Tina Memo No. 2015-007

Internal.

Simulation of Wire-frame Object Representations for a 3D Robotic Vision System.

Jingduo Tian, Neil Thacker and Alexandru Stancu.

Last updated
21 /6 / 2015

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

Simulation of Wire-frame Object Representations for a 3D Robotic Vision System

Jingduo Tian
School of Electrical and Electronic Engineering
University of Manchester, Manchester, U.K.
jingduo.tian@manchester.ac.uk

Neil Thacker, Alexandru Stancu
University of Manchester, Manchester, U.K.
neil.a.thacker@manchester.ac.uk
alexandru.stancu@manchester.ac.uk

Abstract

This paper presents a Monte-Carlo framework for developing a vision-based mobile robotic system which is capable of learning and recognising scenes in an unknown man-made environment. This is done in order to assess the robot performance without depending physically on the machine,

Contemporary vision-based simulations are likely to utilise rendered 3D objects as representations of an environment. To be realistic, they tend to use camera uncertainty sources to perturb the 2D camera projections of the environment. However, these simulations usually aim to assess feature detection and matching algorithms, so their major purpose is in simulating images.

We present an alternative 3D vision-based robotic simulation with wire-frame object representations and associated quantified uncertainty sources. This approach requires a quantitative understanding of the statistical properties of data generated by vision algorithms (such as object recognition, calibration and stereo matching). We firstly specify the computational characteristics of all the functional modules, then build a realistic 3D vision-based robotic simulation. In this paper, the modelling and simulation of uncertainty sources are detailed and their effect on a simulated world is demonstrated.

1. Introduction

Modern robotic applications require robots to undertake increasingly demanding tasks, such as navigation and mapping in an unknown environment. Such tasks require enhancements on current robotic autonomy, which should further include learning and recognition for generic environments. We aim to design a robotic system which is capable of perceiving and exploring an unknown man-made environment, constructing a map which is suitable for visual feedback control. One major problem is the extraction of spatial information from visual features in the world, and to use this information to predict their topological relationships. Once this problem is solved, robot localisation can be accomplished by recognising (and measuring the distance to) configurations of features, thereby supporting robot navigation.

The cornerstone of engineering is simulation based design. Simulations and designs are developed together in order to investigate ongoing challenges in the construction of working systems. For sufficiently complex systems, simulation provides the most efficient way to investigate performance and design choices, and to iterate on modifications to proven designs. New circumstances generate the need for modified simulations, to keep pace with real world problems. In robotics research too, researchers design simulated virtual worlds with a certain level of reality, aiming to assess the performances of various robotic algorithms. Some of the designed simulations generate rendered images of 3D environments to test vision-based robotic algorithms without considering real-world uncertainties (*e.g.* image noise) [1, 2, 3]. Such a lack of consideration leads to an issue that any vision-based robotic algorithm working well in these simulations will not be able to conduct the same performance in real-world tasks. To fill the gap, [4] establishes several camera uncertainty sources in a rendered 3D robotic simulation. The uncertainties include camera blurring, image noise and image distortion. The objective of this simulation is to develop and assess vision-based robotic algorithms for Robocup four legged robots [5] under realistic conditions. [6] simulates more camera uncertainties such as chromatic aberration and vignetting. This simulation aims to render images of a 3D generated world as realistically as possible. However, it requires a very high computation speed when the simulation is used to provide a real-time rendering. In order to acquire enough computation speed in realistic 3D rendering, [7] uses a GPU to share the workload with CPU, but it does not reduce the total workload. [8] simulates camera errors on a sequence of synthetic images and assesses the effects of these errors on some vision-based algorithms. A pre-defined image sequence without

optical errors is required by this simulation as input data, thus it can not generate real-time scenes according to the motion of a virtual robot. [9] presents a 3D rendered simulation of a vision-based manipulator positioning system. This system uses stereo vision to acquire the position of an object relative to a manipulator. System imperfections such as camera calibration error and stereo match error are simulated, but more uncertainties including image noise and illumination variation are not considered. In [10], a collision avoidance algorithm using laser scanner is tested in a 2D planar virtual world. Several uncertainty sources on laser scanner are simulated to perturb the laser range measurements, and thereby affect the path planning result of the collision avoidance algorithm.

A possible solution of the previous localisation and recognition problem is to generate a database of encoded visual landmarks from extracted image features (*e.g.* edges), with each landmark accompanied by an estimation of robot position.

The pairwise geometric histogram (PGH) [11, 12, 13, 14] is an appropriate descriptor for landmark generation, it encodes the probability of geometric co-occurrences between an edge feature and the other edges that define an object [14]. The underlying principle of the PGH is that edge features are the most informative features defining the content and location of an object [15]. However, it is well known that simple geometric descriptions of edges can be highly variable and quite unsuitable for geometric reasonsing, due to image formation effects, such as illumination and occlusion, but also due to uncertainties in edge grouping and curve fitting.

