Rigid Transformation Using Interval Analysis for Robot Motion Estimation

DOI:
10.1109/CSCS.2015.98

2015 20th International Conference on Control Systems and Science

Rigid Transformation using Interval Analysis for Robot Motion Estimation

M. Mustafa, A. Stancu, S. P. Gutierrez, E. A. Codres
School of EEE, Control Systems Research Group
The University of Manchester
M13-9PL, Manchester, United Kingdom
Emails: {mohamed.mustafa, alexandru.stancu, salvador.pachecogutierrez, eduard.codres}@manchester.ac.uk

L. Jaulin
ENSTA-Bretagne
Bureau E116, 2 rue Franois Verny
29806 Brest Cdex 09, France
Email: luc.jaulin@ensta-bretagne.fr

***Abstract*—Rigid transformation is a popular method to estimate the robot motion given two sets of corresponding points seen from two different locations. Range imaging sensors, such as stereo camera and structured-light 3-D scanner, can be used to provide these corresponding points, however, such sensors have measurement uncertainty defined by intervals with upper and lower bounds. This paper presents a new approach to use interval analysis and simultaneously estimate the following: (1) the robot motion and, (2) the position of the landmarks with respect to the initial frame of reference, i.e. the robot initial pose. We will show that using this approach, the uncertainties of the landmark positions decrease over time, which causes the uncertainty of the robot pose to remain bounded. Our approach is illustrated with examples using simulated data, and real data acquired by Kinect sensor.**

***Keywords*-Interval Analysis; Optimization; Parameter Estimation; Range Imaging Sensor; Mobile Robot Localization;**

I. INTRODUCTION

Estimating robot motion is essential for Simultaneous Localization and Mapping problem (SLAM), and it can be achieved by employing either proprioceptive sensors or exteroceptive sensors. The latter has the advantage where the uncertainty of the estimated parameters is bounded according to the uncertainty of sensor measurement. Rigid transformation is one of many approaches used to estimate the robot motion using exteroceptive sensors, such as laser scanners and cameras. The method of estimation depends on several factors including: the type of measurement uncertainty and the availability of features correspondence between different frames of reference. If the measurement uncertainty follows a Gaussian probability distribution and the features correspondence between frames is not available, Iterative Closest Point (ICP) algorithm can be used to find the translation and rotation between two different frames [1]. If there exists correspondence between different frames, least squares techniques with tools such as: Singular Value Decomposition (SVD) and unit quaternion, can be applied to estimate the rigid transformation by minimizing the distance between corresponding points [2], [10].

Range imaging sensors, such as: stereo camera and structured-light 3-D scanner, have measurement uncertainty defined by upper and lower bounds, with no further assumption about its probability distribution [3], [4], [7]. Though, it is possible to approximate such error with some probability distribution and use probabilistic tools [3], we propose to apply Interval Analysis [5] directly to the measurement intervals and compute solution regions for the rigid transformation problem. In [5] and [6], Jaulin, et al., used interval analysis to solve range-only SLAM using range sensors such as laser scanners and sonars. The approach proposed in this paper deals with measurements acquired by stereo cameras or structured-light 3-D scanners such as Microsoft Kinect, in which the landmark positions are measured with bounded uncertainty. Bethencourt and Jaulin [8] used interval analysis with data acquired by Kinect to reconstruct 3-D scenes. In our approach, we extend that idea to estimate the robot motion, and reduce the uncertainty of the landmark positions with respect to the robot initial pose, i.e., the initial frame of reference.

The paper is organized as follows: Section II presents the relationship between the rigid transformation and robot motion, and how it can be computed using exteroceptive sensors. This section also gives an introduction to range imaging sensors and the nature of error associated with their measurements. Interval analysis is well-suited for such type of error, and some of the useful tools in this field are introduced in Section III, such as: Set Inversion Via Interval Analysis (SIVIA), and contractors. The rigid transformation problem is defined in the framework of interval analysis in Section IV, and the approach is illustrated by several examples using simulated data and real data. Section V concludes the paper, and presents some comments about future work.

II. RIGID TRANSFORMATION AND ROBOT MOTION

Consider a robot traversing in a static world, and it has the ability to measure the position of distinctive landmarks

978-1-4799-1780-8/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCS.2015.98

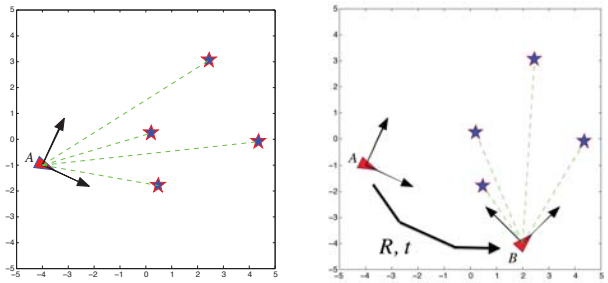
in the environment with respect to its location as shown in Fig. 1(a). As the robot moves in the world, it detects the landmarks at different positions relative to its new location as shown in Fig. 1(b). We assume the robot is able to solve the data association and correspondence problem, such that each landmark seen from the first location has a matching landmark seen from the second location [1], [9], [20] using algorithms such as: Iterative Closest Point, Single Compatibility, and Joint Compatibility Branch and Bound [1], [21]. Hence, the rigid transformation problem is defined as follows: given two sets of corresponding points in two different frames of reference $S_A = \{\mathbf{x}_1^a, \mathbf{x}_2^a, ..., \mathbf{x}_m^a\}$ and $S_B = \{\mathbf{x}_1^b, \mathbf{x}_2^b, ..., \mathbf{x}_m^b\}$, where $\mathbf{x}_i^a, \mathbf{x}_i^b \in \mathbb{R}^{D\times 1}, D \in \{1, 2, 3\}$, estimate the translation t and rotation R that satisfy the following equation:

$$\mathbf{x}_i^b = R^T\ (\mathbf{x}_i^a - t), \quad i = 1 : m, \tag{1}$$

where $t \in \mathbb{R}^{D\times 1}$, $R \in \mathbb{R}^{D\times D}$, and $\det(R) = 1$. This rigid transformation represents the robot motion from frame A to frame B. Equation (1) has a unique solution if the measurements in frame A and frame B are error-free.

