

Representing and evaluating ultrasonic maps using active snake contours and Kohonen's self-organizing feature maps

Kerem Altun · Billur Barshan

Received: 2 May 2008 / Accepted: 18 March 2010 / Published online: 4 May 2010
© Springer Science+Business Media, LLC 2010

Abstract Active snake contours and Kohonen's self-organizing feature maps (SOMs) are employed for representing and evaluating discrete point maps of indoor environments efficiently and compactly. A generic error criterion is developed for comparing two different sets of points based on the Euclidean distance measure. The point sets can be chosen as (i) two different sets of map points acquired with different mapping techniques or different sensing modalities, (ii) two sets of fitted curve points to maps extracted by different mapping techniques or sensing modalities, or (iii) a set of extracted map points and a set of fitted curve points. The error criterion makes it possible to compare the accuracy of maps obtained with different techniques among themselves, as well as with an absolute reference. Guidelines for selecting and optimizing the parameters of active snake contours and SOMs are provided using uniform sampling of the parameter space and particle swarm optimization (PSO). A demonstrative example from ultrasonic mapping is given based on experimental data and compared with a very accurate laser map, considered an absolute reference. Both techniques can fill the erroneous gaps in discrete point maps. Snake curve fitting results in more accurate maps than SOMs because it is more robust to outliers. The two methods and the error criterion are sufficiently general that they can also be applied to discrete point maps acquired with other mapping techniques and other sensing modalities.

Keywords Autonomous robots · Mapping · Map errors · Error criterion · Range sensing · Time-of-flight measurements · Ultrasonic sensing · Laser range finders · Snakes · Active contours · Kohonen's self-organizing feature map · Parameter selection · Parameter optimization · Particle swarm optimization

1 Introduction

Autonomous robots must be aware of their environment and interact with it through sensory feedback. The potential of simple and inexpensive sensors should be fully exploited for this purpose before more expensive alternatives with higher resolutions and resource requirements are considered. Ultrasonic sensors have been widely employed because of their accurate range measurements, robustness, low cost, and simple hardware interface. We explore the limits of these sensors in mapping through intelligent processing and representation of ultrasonic range measurements. When coupled with intelligent processing, ultrasonic sensors are a useful alternative to more complex laser and camera systems. Furthermore, it may not be possible to use the latter in some environments due to surface characteristics or insufficient ambient light. Despite their advantages, the frequency range at which air-borne ultrasonic transducers operate is associated with a large beamwidth that results in low angular resolution and uncertainty at the location of the echo-producing feature. Furthermore, ultrasonic range maps are characterized by echo returns resulting from multiple and higher-order reflections, cross-talk between transducers, and noise. These maps are extremely inefficient and unintuitive representations of even the simplest environmental structures that generate them. Thus, having an intrinsic uncertainty of the actual angular direction of the range measurement and being

K. Altun · B. Barshan (✉)
Department of Electrical and Electronics Engineering,
Bilkent University, Bilkent, 06800 Ankara, Turkey
e-mail: billur@ee.bilkent.edu.tr

K. Altun
e-mail: kaltun@ee.bilkent.edu.tr

prone to the various phenomena mentioned above, a considerable amount of processing, interpretation, and modeling of ultrasonic data is necessary.

In this paper, we propose two approaches for efficiently representing and evaluating discrete point maps of an environment obtained with different ultrasonic arc map (UAM) processing techniques. The first approach involves fitting active snake contours (Kass et al. 1987) to the processed UAMs. Snakes are inherently closed curves suitable for representing the features of an environment on a given map. The second approach involves fitting Kohonen's self-organizing feature maps (SOMs) (Kohonen 1982) (which can be implemented as either closed or open curves, using the positions of map points as input features) to an artificial neural network. In ultrasonic maps, gaps frequently occur where a number of contiguous points are marked as empty despite the fact that they are occupied. Both approaches generate parametric curves that fill the erroneous gaps between map points and allow the map to be represented and stored more compactly and smoothly, with fewer points and using a limited number of parameters. These methods also make it possible to compare the accuracy of maps acquired with different techniques or sensing modalities among themselves, as well as to an absolute reference.