The PGH is therefore designed to provide a high functional degree of invariance to practical uncertainties, including illumination variation, image noise, in-plane image orientation, image positioning, edge fragmentation, curve approximation, background clutter and pattern occlusion [16]. PGH's can also exploit the redundancy available due to the presence of multiple linear features Each linear segment of the wireframe description of a scene, including parts of curves, can be used a the reference for the generation of a PGH, However, the additional properties (listed above) will result in more robustness than systems which aim to gain robustness via redundancy alone. Finally, it has been proved that sets of PGHs can provide a complete and statistically optimal representation of a 2D shape, which is competent for design of a view-based 3D object learning and recognising system [14].

Learning and recognition of shape is provided by matching line segments on the basis of histogram similarity. The Bhattacharyya matching score (match score) [17] is utilised as the criteria for evaluating the similarity between two PGHs, and it is calculated as follows:

$$D_{Bhattacharyya} = \sum_{i=0}^{m} \sqrt{a_i}\sqrt{b_i} \qquad (1)$$

where m is the number of histogram bins and the maximum match score is supposed to be 1.0 indicating that the PGH a and b are identical. This match metric can be derived for independantly distributed Poisson data, such as fund in sample histograms and PGH's.

We propose a PGH based framework for a vision-based robotic system (shown in Figure 1) which is capable of learning and recognising scenes in an unknown man-made environment. In the system, reliable spatial information generated from corner features is used by the camera calibration algorithm, which will then iteratively optimise the calibration parameters. With an accurate camera calibration, stereo matching is able to provide a partial 3D reconstruction of a scene which can then be used to encode 2D reprojections from a range of viewpoints for entry into the PGH database. Scene recognition and robot localisation can be conducted by ranking the match scores between PGHs of a scene and the PGH database. In the end, visual feedback control offers a robust path planning capability for robot navigation.

In the real world implementation, such a robotic system requires good coordination between all the functional modules under practical circumstances. In order to assess the robot performance without depending physically on the machine, we first specify the computational characteristics of these modules, sufficient to allow a realistic simulation. The focus is to model the possible uncertainty sources that will disturb the robotic system, which is not considered in [1, 2, 3, 9, 10]. Figure 1 demonstrates several major uncertainty sources (in coloured boxes) that have dominant effects on our robotic system. All of them are vision-based uncertainties, which arise because the visual representation of the world is highly susceptible to environmental variations (*e.g.* illumination). In the system, (unlike [8]) each of the uncertainties propagates throughout the data streams, leading to an accumulated effect on subsequent modules.

Additionally, in our robotic system, calculation and matching of encoded edge features (i.e. PGHs) are the essence of the learning and recognition algorithm. It is because edges contain a large quantity of useful information defining the content and location of an object. Thus, we are most interested in uncertainties on geometric co-occurrences between edge features, rather than grey level values of specific image pixels. On this basis, generation of 2D images via a 3D rendering (as in [1, 2, 3, 4, 5, 6, 7, 9]) becomes unnecessarily time consuming.

Our intention is to construct a simulation program, using computational models which imitate the performance of each element, and to use this for the purpose of designing a 3D vision-based robot control system. In this pa-

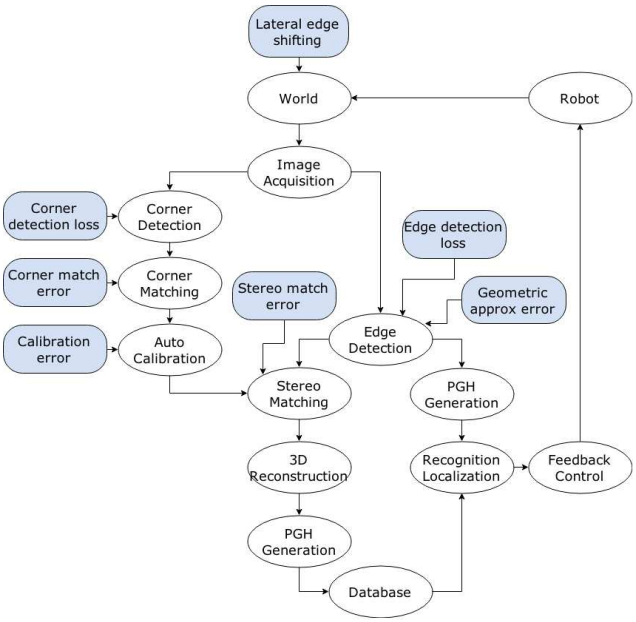


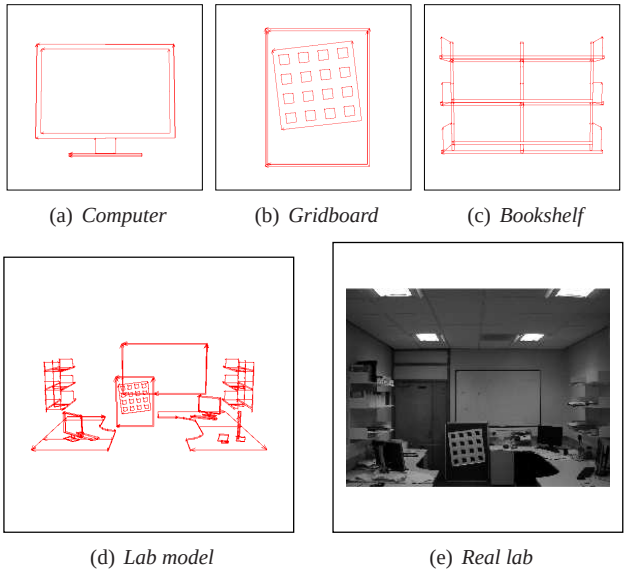
Figure 1. The flowchart of a vision-based robotic framework, where PGH is used as descriptor of extracted edge features. The coloured boxes indicate the practical uncertainties that interfere specific vision modules.