Range imaging sensors such as: stereo cameras and structured-light sensors, use triangulation between two shifted images, as shown in Fig. 2, to determine the position of landmarks with respect to the sensor [3], [7], [16]. Due to image quantization and resolution limits, if the landmark lies anywhere in the shaded region, the sensor will return the same position value. This uncertainty region will increase as the landmark distance from the sensor increases. We will enclose this region by a box that is aligned with the coordinate system of the sensor as shown in Fig. 2; this will allow simple utilization of interval analysis as it will be explained in Section III. Now, consider a robot equipped with a range imaging sensor and moving in 2-D world as shown in Fig. 3. The landmarks are measured with uncertainties defined by regions (*boxes*) that are aligned with the robot frame of reference. As the robot moves from location A to location B, as shown in Fig. 3(a) and (b), new measurements with uncertainties are acquired, and the solution of (1) is no longer unique, but rather it satisfies a range of values for translation and rotation that can be computed using interval analysis.

Assume the robot moves again to a new location as shown in Fig. 3(c). Should it compute its motion with respect to location A or location B? It is important to always cast the motion to a fixed frame in the environment. Since location B is estimated with respect to location A, it will have some uncertainty due to the measurement error. An ideal choice for the fixed frame would be the initial frame A. In this case, the measurements from frame A needs to be as accurate as possible. This can be done by using the measurements from frame B and reduce the uncertainty of the measurements in frame A.



(a) (b)

Figure 1. Robot, represented by red triangle, moves in static 2-D world with 4 landmarks represented by blue stars. (a) Landmarks seen by the robot from location A, (b) landmarks seen from location B.

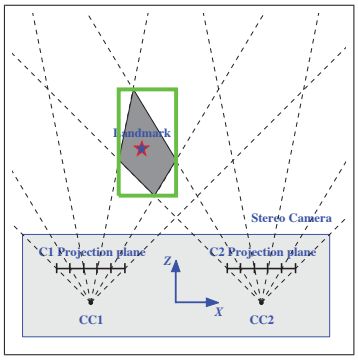


Figure 2. Range imaging geometry showing triangulation uncertainty. The blue box represents the range imaging sensor, and each segment in the projection plane represents a pixel. Any point in the shaded region projects on the same pixel in the right and left planes. This represents an uncertainty about the position of the landmark (blue star). The uncertainty region is enclosed by a green box aligned with the sensor coordinate system to facilitate the use of interval analysis.

III. INTERVAL ANALYSIS

A. Real Intervals Arithmetic

A real interval is a connected, closed subset of $\mathbb{R}$, and it is denoted by $[x] = [x^-, x^+]$ where x^- and x^+ are the lower and upper bounds of the interval, respectively. The set of all real intervals of $\mathbb{R}$ is denoted by $\mathbb{IR}$ [5].

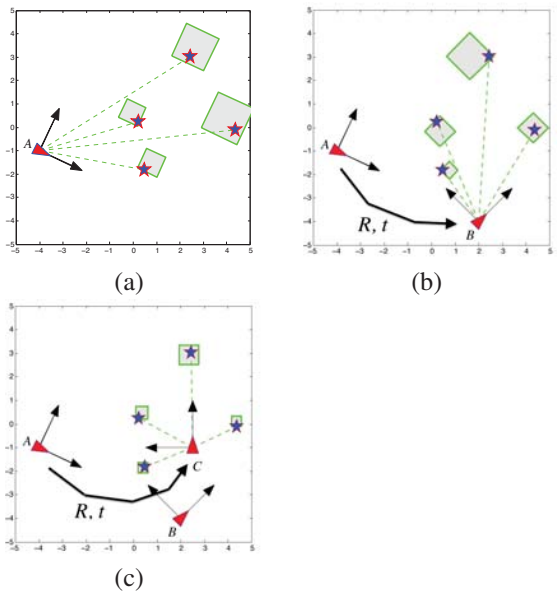
Basic arithmetic operations and functions can be applied to intervals. Let $\diamond$ be an arithmetic operation, i.e., $\diamond \in \{+, -, *, /\}$, and $[x]$, $[y]$ are real intervals, then we can define the following:

$$[x] \diamond [y] \triangleq [\{x \diamond y \in \mathbb{R} \mid x \in [x], y \in [y]\}]. \tag{2}$$

Similarly, elementary functions can be extended to intervals. Consider f as an elementary function from $\mathbb{R}$ to $\mathbb{R}$, where $f \in \{\text{sin, cos, log, exp, sqr, sqrt, ... }\}$, the interval function is defined as:

$$f([x]) \triangleq [\{f(x) \in \mathbb{R} \mid x \in [x]\}]. \tag{3}$$

For instance, $[-1, 4] + [2, 3] = [1, 7]$; $[-1, 4] \times [2, 3] = [-3, 12]$; $([-1, 4])^2 = [0, 16]$; $\sin([-1, 4]) = [-0.8415, 1]$; and $e^{([-1,4])} = [0.3679, 54.5982]$.