In earlier work, there have been two basic approaches to representing ultrasonic data: feature-based and grid-based. Grid-based approaches do not attempt to make difficult geometric decisions early in the interpretation process, unlike feature-based approaches that extract the geometry of the sensor data as the first step. As a first attempt to feature-based mapping, several researchers have fitted line segments to ultrasonic data as features that crudely approximate the room geometry (Crowley 1985; Gex and Campbell 1987; Drumheller 1987). This approach proves to be difficult and brittle because straight lines fitted to time-of-flight (TOF) data do not necessarily match or align with the world model, and may yield many erroneous line segments. Improving the algorithms for detecting line segments and including heuristics do not really solve the problem. A more physically meaningful representation is to use *regions of constant depth* (RCDs) as features, which are circular arcs from specularly reflecting surfaces that are natural features of the raw ultrasonic TOF data. These arcs were first reported in Kuc and Siegel (1987) and further elaborated on in Leonard and Durrant-Whyte (1992). They are obtained by placing a small mark along the line of sight at the range corresponding to the measured TOF value. In specularly reflecting environments, an accumulation of such marks usually produces arc-like features. As a more general approach that is not limited to specularly reflecting surfaces, the angular uncertainty in the range measurements has been represented by UAMs (Başkent and Barshan 1999) that preserve more information (see Fig. 3(a) for a sample UAM). Note

that the arcs in the UAM are uncertainty arcs and different than the arcs corresponding to RCDs. The UAMs are obtained by drawing arcs spanning the beamwidth of the sensor at the measured range, representing the angular uncertainty of the object location and indicating that the echo-producing object can lie anywhere on the arc. The probability of the reflection point being in the middle of the arc is the largest, symmetrically decreasing towards the sides. In the literature, the probability of occupancy along the arc has been modeled by Elfes heuristically (Elfes 1987; Barshan 2007).

Thus, when the same transducer transmits and receives, all that is known is that the reflection point lies on a circular arc of radius r , with a larger probability in the middle of the arc. More generally, when one transducer transmits and another receives, it is known that the reflection point lies on the arc of an ellipse whose focal points are the transmitting and receiving elements. The arcs are tangent to the reflecting surface at the actual point(s) of reflection.

Completely specular and completely diffuse (Lambertian) reflection are both idealizations corresponding to the two extreme cases of the actual spectrum of real reflection characteristics. These two extrema are unreachable in practice and many real environments are, in fact, compositions of both specularly and diffusely reflecting elements. Compared to RCDs, constructing UAMs is more generally applicable in that they can be generated for environments comprised of both specularly and diffusely reflecting surfaces as is the case for many typical indoor environments. On the other hand, RCDs are structures that occur in environments where specular reflections are dominant. In earlier work, techniques based on the Hough transform have been applied to detect line and point features from arcs for both airborne and underwater ultrasonic data (Tardós et al. 2002; Ribas et al. 2007).

In Barshan (2007), the directional maximum technique for processing UAMs was introduced and its performance compared with existing techniques, listed in Table 1. Voting and thresholding (VT), directional maximum (DM), morphological processing (MP), and modified arc-transversal median (ATM-mod) techniques were proposed by our research group; the remaining techniques were developed by other researchers working on ultrasonic sensing. The directional maximum technique was shown to result in more accurate maps of the environment and proved superior in eliminating artifacts and outliers of the UAM. Each processed UAM is comprised of a collection of (usually a large number of) discrete map points. However, most resulting processed UAMs still contain outlier map points due to cross-talk, multiple and higher-order reflections, and noise. Some of the techniques listed in Table 1 process UAMs in such a way that crucial map information is lost, resulting in sparse maps or maps with several erroneous gaps.

Table 1 UAM processing techniques used in this study and their indices

<i>k</i>	UAM processing technique
1	Point marking (PM) (Kuc and Siegel 1987)
2	Voting and thresholding (VT) (Barshan 1999)
3	Directional maximum (DM) (Barshan 2007)
4	Morphological processing (MP) (Başkent and Barshan 1999)
5	Bayesian update (BU) (Elfes 1987)
6	Arc-transversal median (ATM-org) (Choset et al. 2003)
7	Modified ATM (ATM-mod) (Barshan 2007)
8	Triangulation-based fusion (TBF) (Wijk and Christensen 2000b)