per, we present a simulation of a 3D vision-based robotic system with wire-frame representations and quantified uncertainty sources. Compared with the reviewed work, our simulation utilises wire-frame objects to represent a 3D virtual world, so that the computation workload in 3D rendering is avoided. Moreover, we simulate all the dominant uncertainty sources on our proposed robotic vision system, which provides a realistic environment for developing vision-based robotic algorithms.

Section 2 provides an overview of the simulation and details the modelling and simulating approaches of uncertainty sources. Finally, Section 3 provides the simulation result of uncertainty sources and a conclusion of the presented work.

2. Simulation Implementation

To create a 3D virtual world with wire-frame object representations and implement several environmental uncertainties as plug-in functions, we employ our own software environment named the TINA vision system [18]. The TINA vision system was firstly proposed in [19], in which a model based object recognition and localisation algorithm was demonstrated. This system was further developed during the past three decades, and the current version provides 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.



(a) *Computer* (b) *Gridboard* (c) *Bookshelf*

(d) *Lab model* (e) *Real lab*

Figure 2. *3D wire-frame models of several objects (a),(b),(c); 3D model of a potential working environment (d); an image of the real environment.*

The TINA vision system allows flexible construction of 3D wire-frame object models in a GUI environment. A virtual world can then be created by defining the positions and orientations of multiple object models under a Cartesian coordinate. As demonstrated in Figure 2 (a)(b)(c), we construct several object models based on a potential working environment. Each of the object models is a combination of multiple lines or ellipses. The profiles of these models are manually measured, aiming to provide an approximate appearance of the real world, as demonstrated in Figure 2 (d)(e).

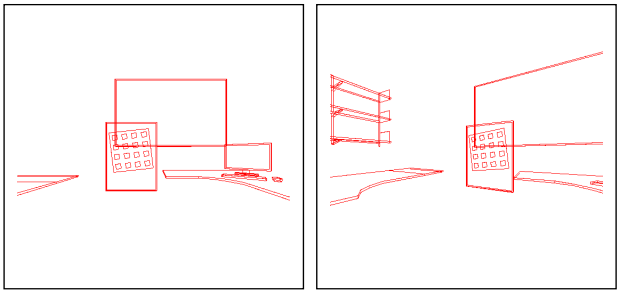
To simulated a stereo system on a virtual mobile robot, we firstly define a camera model using a set of intrinsic and extrinsic coefficients.

$$C = F(f, a_x, a_y, o_x, o_y, k, R, T) \tag{2}$$

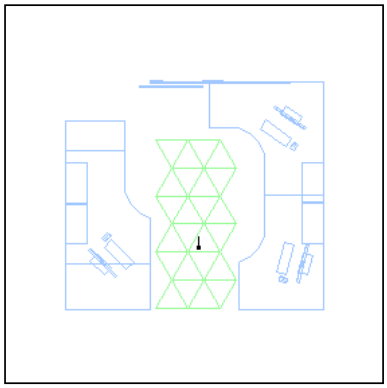
$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{bmatrix} \tag{3}$$

$$R = \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix} \tag{4}$$

Where f is focal length, (a_x, a_y) is aspect ratio, (o_x, o_y) is optical centre, k is radial distortion coefficient, R, T are the coordinate transformation matrices used to define the position and orientation of a camera under a pre-defined coordinate.



(a) *Image in left camera* (b) *Image in right camera*



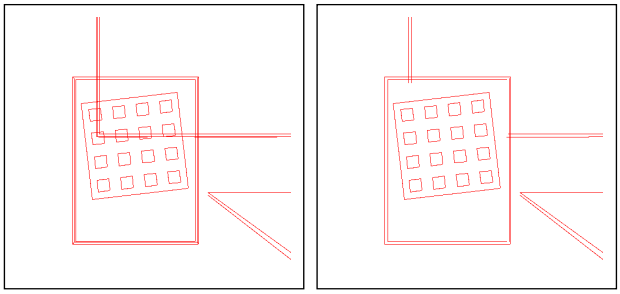
(c) *Top-view map of the world*

Figure 3. *2D projections of 3D world in a virtual stereo system: left camera (a), right camera (b); A top-view map indicating the virtual robot pose (c), where the black dot shows the robot position and the tail represents the robot orientation. The triangular-mesh in (c) is a pre-defined view-point manifold used for resolving view-dependency problem.*

The stereo system is then represented by two individual camera models, and both are moved according to the motion of a virtual robot. Real-time camera 2D projections of the virtual world are generated and displayed on the GUI, as shown in Figure 3. A top-view map is also provided in the GUI, indicating the robot pose.

