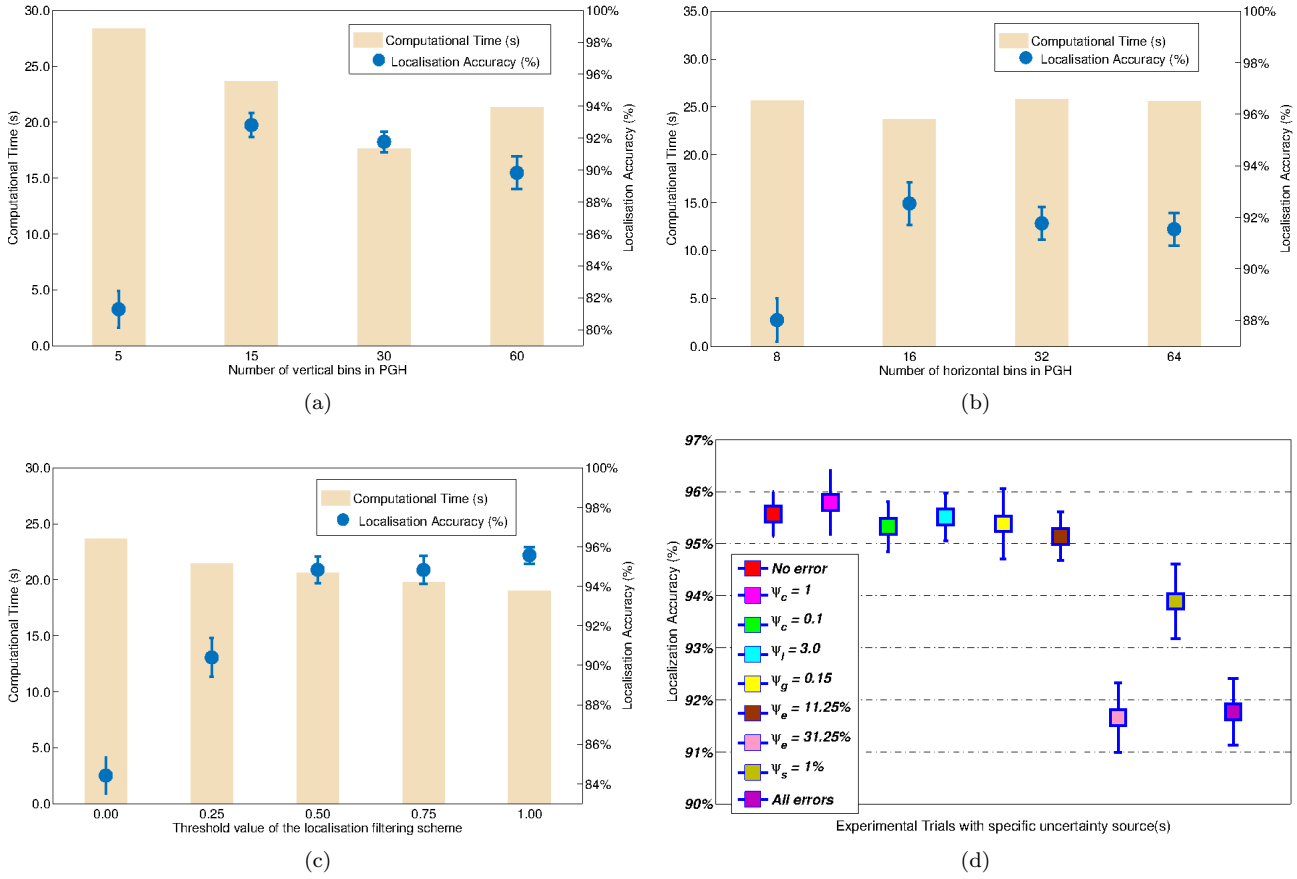

Figure 3: *Parameter tuning and an investigation on individual uncertainty sources.*

a topological map of an environment, and use this dataset for performance evaluation. Once the performance has been validated using image datasets, a real-world implementation will be conducted using specifically designed robotic systems.