(a) (b) (c)

Figure 3. Robot, represented by red triangle, measures landmarks with uncertainties defined by the shaded regions (*boxes*). (a) Landmarks seen by the robot from location A, (b) landmarks seen from location B, (c) Landmarks seen from location C

To deal with intervals in higher dimensional problems, a vector of intervals, also called a *box*, $[\mathbf{x}]$ is introduced [5], [11]. A box is a subset of $\mathbb{R}^n$ that can be defined as the Cartesian product of n intervals, and it can be written as $[\mathbf{x}] = [x_1] \times [x_2] \times ... \times [x_n]$. The set of all n-dimensional boxes is denoted by $\mathbb{IR}^n$.

B. Contractors

Consider the system of m equations with n variables

$$f_j(x_1, x_2, ..., x_n) = 0, \quad j = 1 : m, \tag{4}$$

where each variable x_i belongs to an interval $[x_i]$, and $[\mathbf{x}] = [x_1] \times [x_2] \times ... \times [x_n]$. In vector form, (4) can be written as $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. This corresponds to a *constraint satisfaction problem* (CSP) $\mathcal{H}$, which can be formulated as:

$$\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0},\ \mathbf{x} \in [\mathbf{x}]). \tag{5}$$

The *solution set* of $\mathcal{H}$ is defined as:

$$\mathbb{S} = \{\mathbf{x} \in [\mathbf{x}] \mid \mathbf{f}(\mathbf{x}) = \mathbf{0}\}. \tag{6}$$

Contracting $\mathcal{H}$ means replacing $[x]$ by a smaller domain $[x']$ such that the solution set remains unchanged, i.e., $\mathbb{S} \subset [\mathbf{x}'] \subset [\mathbf{x}]$. Therefore, a *contractor* $\mathcal{C}$ for $\mathcal{H}$ is an operator that compute the subset $[\mathbf{x}']$. The formal definition of a contractor is as follows:

Definition 1. A contractor $\mathcal{C}$ is a mapping from $\mathbb{IR}^n$ to $\mathbb{IR}^n$ such that:

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \quad \mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}], \tag{7}$$

$$\mathcal{C}([\mathbf{x}]) \cap \mathbb{S} = [\mathbf{x}] \cap \mathbb{S}, \tag{8}$$

where $\mathbb{S}$ is the solution set of $\mathcal{H}$. Equations (7) and (8) represent the contractance and correctness properties of the contractor, respectively [5], [11].

Contractors can be built using interval arithmetic and elementary functions [5], [12]. For instance, consider $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0},\ \mathbf{x} \in [\mathbf{x}])$, where $[\mathbf{x}] = [-1, 2] \times [-1, 2]$ and $\mathbf{f}[\mathbf{x}] = x_2 - \log(x_1)$. Now, define a contractor $\mathcal{C} : \mathbb{IR}^2 \to \mathbb{IR}^2$ for $\mathcal{H}$ and apply it to the interval $[\mathbf{x}]$ as follows:

$$\begin{aligned} \mathcal{C}([\mathbf{x}]) &:= (\mathcal{C}_{x_1}, \mathcal{C}_{x_2})([\mathbf{x}]) \\ &:= ([x_1] \cap e^{[x_2]}, [x_2] \cap \log[x_1]) \\ &= [0.3679, 2] \times [-1, 0.6931]. \end{aligned} \tag{9}$$

Fig. 4 illustrates result of applying contractor $\mathcal{C}$ to $[\mathbf{x}]$.

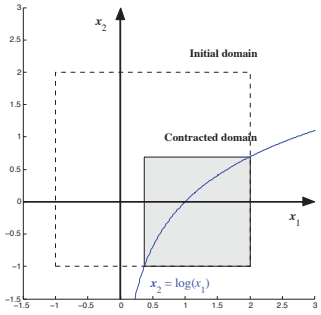


Figure 4. Contractor $\mathcal{C}$ applied to $[\mathbf{x}] = [-1, 2] \times [-1, 2]$ and results in $[\mathbf{x}'] = [0.3679, 2] \times [-1, 0.6931]$.

There are different ways to build contractors such as: Gauss elimination that works well with linear constraints, and Newton with preconditioning which can handle non-linear constraints. However, one of the most efficient is *forward-backward propagation* contractor [5]. In fact, the example shown in Fig. 4 uses this type of contractors. To illustrate that, define $y = \mathbf{f}([\mathbf{x}])$. The forward propagation uses the constraints $\mathbf{f}(\mathbf{x})$ to compute y as follows:

$$\begin{aligned} [a_1] &:= \log[x_1]; \\ [y] &:= [x_2] - [a_1]; \end{aligned} \tag{10}$$

where a_1 is a temporary variables. Since $\mathbf{f}(\mathbf{x}) = \{0\}$, then, the domain of y should be taken equal to the singleton $\{0\}$ as follows:

$$[y] := [a_2] \cap \{0\} \tag{11}$$

The backward propagation is defined as:

$$\begin{aligned} [x_2] &:= [x_2] \cap ([y] + [a_1]) \\ [a_1] &:= [a_1] \cap ([x_2] - [y]) \\ [x_1] &:= [x_1] \cap e^{[a_1]} \end{aligned} \tag{12}$$

This procedure produces the same result as (9). Because the constraint here is a monotonic function, this contractor is applied once to produce the smallest box containing the solution set. However, for other CSPs, forward-backward

propagation contractors may be applied several times to result in the smallest possible box [5].