At a view-point in the virtual world, only those features that are visible are displayed in the GUI, while all the self-occluding features are hidden. After this process, our virtual world is evidently more realistic, as demonstrated in Figure 4. This process is achieved with reference to a corresponding view-dependency file, which details which features are most likely to be visible from a finite number of pre-defined view-points in the virtual world. The nearest pre-defined view-point to the current robot position is taken to indicate which features should be visible. Moreover, the view-points follow a triangular-mesh manifold, which can be seen in Figure 3 (c).

In the simulation, several environmental uncertainties are also implemented as plug-in functions. The uncertainties include edge detection loss, lateral edge shifting,



(a) *Without view-dependency* (b) *With view-dependency*

Figure 4. *3D wire-frame model without view-dependency (a); 3D wire-frame model with view-dependency (b).*

geometric approximation error, corner match error, stereo match error and camera calibration error. Each of them has individual effect on all the visible features displayed at a view-point. The modelling and simulating approaches of these uncertainties are detailed as follows.

2.1. Edge Detection Loss and Lateral Edge Shifting

Contemporary edge detectors (*e.g.* Canny [20]) use grey level gradients for edge extraction. Since edges are usually represented by the discontinuities of brightness in an image, they are highly susceptible to illumination conditions. Due to several factors, such as blurring and lack of contrast with the background, genuine edges are not always detectable. More specifically, a part of an edge can either be totally undetectable or be detected with a biased orientation and location, and all these effects will result in an uncertainty to the subsequent vision algorithms.

In the real world, physical edges are usually intuitively defined as the discontinuities of smoothness of a surface. However, for humans, who are often used to define a ground truth, subjective definitions of physical edges vary [21]. The difference between edge definitions leads to a problem. This problem is, some of the manually defined edges can never be detected by an edge detector, due to their lack of contrast.

To simulate occluding boundaries on curved surfaces, dynamic line features are defined between ellipse features in our simulation. Their image projections are calculated and updated with reference to the current view-point. Such curves can acount for as much as 50% of the detectable edges in images of man made objects.

When comparing the actual detected location of edges with model predictions, some systematic problems become apparent. Under a distant illumination condition and for Lambertian reflectance models (excluding shadows), a planar surface with homogeneous texture is assumed to have the same grey-level. Illumination changes apply systematically along an extended boundary of the surfaces, i.e. lat-

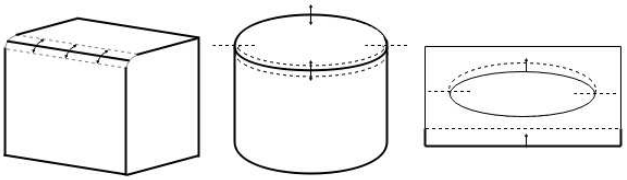


Figure 5. Examples of lateral edge shifting on various objects; the location of a straight line along a curved surface boundary (left), the elliptical edge on top of a cylinder (middle), the inner edge of a hole (right). All the elliptical edges are split along their major axis, with each half being shifted independently. [16]

eral edge shifting [16]. Figure 5 shows some examples of lateral edge shifting on various objects, it can be seen that some of the edges shift laterally within a certain range along an extended boundary. Notablely, lateral edge shifting is a physical phenomenon that happens in the real-world and affects all edge detection algorithms.

Given an image, edge detection loss is defined as the percentage of genuine edges that can not be properly detected by an edge detector. In our approach, we are only interested in analysing the performance of the Canny edge detector. To quantify the edge detection loss of the Canny edge detector when applying to a generic image, [22] proposes a quantitative verification on Canny edge detection using a power law model. This approach manually creates wire-frame models of a selection of referenced man-made objects constructed from a variety of materials. The approach assumes that a faithfully representative wire-frame model has been accurately projected onto the corresponding image data, and also the wire-frame model reliably represents any occluded edges across changes in view-point (*e.g.* occluded boundaries). The image data and model projections are shown in Figure 6. The concept of this verification is to statistically test the probability that a specified model point projects onto a verified image point. This probability can then be accumulated across a sampled model to support a model match verification. Additionally, a process of lateral edge shifting has been adopted in this approach to ensure that all the shifted edges are optimally aligned onto the model projection. The experimental verification result is demonstrated in Figure 7, in which the average edge detection rate is 71%.

Assuming the wire-frame models perfectly represent all the non-occluded genuine edges of corresponding objects in all cases, the experimental verification result indicates that Canny edge detector can extract typically 71% of the possible (human defined) edges for a selection of sampled objects under a specific illumination condition. Although we cannot say how many edges should have been detected (non-subjectively), this still provide a quantitative approximation of edge detection loss (29%) that Canny detector may have when applied to an image.