In this paper, active snake contours and Kohonen's self-organizing maps are employed to compactly and efficiently represent and evaluate the results of processing UAMs by each technique. Representing the map points with snake curves or SOMs allows us to compare maps obtained with different techniques among themselves, make an assessment of goodness of fit, and compare the results with an absolute reference. For these purposes, we develop and employ a generic error criterion that can be used to compare two different sets of points based on the Euclidean distance measure. The two sets of points can be (i) two different sets of map points acquired with different mapping techniques or different sensing modalities (e.g., ultrasonic data and laser data), (ii) two sets of fitted curve points (e.g., two snake curves) to maps extracted by different mapping techniques or different sensing modalities, or (iii) a set of extracted map points and a set of curve points fitted to them (i.e., assessing goodness of fit). Through the use of a demonstrative example, a comparison with a very accurate laser map of the same environment (considered an absolute reference) is provided based on the error criteria defined. The two approaches are shown to be capable of fitting accurately to features with high curvature, such as edges and corners, as well as smoother features, such as planar walls, in typical indoor environments. They can be conveniently employed to fill the erroneous gaps in discrete point maps. The two approaches are sufficiently general that they can also be applied to discrete point maps acquired with other mapping techniques and other sensing modalities.

This paper is organized as follows: In Sect. 2, a generic error criterion for comparing two different sets of points based on the Euclidean distance measure is presented. Sections 3 and 4 provide some background information on active snake contours and SOMs, respectively. In Sect. 5, we provide a physical interpretation and guidelines on selecting and optimizing the parameters involved in the two approaches, employing uniform sampling of the parameter space and particle swarm optimization. In Sect. 6, the details of representing the extracted map points using active snake

contours and SOMs are provided. The error criterion is used in three different ways to evaluate map accuracy. Using the best-fitting parameter values, the results based on real experimental data are presented. Section 7 provides a discussion and Sect. 8 concludes the paper by providing some directions for possible extensions of this work.

2 The error criterion

Let $P \subset \mathbb{R}^3$ and $Q \subset \mathbb{R}^3$ be two finite sets of arbitrary points with N_1 points in set P and N_2 points in set Q . We do not require the correspondence between the two sets of points to be known. Each point set could correspond to either (i) a set of map points acquired by different mapping techniques or different sensing modalities (e.g., laser, ultrasonic, or infrared map points), (ii) discrete points corresponding to an absolute reference (the true map), or (iii) some curve (in 2-D) or shape (in 3-D) fit to the map points (e.g., polynomials, snake curves, or spherical caps). The absolute reference (or ground truth) could be an available true map or plan of the environment or could be acquired by making range or time-of-flight measurements through a very accurate sensing system.

The well-known Euclidean distance $d(\mathbf{p}_i, \mathbf{q}_j) : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^{\geq 0}$ of the i th point in set P with position vector $\mathbf{p}_i = (p_{xi}, p_{yi}, p_{zi})^T$ to the j th point $\mathbf{q}_j = (q_{xj}, q_{yj}, q_{zj})^T$ in set Q is given by:

$$d(\mathbf{p}_i, \mathbf{q}_j) = \sqrt{(p_{xi} - q_{xj})^2 + (p_{yi} - q_{yj})^2 + (p_{zi} - q_{zj})^2}$$

$$i \in \{1, \dots, N_1\}, j \in \{1, \dots, N_2\} \quad (1)$$

In Barshan (2008), we consider and compare three different metrics to measure the similarity between two sets of points, each with certain advantages and disadvantages. In this work, we use the most favorable of them to measure the closeness or similarity between sets P and Q :

$$\mathcal{E}_{(P-Q)} = \frac{1}{2} \left(\frac{1}{N_1} \sum_{i=1}^{N_1} \min_{\mathbf{q}_j \in Q} \{d(\mathbf{p}_i, \mathbf{q}_j)\} \right. \\ \left. + \frac{1}{N_2} \sum_{j=1}^{N_2} \min_{\mathbf{p}_i \in P} \{d(\mathbf{p}_i, \mathbf{q}_j)\} \right) \quad (2)$$

According to this criterion, we take into account all points in the two sets and find the distance of every point in set P to the nearest point in set Q and average them, and vice versa. The two terms in (2) are also averaged, so that the criterion is symmetric with respect to P and Q . If the two sets of points are completely coincident, the average distance between them will be zero. If one set is a subset of the other, there will be some error. Had an asymmetric criterion been

employed, say, including only the first (or the second) term in (2), the error would have been zero when $P \subset Q$ (or $Q \subset P$). Gaps occurring in the maps and sparsity are penalized by the error criterion, resulting in larger errors on average.