C. Set Inversion Via Interval Analysis (SIVIA)

Consider the system of equations $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, the *set inversion* [5] is the problem of characterizing the set $\mathbb{S}_\mathbf{x}$ of all vectors $\mathbf{x}$ that satisfy the system equations and belong to a large search box $[\mathbf{x}]$. The recursive algorithm shown in Table I computes a set of non-overlapping boxes, also called *subpaving*, $\mathcal{L}$, that contains $\mathbb{S}_\mathbf{x}$. $\mathcal{C}_{\mathbb{S}_\mathbf{x}}$ used in Step 01 is a contractor for $\mathbb{S}_\mathbf{x}$ such that $\mathcal{C}_{\mathbb{S}_\mathbf{x}}([\mathbf{x}]) \cap \mathbb{S}_\mathbf{x} = [\mathbf{x}] \cap \mathbb{S}_\mathbf{x}$. $w([\mathbf{x}])$ represents the *width* of the box $[\mathbf{x}]$, and it is defined as follows:

$$w([\mathbf{x}]) \triangleq \max_{i=1:n} w([x_i]) = \max_{i=1:n} (x_i^+ - x_i^-). \qquad (13)$$

Table I
ALGORITHM SIVIAX FOR SOLVING A SYSTEM OF EQUATIONS [5].

Algorithm SIVIAX(in: $[\mathbf{x}]$, $\mathcal{C}_{\mathbb{S}_\mathbf{x}}$, ϵ; inout: $\mathcal{L}$)	
01	$[\mathbf{x}] := \mathcal{C}_{\mathbb{S}_\mathbf{x}}([\mathbf{x}])$;
02	if($[\mathbf{x}] = \phi$), return;
03	if($w([\mathbf{x}]) < \epsilon$),
04	$\mathcal{L} := \mathcal{L} \cup [\mathbf{x}]$; return;
05	bisect $[\mathbf{x}]$ into $[\mathbf{x}_1]$ and $[\mathbf{x}_2]$
06	SIVIAX($[\mathbf{x}_1]$, $\mathcal{C}_{\mathbb{S}_\mathbf{x}}$, ϵ, $\mathcal{L}$); SIVIAX($[\mathbf{x}_2]$, $\mathcal{C}_{\mathbb{S}_\mathbf{x}}$, ϵ, $\mathcal{L}$).

Set inversion via interval analysis, SIVIA, can be applied to CSPs, for example, consider the problem of characterizing the set of vectors $[\mathbf{x}]$ that satisfy the following constraints:

$$\begin{aligned} x_2 - x_1^2 &= y; \\ \sqrt{x_1} - x_2 &= y; \end{aligned} \qquad (14)$$

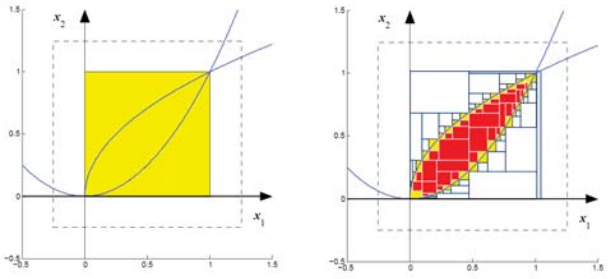
where $y \in [0, \infty)$. Fig. 5 shows the result of applying SIVIA with contractors to characterize the region enclosed by (14), where the initial search box is $[\mathbf{x}] = [-0.25, 1.25] \times [-0.25, 1.25]$. More details about SIVIA and contractors can be found in [5].

IV. RIGID TRANSFORMATION AS CONSTRAINT SATISFACTION PROBLEM

The rigid transformation problem can be seen in the framework of interval analysis as a constraint satisfaction problem (CSP) as follows: given two sets of corresponding points in two different frames of reference $S_A = \{\mathbf{x}_1^a, \mathbf{x}_2^a, ..., \mathbf{x}_m^a\}$ and $S_B = \{\mathbf{x}_1^b, \mathbf{x}_2^b, ..., \mathbf{x}_m^b\}$, where $\mathbf{x}_i^a \in [\mathbf{x}_i^a]$ and $\mathbf{x}_i^b \in [\mathbf{x}_i^b]$, estimate the solution set of translation $[\mathbf{t}]$, rotation $[R]$, and the landmark positions with respect to the initial robot position $[\mathbf{x}_i^a]$, that satisfy the following equation:

$$[\mathbf{x}_i^b] = [R]^T \ ([\mathbf{x}_i^a] - [\mathbf{t}]), \quad i = 1 : m, \qquad (15)$$

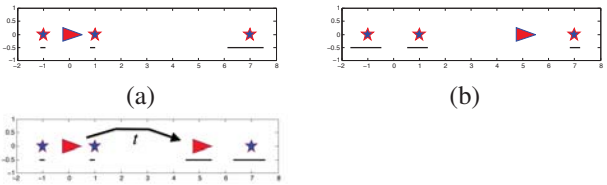
This CSP can be solved using SIVIA with contractors to find the solution set that are consistent with (15). In this paper, we assume that all landmarks in the environment are seen from all different robot locations. Also, the correspondence problem is assumed solved, i.e., the matching between



(a) (b)

Figure 5. The goal is to find the region enclosed by two inequalities $x_2 \geq x_1^2$ and $x_2 \leq \sqrt{x_1}$, where the initial box is $[-0.25, 1.25] \times [-0.25, 1.25]$. (a) Applying the contractor only results in the shaded box. (b) SIVIA with contractors produces three regions: red, white, and yellow. The red region represents all boxes that are part of the solution, e.g., boxes that are definitely enclosed by the two inequalities. The white region represents all box that are not part of the solution. The yellow region represents the set of boxes that have some part inside the solution region and some part outside the solution region, and the size of each box is determined by ϵ. This solution was generated using IBEX 2.1.7 [22].

different sets of landmarks seen from different locations is assumed available and accurate [16], [20]. Although different contractors can be applied [5], all implementations in this paper use forward-backward propagation contractors with SIVIA.



(a) (b)

(c)

Figure 6. Red triangular robot moves in 1-D world with 3 landmarks, in blue stars. Black horizontal lines represent the uncertainty in position of each item. (a) Robot at first location with measurements, (b) robot at the second location with measurements, (c) robot and landmarks estimated positions after applying SIVIA with $\epsilon = 0.3$.