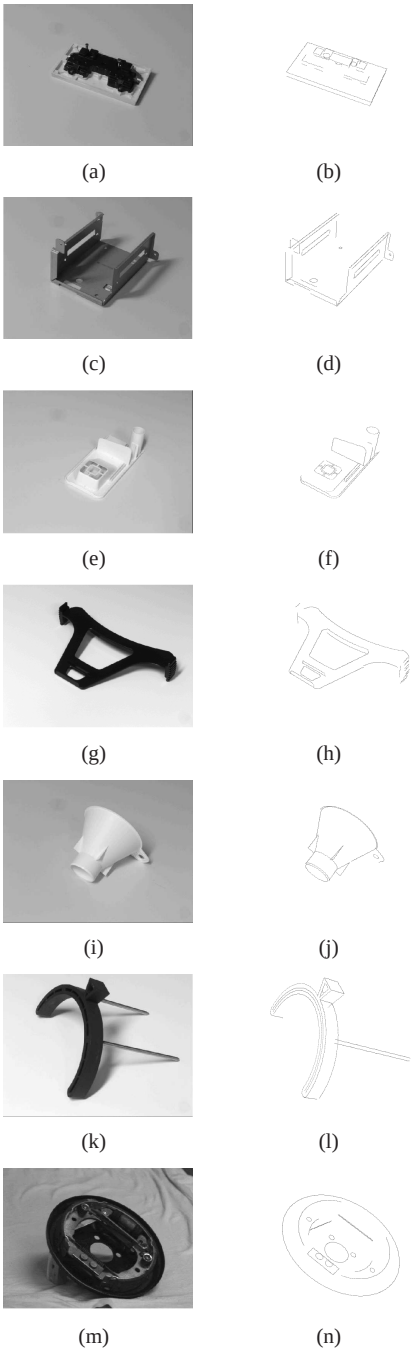


Figure 6. *Test objects and the features re-projected from the corresponding view-based 3D models [22]. Reading from top to bottom by referenced name: plug, tray, widget, aframe, funnel, stand, brake.*

In our simulation, edge detection loss is modelled as an uncertainty source that will remove 29% of the length of all the visible features. As a result, the remaining edge information is supposed to be at a similar level with the predicted Canny performance. Although the simulated features may

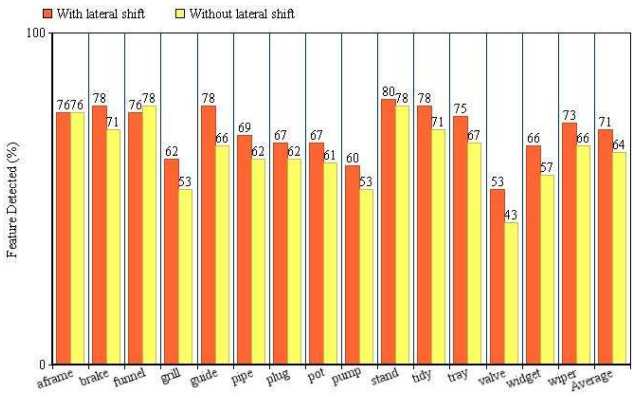


Figure 7. The percentages of verified image edges versus edges re-projected from the corresponding models.

be different to the real image evidence, they are intended to illicit a similar effect to edge-based object recognition algorithms. In addition, the uncertainty of lateral edge shifting moves every feature laterally within a pre-defined pixel range (*e.g.* 3 pixels). For this process, ellipse features are first separated along theri major axis, with each half being shifted independently.

2.2. Geometric Approximation Error

Geometric approximation is the process in which a geometric pattern (*e.g.* a line) is fitted to a set of detected image points (*e.g.* edge points). This approximation is a necessity for all the geometry-based computer vision algorithms, as it provides fitted known geometries which are suitable for high-level vision-based analysis. The error of geometric approximation is defined as the residual between a fitted geometric pattern and its corresponding image points. In our approach, a function *poly_prox()* [23] (inspired by [24]) is used as the geometric approximation algorithm which iteratively fits a set of lines to the detected edge points until the maximum approximation error on each of the fitted lines is less than a pre-defined threshold (e.g 0.15 pixels). Such an error leads a sub-pixel spatial perturbation to all the fitted lines. In order to simulate this error, we define circular uncertainty regions around both of the end points of a line feature. In the simulation, the positions of the end points are shifted according to a 2D *Gaussian* noise in the 2D image coordinate. The standard deviation of this 2D *Gaussian* noise is set to be consistent with the same with the geometry threshold. In this way, all the simulated line features contain the same level of geometric approximation error as the real line fits.

2.3. Corner Detection Loss and Corner Match Error

Corner detection loss is the proportion of corners which can not be detect again given that they have already been detected in one image. It is usually caused by the change on local contrast of an image due to view-point shifting. Corner match error is the amount of incorrect corner matches that are accepted in a corner matching process. It happens due to the match ambiguities between different match candidates.

We adopt a corner matching algorithm [25] which is based on *Harris* and *Stephens* corner detector [26]. On average the probability of finding a corner in image 2 given that it has been detected in image 1 is approximately 0.85 [25] (this varies for specific algorithm configurations and datasets). The algorithm is also demonstrated with an intensively textured image pair, where the number of incorrect matches is approximately 1% of all matches. The number of incorrect matches is correlated with a pre-defined searching region within which the algorithm searches for the best match candidate. The default region is a circular area with a radius of 10 pixels.

In our simulation, corner features are manually defined on the wire-frame models. We simulate corner detection loss by randomly deleting equal proportions of corners in the left and right virtual cameras respectively. Next, 1% of all the corners in both cameras are shifted randomly in a region consistent with the match search, so that the corner match error is simulated.