The error criterion we propose is sufficiently general that it can be used to compare any two arbitrary sets of points. This makes it possible to compare the accuracy of discrete point maps acquired with different techniques or sensing modalities with an absolute reference, as well as among themselves, both in 2-D and 3-D. When curves or shapes (e.g., lines, polynomials, snakes, spherical or elliptical caps) are fitted to the map points, the criterion proposed here also enables us to assess the goodness of fit of the curve or shape to the map points. In other words, a fitted curve or shape comprised of a finite number of points can be treated in exactly the same way.

3 Fitting active snake contours to UAMs

A snake, or an active contour, first introduced by Kass et al. (1987), can be described as a continuous deformable closed curve. Active snake contours have been commonly used in image processing for edge detection and segmentation (Kass et al. 1987; Menet et al. 1990; Cohen 1991; Cohen and Cohen 1993), and have been mostly classified into three categories (Jacob et al. 2004):

- point-based snakes, where the curve is represented as a collection of discrete points,
- parametric snakes, where the curve is described using combinations of basis functions, and
- geometric snakes, where the curve is represented as a level set of a higher-dimensional surface.

This classification is not universal and many authors categorize snakes merely into two, namely parametric and geometric, with the first category stated above considered a special kind of parametric snake. The snake used in this study belongs to the first category. The variants of snakes are analyzed in a unified manner in Liang et al. (2006).

We define a snake as a parameterized closed curve $\mathbf{v}(s) = [p_x(s), p_y(s)]^T$, $s \in [0, 1]$, where $p_x(s)$ and $p_y(s)$ are functions representing the Cartesian coordinates of the snake in 2-D and s is the normalized arc length parameter of the snake curve. This parameterization is dimensionless. The deformation of the snake is controlled by internal and external forces. Internal forces impose elasticity and rigidity constraints on the curve, whereas external forces stretch or shrink the curve to fit to the image data. The total energy of the snake curve is given by the functional

$$E_{\text{snake}} = \int_0^1 [E_{\text{int}}(\mathbf{v}(s)) + E_{\text{ext}}(\mathbf{v}(s))] ds \tag{3}$$

The internal energy component is given by

$$E_{\text{int}}(\mathbf{v}(s)) = \frac{1}{2} \left(\alpha \left\| \frac{d(\mathbf{v}(s))}{ds} \right\|^2 + \beta \left\| \frac{d^2(\mathbf{v}(s))}{ds^2} \right\|^2 \right) \tag{4}$$

where α is the elasticity parameter and β is the rigidity parameter, and $\| \cdot \|$ denotes the 2-norm. The first derivative term in (4) penalizes long curves, whereas the second derivative term penalizes sharp curvatures. This internal energy definition is used in many applications, possibly with varying $\alpha(s)$ and $\beta(s)$. The use of other internal energy expressions is not very common.

The external energy component is denoted by $E_{\text{ext}}(\mathbf{v}(s)) = U(\mathbf{v}(s))$, where U is a potential function that depends on the image data. In general, the potential function can be selected in different ways, depending on the application. However, it must be at minimum on the edges of the image if the snake is to be used for edge detection, segmentation, or finding the boundaries of an environment as in our application. Kass et al. (1987) suggest using the negative of the image gradient magnitude as a potential function. However, this is only feasible if the snake is initialized close to the image boundaries, otherwise the snake curve would be stuck in local minima or a flat region of the potential function. Filtering the image with a Gaussian low-pass filter is also suggested in the same paper to increase the capture range of the snake, but this causes the edges to become blurry, thus reducing map accuracy. In black-on-white and gray-level images, the image intensity can be used as the potential function, either in binary form or convolved with a Gaussian blur (Cohen 1991). Obviously, this method also suffers from the drawbacks stated above. Another solution, proposed in Cohen and Cohen (1993), is using a distance map as a potential function to increase the capture range of the contour, which is the approach used in this study.