Consider a robot moving in 1-D world as shown in Fig. 6. In this case, $R = 1$ and $\mathbf{t} \in [t_x], \mathbf{x}_i^a \in [x_i^a]$, and $\mathbf{x}_i^b \in [x_i^b]$. We build the contractor shown in Table II and use it with SIVIA algorithm described in Section III to compute the solution sets for t_x and x_i^a. The results are in Fig. 6(c) where the robot pose is estimated using the measurements shown in Fig. 6(a) and (b).

Fig. 6(c) shows that the uncertainty of the landmark at position 7 is reduced compared to the uncertainty in Fig. 6(a). In fact, as the robot keeps moving in the world, the uncertainty of all landmarks decreases over time as illustrated in Fig. 7, and this result is due to the geometric constraints of the landmarks. Therefore, the estimated landmark positions with respect to the initial robot pose become

Table II
FORWARD-BACKWARD PROPAGATION CONTRACTOR FOR ROBOT MOVING IN 1-D WORLD

	Algorithm *Contractor1D* (in: $[x_i^b]$; inout: $[t_x], [x_i^a], i = 1:n$)	
01	do	
02	for $i := 1$ to n,	
03	$[z_1] = [x_i^a] - [t_x]$;	// forward
04	$[z_1] = [z_1] \cap [x_i^b]$;	// projection
05	$[x_i^a] = [x_i^a] \cap ([z_1] + [t_x])$;	// backward
06	$[t_x] = [x^t] \cap ([x_i^a] - [z_1])$;	// backward
07	endfor	
08	while contraction is significant.	

more accurate, and the robot motion estimation does not increase over time.

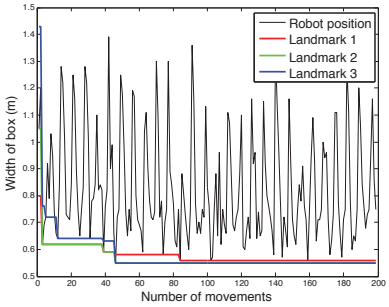


Figure 7. Robot moving in 1-D world: The uncertainty of the robot pose does not increase over time, however, the landmark uncertainties decrease as the robot moves in the environment.

Consider now the case of a robot moving in 2-D world and its motion is defined by translation only without rotation, hence $R = I_2$ and $t = [t_x\ t_y]^T$, where I_2 is the identity matrix. The robot can measure landmarks in the environment with uncertainties using the sensor described in Fig. 2. We extend the contractor shown in Table II to handle the example in 2-D world as illustrated in Table III.

Table III
FORWARD-BACKWARD CONTRACTOR FOR ROBOT MOVING, WITHOUT ROTATION, IN 2-D WORLD

	Algorithm *Contractor2D_basic* (in: $[x_i^b], [y_i^b]$; inout: $[t_x], [t_y], [x_i^a], [y_i^a], i = 1:n$)	
01	do	
02	for $i := 1$ to n,	
03	$[z_1] = [x_i^a] - [t_x]$;	// forward
04	$[z_2] = [y_i^a] - [t_y]$;	// forward
05	$[z_1] = [z_1] \cap [x_i^b]$;	// projection
06	$[z_2] = [z_2] \cap [y_i^b]$;	// projection
07	$[t_x] = [t_x] \cap ([x_i^a] - [z_1])$;	// backward
08	$[t_y] = [t_y] \cap ([y_i^a] - [z_2])$;	// backward
09	$[x_i^a] = [x_i^a] \cap ([z_1] + [t_x])$;	// backward
10	$[y_i^a] = [y_i^a] \cap ([z_2] + [t_y])$;	// backward
11	endfor	
12	while contraction is significant.	

Fig. 8(a) show the robot measuring the landmark positions from the first location, and Fig. 8(b) shows the robot after translation only measuring all the landmark positions. SIVIA algorithm and $contractor2D\_basic$ estimate the subpaving $\mathcal{L}$ for the robot location and the landmark positions. Fig. 8(c) shows the subpaving that represents the robot pose with small green boxes around the robot. The subpaving that represents the landmark positions is enclosed with an outer approximation, e.g. a single shaded box, that includes all solutions. This enclosure is done because the estimated landmark positions are needed for the next estimation step. The uncertainty of some landmarks, e.g., at position $[3.9,\ 1.4]^T$, is reduced after motion. As the robot moves in the environment and repeats the estimation process, the uncertainty of the landmarks decreases over time as illustrated in Fig. 9.

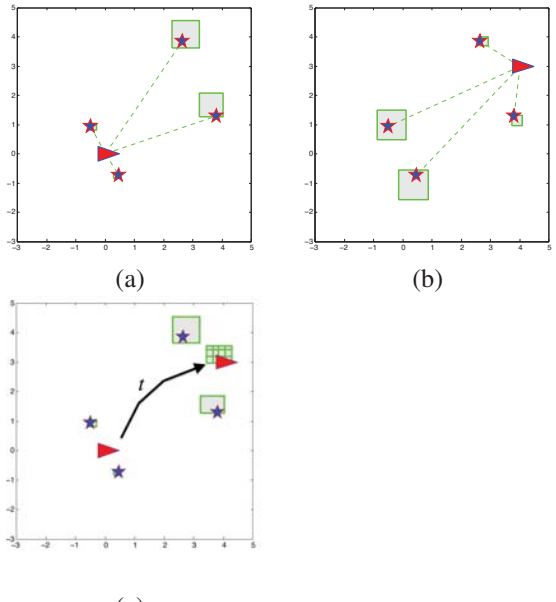


Figure 8. Red triangular robot moves in 2-D world with 4 landmarks, in blue stars. The shaded regions represent the uncertainty in position of each item. (a) Robot at first location with measurements, (b) robot at the second location with measurements, (c) robot and landmarks estimated positions after applying SIVIA with $\epsilon = 0.3$.