2.4. Stereo Match Error

Stereo matching is the process to match two image features which represent the same physical entity. This process is useful for extracting distance information from a stereo image pair, and thereby reconstructing an object model. In our approach, we adopt a multi-scale temporal correlation research algorithm [27] for stereo matching. Given a referenced feature in image 1, the algorithm obtains its best match candidate in image 2 by searching within a region along the corresponding epi-polar line. The searching region is defined as a disparity range on epi-polar line, and it is select as 20 pixels on each side. Image correlation measure is used to determine the best match candidate.

The ambiguity on correlation always result in mismatches between features. A stereo mis-match will likely generate an outlier in a reconstructed 3D model, and it happens more frequently when an image contains repeated patterns (*e.g.* a grid). In order to obtain the mis-match rate of the algorithm, [27] conducted a quantitative analysis. As reported, the typical mis-match rate is 1%.

In addition, features that are parallel with epi-polar lines can not be reliably matched using correlation-based stereo matching algorithms. It is because the correlation ambigui-

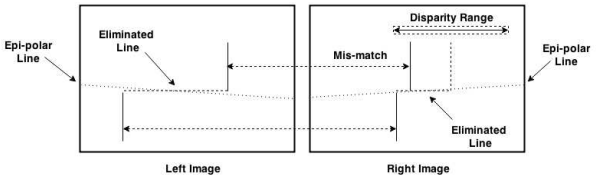


Figure 8. Stereo mis-match is simulated by firstly eliminating all the lines that are closely parallel with epi-polar lines, and then shifting a small amount of lines within a disparity range.

ties are significant when searching among these features. To be simplistic, this stereo matching algorithm inherently ignore all the features which are $\pm$ 5 degrees parallel to their intersected epi-polar lines.

In our simulation, stereo match error firstly eliminates all the features which are $\pm$ 5 degrees parallel to their intersected epi-polar lines. Afterwards, 1% of the remaining feature are shifted randomly within the disparity range (20 pixels on each side). This procedure is demonstrated in Figure 8.

2.5. Camera Calibration Error

Camera calibration estimates the intrinsic and extrinsic parameters of a camera model using information from image evidence. A citation-classic calibration approach (i.e. *Tsai* algorithm [28]) requires a known grid-board stimuli in a 3D coordinate and the corresponding 2D image pixels. In a stereo calibration process [29], the two camera models are intended to be firstly calibrated independently. These models are then optimised by iteratively minimising the error between model geometry and image evidence using matched image feature pairs (*e.g.* matched corner pairs). We use *Tsai* algorithm to obtain an initial calibration and then construct a χ_t^2 likelihood function which determines the residual between the image evidence and its theoretical estimation given a set of camera parameter.

$$\chi_t^2(A) = \sum \Big[y_j - y(x_j; A) \Big]^2 \tag{5}$$

Where y_j is the image evidence and $y(x_j; A)$ is the estimation with respect to input feature x_j and the parameter set A. The subsequent stereo calibration algorithm combines this function with two extra terms χ_e and χ_c, which represent the accumulated error on epi-polar lines and the error on previous estimated parameters,

$$\chi^2 = \chi_t^2 + \chi_e^2 + \chi_c^2 \tag{6}$$

This is done via a process of covariance combination [29].

Calibration error is described by a covariance matrix, defined as second partial derivatives of the χ^2 likelihood function with respect to parameters a_p and a_q $(p, q = 1, 2...m)$.

$$cov(a_p, a_q) = \begin{bmatrix} \frac{\partial^2 \chi^2}{\partial a_1 \partial a_1} & \frac{\partial^2 \chi^2}{\partial a_1 \partial a_2} & \cdots & \frac{\partial^2 \chi^2}{\partial a_1 \partial a_m} \\ \frac{\partial^2 \chi^2}{\partial a_2 \partial a_1} & \ddots & & \\ \vdots & & \ddots & \\ \frac{\partial^2 \chi^2}{\partial a_m \partial a_1} & & & \frac{\partial^2 \chi^2}{\partial a_m \partial a_m} \end{bmatrix} \tag{7}$$

We apply singular value decomposition (SVD) to the covariance matrix, in order to derive singular vectors V_i and singular values w_i $(i = 1, 2...m)$ of the covariance matrix, where the former indicates the error contributions of all parameters and the latter indicates the inverse magnitude of error distribution on the corresponding vector parameters.

$$cov(a_p, a_q) = \begin{bmatrix} V_1 & \cdots & V_m \end{bmatrix} \times \begin{bmatrix} \frac{1}{w_1} & & \\ & \ddots & \\ & & \frac{1}{w_m} \end{bmatrix} \times \begin{bmatrix} D \end{bmatrix} \tag{8}$$

Where V_i is a vector with the dimension of $1 \times m$ and D is an arbitrary matrix with the dimension of $m \times m$.