References

[1] Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: ECCV'14. (2014) 834–849

[2] Napier, A., Sibley, G., Newman, P.: Real-time bounded-error pose estimation for road vehicles using vision. In: ITSC'10. (2010) 1141–1146

[3] Fang, Y., Liu, X., Zhang, X.: Adaptive active visual servoing of nonholonomic mobile robots. IEEE Transactions on Industrial Electronics **59** (2012) 486–497

[4] Concha, A., Civera, J.: Using superpixels in monocular slam. In: ICRA'14. (2014) 365–372

[5] Linegar, C., Churchill, W., Newman, P.: Work smart, not hard: Recalling relevant experiences for vast-scale but time-constrained localisation. In: ICRA'15. (2015) 90–97

[6] Linegar, C., Churchill, W., Newman, P.: Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera. In: ICRA'16. (2016) 787–794

[7] Pascoe, G., Maddern, W., Newman, P.: Robust direct visual localisation using normalised information distance. In: BMVC'15. Volume 3. (2015) 4

[8] Churchill, W., Newman, P.: Practice makes perfect? managing and leveraging visual experiences for lifelong navigation. In: ICRA'12. (2012) 4525–4532

[9] Eade, E., Drummond, T.: Edge landmarks in monocular slam. In: BMVC'06. (2006)

[10] Johns, E., Yang, G.: Global localization in a dense continuous topological map. In: ICRA'11. (2011) 1032–1037

[11] Zhang, H.: Borf: Loop-closure detection with scale invariant visual features. In: ICRA'11. (2011) 3125–3130

[12] F. Dayoub, G.C., Duckett, T.: A sparse hybrid map for vision-guided mobile robots. In: ECMR'11. (2011) 213–218

[13] Pinel, J.: Biopsychology. Pearson Education (1997)

[14] D, X., Chantler, M.J.: Texture similarity estimation using contours. In: BMVC'14. (2014)

[15] Xingshuai D, Xinghui D, J.D.: Monocular visual-imu odometry: A comparative evaluation of the detector-descriptor based methods. In: CVRSUAD'16. (2016)

[16] Gauglitz, S., Höllerer, T., Turk, M.: Evaluation of interest point detectors and feature descriptors for visual tracking. International journal of computer vision **94** (2011) 335–360

[17] X. Zou, H.Z., Lu, J.: Virtual manipulator-based binocular stereo vision positioning system and errors modelling. Machine Vision and Applications **23** (2012) 43–63

[18] Kučiš, M.: Simulation of camera features. In: the 16th Central European Seminar on Computer. (2012) 117–123

[19] A. Hinkenjann, T.R., Millberg, J.: Real-time simulation of camera errors and their effect on some basic robotic vision algorithms. In: CRV'13. (2013) 218–225

[20] K. Asanuma, K. Umeda, R.U., Arai, T.: Development of a simulator of environment and measurement for autonomous mobile robots considering camera characteristics. In: RoboCup 2003: Robot Soccer World Cup VII. Springer (2004) 446–457

[21] K. Okada, Y.K., Kanehiro, F.: Rapid development system for humanoid vision-based behaviors with real-virtual common interface. In: IROS'02. (2002) 2515–2520

[22] I. Ulusoy, U.H., Leblebicioglu, K.: 3d cognitive map construction by active stereo vision in a virtual world. In: ISCIS'04. Springer (2004) 400–409

[23] Peris, M., Martull, S., Maki, A., Ohkawa, Y., Fukui, K.: Towards a simulation driven stereo vision system. In: ICPR'12. (2012) 1038–1042

[24] Klaser, R., Wolf, D.: Simulation of an autonomous vehicle with a vision-based navigation system in unstructured terrains using octomap. In: SBESC'13. (2013) 177–178

[25] TINA: Tina open source computer vision development environment. http://www.tina-vision.net (2016) Accessed: 2016-08-08.

[26] Coupe, S.: Machine Learning of Projected 3D Shape. PhD thesis, University of Manchester (2009)

[27] Coupe, S., Thacker, N.: Quantitative verification of projected views using a power law model of feature detection. In: CRV'08. (2008) 352–358

[28] A. Ashbrook, NA. Thacker, P.R., Brown, C.: Robust recognition of scaled shapes using pairwise geometric histograms. In: BMVC'95. (1995) 503–512

[29] Thacker, N., Courtney, P.: Statistical analysis of a stereo matching algorithm. In: BMVC'92. (1992) 316–326

[30] Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey vision conference. Volume 15. (1988) 147–152

[31] Crossley, S.: Robust Temporal Stereo Computer Vision. PhD thesis, University of Sheffield (2000)

[32] Thacker, N., Mayhew, J.: Optimal combination of stereo camera calibration from arbitrary stereo images. Image and vision computing **9** (1991) 27–32

[33] Tian, J.: Quantitative optimisation of a vision-based robotic localisation and navigation algorithm. http://www.tina-vision.net (2016) [Online]. Available: http://www.tina-vision.net/docs/memos.php. [Accessed 16-May-2016].

[34] NA. Thacker, P.R., Yates, R.: Assessing the completeness properties of pairwise geometric histograms. Image and Vision Computing **13** (1995) 423–429

[35] FJ. Aherne, N.T., Rockett, P.: The bhattacharyya metric as an absolute similarity measure for frequency coded data. Kybernetika **34** (1998) 363–368