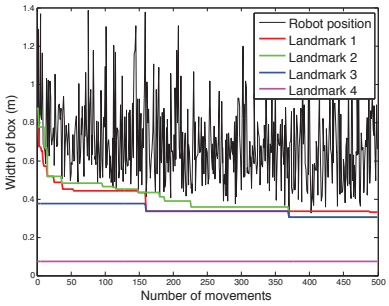


Figure 9. Robot moving in 2-D world (motion is translation only): The uncertainty of the robot pose does not increase over time, however, the landmark uncertainties decrease as the robot moves in the environment.

The third example to consider is a robot moving in 2-D

world, and its motion is defined by translation t and rotation $R(\theta)$. The robot pose parameters are t_x, t_y, θ, and the rigid transformation between points in two different frames can be written as:

$$x_i^b = \cos(\theta)\ (x_i^a - t_x) + \sin(\theta)\ (y_i^a - t_y)$$
$$y_i^b = -\sin(\theta)\ (x_i^a - t_x) + \cos(\theta)\ (y_i^a - t_y) \qquad (16)$$

We build the contractor shown in Table IV and use it with SIVIA algorithm to estimate the following parameters: $[t_x], [t_y], [\theta], [x_i^a]$, and $[y_i^a]$ where $i = 1 : n$. Fig. 10(a) and (b) show the robot and landmarks in the first and second frame, respectively, and Fig. 10(c) illustrates the subpaving estimated for the robot position and the outer approximation for the landmark positions after applying SIVIA algorithm. Fig. 11 shows that the uncertainties of landmark positions decrease as the robot keeps moving in the environment.

Table IV
FORWARD-BACKWARD CONTRACTOR FOR ROBOT MOVING, WITH TRANSLATION AND ROTATION, IN 2-D WORLD

Algorithm *Contractor2D_full* (in: $[x_i^b], [y_i^b]$; inout: $[t_x], [t_y], [\theta], [x_i^a], [y_i^a], i = 1 : n$)

```
do
  for i := 1 to n,
    [z1] = cos([θ]);                          // forward 1
    [z2] = sin([θ]);                          // forward 2
    [z3] = [x_i^a] − [t_x];                   // forward 3
    [z4] = [y_i^a] − [t_y];                   // forward 4
    [z5] = [z1] ∗ [z3];                       // forward 5
    [z6] = [z2] ∗ [z4];                       // forward 6
    [z7] = [z2] ∗ [z3];                       // forward 7
    [z8] = [z1] ∗ [z4];                       // forward 8
    [z9] = [z5] + [z6];                       // forward 9
    [z10] = [z8] − [z7];                      // forward 10
    [z9] = [z9] ∩ [x_i^b];                    // projection
    [z10] = [z10] ∩ [y_i^b];                  // projection
    [z7] = [z7] ∩ ([z8] − [z10]);             // backward 10
    [z8] = [z8] ∩ ([z10] + [z7]);             // backward 10
    [z5] = [z5] ∩ ([z9] − [z6]);              // backward 9
    [z6] = [z6] ∩ ([z9] − [z5]);              // backward 9
    [z1] = [z1] ∩ ([z8]/[z4]);                // backward 8
    [z4] = [z4] ∩ ([z8]/[z1]);                // backward 8
    [z2] = [z2] ∩ ([z7]/[z3]);                // backward 7
    [z3] = [z3] ∩ ([z7]/[z2]);                // backward 7
    [z2] = [z2] ∩ ([z6]/[z4]);                // backward 6
    [z4] = [z4] ∩ ([z6]/[z2]);                // backward 6
    [z1] = [z1] ∩ ([z5]/[z3]);                // backward 5
    [z3] = [z3] ∩ ([z5]/[z1]);                // backward 5
    [y_i^a] = [y_i^a] ∩ ([z4] + [t_y]);       // backward 4
    [t_y] = [t_y] ∩ (y_i^a − [z4]);           // backward 4
    [x_i^a] = [x_i^a] ∩ ([z3] + [t_x]);       // backward 3
    [t_x] = [t_x] ∩ (x_i^a − [z3]);           // backward 3
    [θ] = [θ] ∩ (arcsin([z2]));               // backward 2
    [θ] = [θ] ∩ (arccos([z1]));               // backward 1
  endfor
while contraction is significant.
```

The algorithm shown in Table IV is applied to a simulated mobile robot moving in 2-D environment shown in Fig. 12(a) where the robot is equipped with a simulated range imaging sensor, e.g., Microsoft Kinect, to observe distinctive landmarks. Fig. 12(b) shows the true robot path

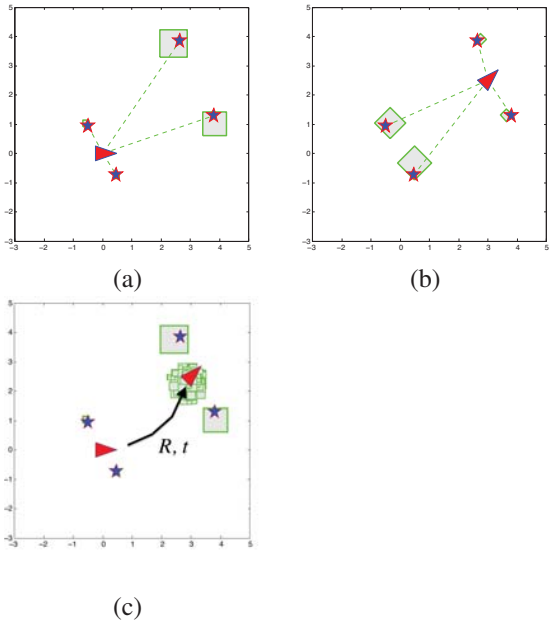


Figure 10. Red triangular robot moves in 2-D world with 4 landmarks, in blue stars. The shaded regions represent the uncertainty in position of each item. (a) robot at first location with measurements, (b) robot at the second location with measurements, (c) robot and landmarks estimated positions after applying SIVIA with $\epsilon = 0.3$.