Given a white noise σ_i $(i = 1, 2, ...m)$, the theoretical error distribution on each of the camera parameters a_i can be calculated.

$$\begin{vmatrix} \varphi_1 & \cdots & \varphi_m \end{vmatrix} = \begin{vmatrix} V_1 & \cdots & V_m \end{vmatrix} \times \begin{vmatrix} \frac{\sigma_1}{\sqrt{w_1}} \\ \vdots \\ \frac{\sigma_m}{\sqrt{w_m}} \end{vmatrix} \tag{9}$$

Where φ_i $(i = 1, 2, ...m)$ is the estimated error on the camera parameter a_i.

To obtain a stereo camera model and a realistic covariance matrix from the stereo calibration, we set up a physical stereo system in the real world and calibrate it using *Tsai* and [29]. The calibrated stereo camera model is directly used in our simulation and its corresponding covariance matrix is utilised to generate calibration error.

In our simulation, σ_i is defined as a *Gaussian* random noise (*e.g.* with a standard deviation of 0.01). The calculated φ_i is generated to the corresponding camera parameter a_i in our pre-defined camera model. The calibration error leads to a direct effect on the spatial distribution of features, and thereby affects the subsequent recognition algorithms.

3. Simulation Result and Conclusion

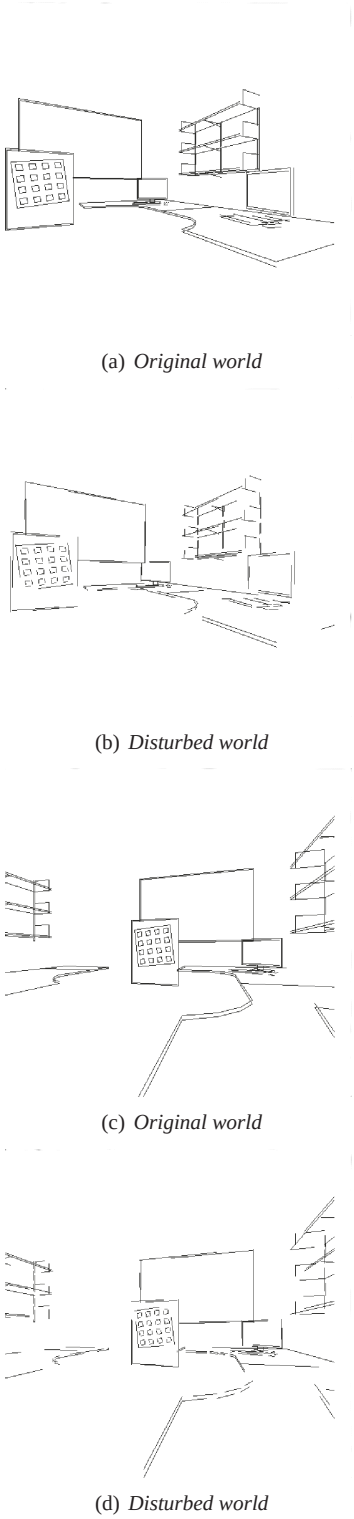
The uncertainty sources are designed as plug-in functions in our simulation. They have independent effects and can be applied consecutively. The order that the uncertainties take place should follow the data flow of the real system, so that no extra error which is caused by the interactive effect between uncertainties will be generated. Several

simulation results are demonstrated in Figure 9, in which an accumulative effect combining a selection of uncertainty sources is applied to the virtual world.

To conclude, this paper proposes a vision-based robotic simulation based on wire-frame object representations and quantified uncertainty sources. With this simulation, we aim to provide a realistic environment for assessing the performance of the proposed robotic system, which is designed able to learn and recognise an environment using feature-based vision algorithms.

The focus of the presented work is to state all the functional modules of the system, modelling their characteristics and simulating them in a 3D virtual environment. We use both statistical analysis on image evidences (*e.g.* for edge detection loss) and explicit mathematics derivation (*e.g.* for calibration error) to obtain and validate algorithm models. We believe the obtained models are good representations of the real-world perturbations. Moreover, the modelling process also provide the possible magnitudes of uncertainties. These magnitudes are likely to be biased, because they only follow a specific set of tested data, and they are also affected by subjective measurements. However, this is not an issue since we tend to adjust the uncertainty magnitudes to perform sensitivity analyses on crucial aspects of our robotic system (*e.g.* how accurate does the 3D data need to be before being useful for a specific task). In this way, we can specify the required characteristics of algorithms before starting to build the final system.

This simulation is only a starting point for the designing of a vision based robotic control system. Future work includes design and simulation of the presented vision-based robotic system, and finally implementation on a real-world machine. Evaluation of design choices is expected to require many thousands of simulations of recognition, planning and robot control scenarios, in order to obtain sufficient statistics to assess preformance. We believe if simulated appropriately, the final optimised system should simply work. If not, deficinecies in the evaluation process will be identified and the simulation modified in order to investigate solutions to specific failure modes, as outlined in the introduction.



(a) *Original world*

(b) *Disturbed world*

(c) *Original world*

(d) *Disturbed world*

Figure 9. *The effect of uncertainty sources (where lateral edge shifting, edge detection loss, geometric approx error, calibration error and stereo match error are applied)*

References

[1] Kei Okada, Yasuo Kino, Fumio Kanehiro, Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Rapid development system for humanoid vision-based behaviors with real-virtual common interface. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2515–2520. IEEE, 2002. 1, 2

[2] Ilkay Ulusoy, Ugur Halici, and Kemal Leblebicioglu. 3d cognitive map construction by active stereo vision in a virtual world. In *Computer and Information Sciences-ISCIS 2004*, pages 400–409. Springer, 2004. 1, 2

[3] Rafael Luiz Klaser, Fernando Santos Osório, and Denis Fernando Wolf. Simulation of an autonomous vehicle with a vision-based navigation system in unstructured terrains using octomap. In *Computing Systems Engineering (SBESC), 2013 III Brazilian Symposium on*, pages 177–178. IEEE, 2013. 1, 2

[4] Kazunori Asanuma, Kazunori Umeda, Ryuichi Ueda, and Tamio Arai. Development of a simulator of environment and measurement for autonomous mobile robots considering camera characteristics. In *RoboCup 2003: Robot Soccer World Cup VII*, pages 446–457. Springer, 2004. 1, 2

[5] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawai, and Hitoshi Matsubara. Robocup: A challenge problem for ai and robotics. In *RoboCup-97: Robot Soccer World Cup I*, pages 1–19. Springer, 1998. 1, 2

[6] Michal Kučiš and Pavel Zemčík. Simulation of camera features. In *Proceedings of the 16th Central European Seminar on Computer*, pages 117–123. Technical University Wien, 2012. 1, 2

[7] Jürgen Rossmann, Nico Hempe, Markus Emde, and Thomas Steil. A real-time optical sensor simulation framework for development and testing of industrial and mobile robot applications. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–6. VDE, 2012. 1, 2

[8] André Hinkenjann, Thorsten Roth, Jessica Millberg, Hojun Yun, and Yongmin Li. Real-time simulation of camera errors and their effect on some basic robotic vision algorithms. In *Computer and Robot Vision (CRV), 2013 International Conference on*, pages 218–225. IEEE, 2013. 1, 2

[9] Xiangjun Zou, Haixin Zou, and Jun Lu. Virtual manipulator-based binocular stereo vision positioning system and errors modelling. *Machine Vision and Applications*, 23(1):43–63, 2012. 2

[10] Nak Yong Ko, Dong Jin Seo, Gwang Jin Kim, Yongseon Moon, and Youngchul Bae. Simulation of mobile robot motion considering uncertainties in robot model. In *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on*, pages 389–394. IEEE, 2008. 2

[11] PA Riocreux, Neil A Thacker, and RB Yates. An analysis of pairwise geometric histograms for view-based object recognition. In *BMVC*, pages 1–10, 1994. 2

[12] Anthony Ashbrook, Neil A Thacker, Peter Rockett, and CI Brown. Robust recognition of scaled shapes using pairwise geometric histograms. In *BMVC*, volume 95, pages 503–512, 1995. 2

[13] Alun Evans, Neil A Thacker, and John EW Mayhew. The use of geometric histograms for model-based object recognition. In *BMVC*, volume 93, pages 429–438. Citeseer, 1993. 2

[14] Neil A Thacker, PA Riocreux, and RB Yates. Assessing the completeness properties of pairwise geometric histograms. *Image and Vision Computing*, 13(5):423–429, 1995. 2

[15] John PJ Pinel. *Biopsychology*. Pearson Education, 1997. 2

[16] Simon Coupe. *Machine Learning of Projected 3D Shape*. PhD thesis, University of Manchester, 2009. 2, 5

[17] Frank J Aherne, Neil A Thacker, and Peter I Rockett. The bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 34(4):363–368, 1998. 2

[18] Tina open source computer vision development environment. http://www.tina-vision.net. Accessed: 2015-05-21. 3

[19] J Porrill, SB Pollard, TP Pridmore, JB Bowen, JEW Mayhew, and JP Frisby. Tina: The sheffield aivru vision system. In *Proc. 9th IJCAI*, volume 2, pages 1138–1144, 1987. 3

[20] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 8(6):679–698, 1986. 4

[21] P. Mowforth and L. Gillespie. *Edge Detection as an Ill-posed Task*. The Turing Institute, 1987. 4

[22] Simon Coupe and Neil Thacker. Quantitative verification of projected views using a power law model of feature detection. In *Computer and Robot Vision, 2008. CRV'08. Canadian Conference on*, pages 352–358. IEEE, 2008. 5

[23] Tina open source computer vision development environment. http://www.tina-vision.net/lxr/tina6-all/http/search. Accessed: 2015-05-21. 6

[24] David G Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial intelligence*, 31(3):355–395, 1987. 6

[25] Neil A Thacker and Patrick Courtney. Statistical analysis of a stereo matching algorithm. In *BMVC92*, pages 316–326. Springer, 1992. 6

[26] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988. 6

[27] Simon Crossley. *Robust Temporal Stereo Computer Vision*. PhD thesis, University of Sheffield, 2000. 6

[28] Roger Y Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, 1987. 7

[29] Neil A Thacker and John EW Mayhew. Optimal combination of stereo camera calibration from arbitrary stereo images. *Image and vision computing*, 9(1):27–32, 1991. 7