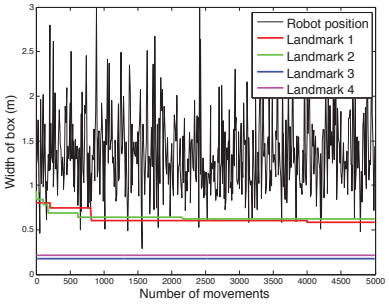
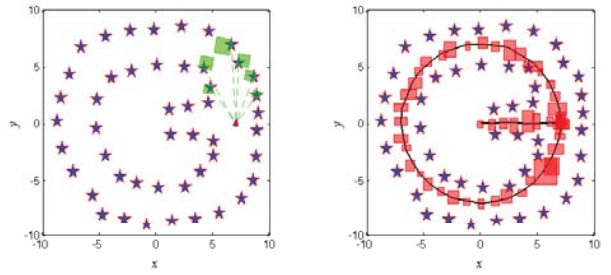


Figure 11. Robot in 2-D world (motion is translation and rotation): The uncertainty of the robot pose does not increase over time, however, the landmark uncertainties decrease as the robot moves in the environment.

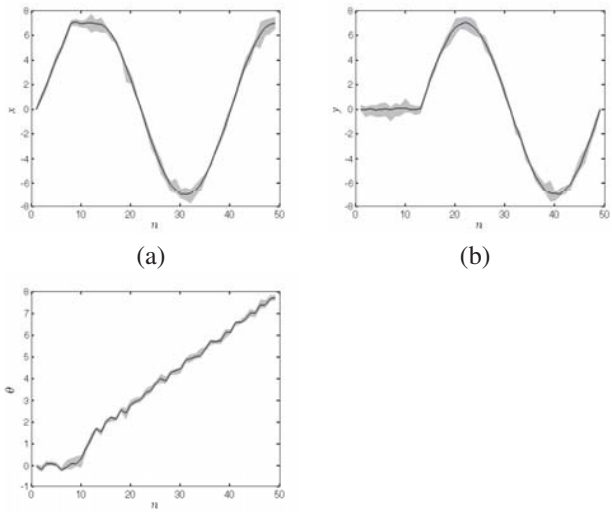
represented by the solid line, and the estimated xy poses (robot path) represented by the red boxes, where each box represents the *hull* of the resulting subpaving. As shown in Fig. 13, the uncertainties of the robot pose parameters $([x], [y], [\theta])$ do not increase over time, however, they do depend on the sensor noise and the robot true pose.

Finally, we test our SIVIA localization approach with real data acquired by Microsoft Kinect range imaging sensor mounted on a mobile robot as shown in Fig. 14. When Kinect captures a scene, it records two objects: (1) the RGB image of the scene, and (2) 3-D point cloud where each 3-D point corresponds to a pixel in the RGB image. When two scenes are available, corresponding 3-D points across scenes can be found using one of many features match-



(a) (b)

Figure 12. Mobile Robot Localization using SIVIA: (a) Robot (red triangle) observing (green boxes) landmarks (blue stars) using range imaging sensor. (b) The robot true path is represented by the black solid line, the estimated poses (x and y positions only) are represented by the red boxes. The pose estimation always includes the true robot pose.



(a) (b)

(c)

Figure 13. Robot localization using SIVIA (shaded region) compared to ground truth (solid line) over time. (a) x position, (b) y position, and (c) robot orientation θ.

ing algorithm [16] whose procedures can be summarized as follows: (1) from each RGB image, extract distinctive descriptors using feature detectors such as SIFT [18] and SURF [19], (2) match descriptors across scenes and find corresponding pixels, (3) use these pixels to find the set of corresponding 3-D points from the point clouds. Fig. 15 shows the result of the first two steps of the matching algorithm. We use these sets of corresponding points over several time frames to estimate the robot path using SIVIA and *contractor2D_full*. The results illustrated in Fig. 16 show that the true pose is always included in the estimated region. In this exercise, the robot average velocity is 0.2 m/sec, and by applying SIVIA combined with contractors for large number of observed landmarks, i.e., large number of constraints expressed by (16), the time complexity is no longer NP-hard, and solving for *local consistency* [6] is sufficient for real-time applications [5].

Figure 14. Microsoft Kinect mounted on Pioneer P3-DX mobile robot.

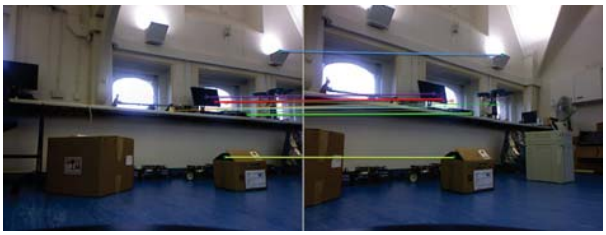


Figure 15. Microsoft Kinect is used to acquire two RGB images and the corresponding point clouds from two different scenes. Then, feature detector algorithms such as SIFT [18] or SURF [19], are used to find pixels with distinctive descriptors. These descriptors are matched across scenes, and the corresponding 3-D points from different point clouds are found [16]. Matched features are connected by yellow lines.

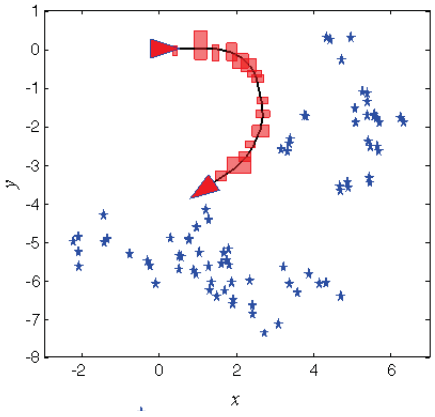


Figure 16. Mobile Robot Localization using SIVIA with data from Microsoft Kinect. The robot true path is represented by the black solid line, the estimated poses (x and y positions only) are represented by the red boxes, and the blue stars represent the observed landmarks. The pose estimation always includes the true robot pose.

V. CONCLUSIONS

This paper presented a new approach to compute the rigid transformation using range imaging sensors such as stereo cameras and structured-light 3-D scanner, with the help of interval analysis. This approach estimates the robot

motion, as well as the landmark positions, with respect to the initial robot pose, i.e., the initial frame of reference. Interval analysis tools such as: set inversion via interval analysis (SIVIA) and contractors, were used to compute the solution set. It is illustrated in the paper that this approach can be used to solve the robot localization problem where the robot pose uncertainty does not increase over time, and the uncertainty of landmark positions decreases as the robot moves in the environment. The approach can also be extended to solve the SLAM problem [17] by considering the following topics in the framework of interval analysis: (a) incorporate the data acquired by internal sensors such as encoders and IMU [8], (b) new landmarks are measured as the robot moves in the environment, and they need to be projected to the initial frame of reference in order to build a single large map, (c) matching can be available to only a subset of the landmarks in the environment, and (d) how to handle loop-closure and reduce the uncertainty of all landmark positions in the map. All of these issues will be considered as part of the future work.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the funding bodies and industrial partners who have supported the work presented in this paper including: the Innovate UK, Sellafield Ltd, the National Nuclear Laboratory (NNL), Dalton Nuclear Institute (DNI), and the UK Nuclear Decommissioning Authority (UKNDA).

REFERENCES

[1] P. J. Besl, and N. D. McKay, "A Method for Registration of 3-D Shapes," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 14, pp. 239-256, Feb. 1992.

[2] D. W. Eggert, A. Lorusso, and R.B. Fisher, "Estimating 3-D rigid body transformations: a comparison of four major algorithms," Machine Vision and Applications, vol. 9, pp. 272-290, Mar. 1997.

[3] L. Matthies, and S. A. Shafer, "Error Modeling in Stereo Navigation," IEEE Journal of Robotics and Automation, vol. RA-3, No. 3, pp. 239-250, Jun. 1987.

[4] K. Khoshelham, and E. Q. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," Sensors, vol. 12, pp. 1437-1454, 2013.

[5] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*, Sprinter, 2001, ISBN-13: 978-1852332198.

[6] L. Jaulin, "A Nonlinear Set Membership Approach for the Localization and Map Building of Underwater Robots," IEEE Transactions on Robotics, vol. 25, no. 1, pp. 88-98, Feb. 2009.

[7] D. Murray and J. J. Little, "Using Real-Time Stereo Vision for Mobile Robot Navigation," Autonomous Robots, vol. 8, pp. 161-171, Apr. 2000.

[8] A. Bethencourt, and L. Jaulin, "3D Reconstruction Using Interval Methods on The Kinect Device Coupled With an IMU," International Journal of Advanced Robotic Systems, vol. 10, 2013.

[9] A. S. Ogale, and Y. Aloimonos, "Shape and the Stereo Correspondence Problem, " International Journal of Computer Vision, vol. 65, pp. 147-162, Dec. 2005.

[10] B. Horn, "Closed-form solution of absolute orientation using unit quaternions," Journal of the Optical Society of America, vol. 4, pp. 629, Apr. 1987.

[11] O.Reynet, O. Voisin, and L. Jaulin, "Anchor-Based Localization Using Distributed Interval Contractors," 2013.

[12] L. Jaulin and E. Walter, Set inversion via interval analysis for nonlinear bounded-error estimation, Automatica, vol. 29, no. 4, pp. 10531064, 1993.

[13] M. Kieffer, L. Jaulin, and E. Walter, "Guaranteed Recursive Nonlinear State Estimation Using Interval Analysis," Proceedings of the 37th IEEE Conference on Decision & Control, vol. 4, pp. 3966 - 3971, Dec. 1998.

[14] F. Le Bars, J. Sliwka, O. Reynet, and L. Jaulin, "Set-membership state estimation with fleeting data," Automatica, vol. 48, pp. 381387, Feb. 2012.

[15] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, Cambridge University Press, 2009, ISBN-13: 978-0898716696.

[16] O. Enqvist, "Correspondence Problem in Geometric Vision," Ph.D. dissertation, Centre for Mathematical Sciences, Lund University, Sweden, 2009.

[17] H. D. Whyte, and T. Bailey, "Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms," Robotics & Automation Magazine, IEEE, vol. 13, pp. 99-110, Jun. 2006.

[18] D. G. Lowe, "Object recognition from local scale-invariant features," The Proceedings of the IEEE International Conference on Computer Vision, vol. 2, pp. 1150-1157, Sep. 1999.

[19] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," Computer Vision and Image Understanding (CVIU), vol. 110, pp. 346-359, 2008.

[20] A. Gil, O. Reinoso, O. Mozos, C. Stachniss, and W. Burgard, "Improving Data Association in Vision-based SLAM," Proceedings of the 2006 IEEE/RSJ, International Conference on Intelligent Robots and Systems, pp. 2076-2081, Oct. 2006

[21] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza *Introduction to Autonomous Mobile Robots*, The MIT Press, 2011, ISBN-13: 978-0262015356.

[22] G. Chabert, and L. Jaulin *Contractor Programming*, Artificial Intelligence, Volume 173, Issue 11, pp. 10791100, Jul. 2009.