In this work, we chose to use a potential function for the external energy term based on the Euclidean distance map, as suggested in Cohen and Cohen (1993). Although our problem is in 2-D, let us first make a more general definition of a distance map in 3-D. Let $Q \subset \mathbb{R}^3$ be a finite set of arbitrary points. For all points \mathbf{p} of the mapped region, we define a distance map $D_Q(\mathbf{p}) : \mathbb{R}^3 \rightarrow \mathbb{R}^{\geq 0}$ between a point \mathbf{p} and set Q as the minimum of the Euclidean distances of that point to all the points in the set Q . That is,

$$D_Q(\mathbf{p}) = \min_{\mathbf{q}_j \in Q} \{d(\mathbf{p}, \mathbf{q}_j)\} \quad j \in \{1, \dots, N_2\} \tag{5}$$

where \mathbf{q}_j is the position vector of the j th point in the set Q . According to this definition, the two summands in (2) are, in fact, nothing but distance functions and the error criterion can be rewritten as:

$$\mathcal{E}_{(P-Q)} = \frac{1}{2} \left(\frac{1}{N_1} \sum_{i=1}^{N_1} D_Q(\mathbf{p}_i) + \frac{1}{N_2} \sum_{j=1}^{N_2} D_P(\mathbf{q}_j) \right) \tag{6}$$

Computing the Euclidean distance map is costly, and a number of algorithms and other distance functions have been proposed in the literature to approximate it (Borgefors 1986; Rosenfeld and Pfaltz 1968). In this study, the Euclidean distance map is implemented in its original form and the potential function is chosen as:

$$U(\mathbf{p}) = D_Q(\mathbf{p}) = \min_{\mathbf{q}_j \in Q} \{d(\mathbf{p}, \mathbf{q}_j)\} \quad j \in \{1, \dots, N_2\} \quad (7)$$

for all points \mathbf{p} of the mapped region and a point set Q .

Approaches that do not use a potential function as the external energy term also exist in the literature (Xu and Prince 1998). Such approaches relax the constraint that the external forces pulling the snake towards the edges should be conservative, i.e., derived from a potential field. For example, Xu and Prince (1998) define a non-conservative force field representing the external forces and use force-balance equations rather than an energy-based approach to solve the problem.

With the above definitions of external and internal energy, calculus of variations can be used to find the curve that minimizes the energy functional in (3). The minimizing curve should satisfy the following Euler-Lagrange equation (Kass et al. 1987):

$$\alpha \frac{d^2 \mathbf{v}(s)}{ds^2} - \beta \frac{d^4 \mathbf{v}(s)}{ds^4} - \nabla U(\mathbf{v}(s)) = 0 \quad (8)$$

Although it may be possible to solve this equation analytically for some special cases, a general analytical solution does not exist. The common practice is to initialize an arbitrary time-dependent snake curve $\mathbf{v}(s, t)$. Equation (8) is then set equal to the time derivative of the snake, where a solution is found when the time derivative vanishes. That is,

$$\alpha \frac{\partial^2 \mathbf{v}(s, t)}{\partial s^2} - \beta \frac{\partial^4 \mathbf{v}(s, t)}{\partial s^4} - \nabla U(\mathbf{v}(s, t)) = \frac{\partial \mathbf{v}(s, t)}{\partial t} \quad (9)$$

These equations are then discretized to find a numerical solution. The snake is treated as a collection of discrete points joined by straight lines and is initialized on the image. Approximating the derivatives by finite differences, the evolution equations of the snake reduce to the following:

$$\mathbf{p}_x(n+1) = (\mathbf{A} + \gamma \mathbf{I})^{-1} \left(\gamma \mathbf{p}_x(n) - \kappa \frac{\partial U}{\partial p_x} \Big|_{[\mathbf{p}_x(n), \mathbf{p}_y(n)]} \right) \quad (10)$$

$$\mathbf{p}_y(n+1) = (\mathbf{A} + \gamma \mathbf{I})^{-1} \left(\gamma \mathbf{p}_y(n) - \kappa \frac{\partial U}{\partial p_y} \Big|_{[\mathbf{p}_x(n), \mathbf{p}_y(n)]} \right) \quad (11)$$

Here, n is the current time (or iteration) step, $\mathbf{p}_x(n)$ and $\mathbf{p}_y(n)$ are vectors representing the positions of the collection of discrete points on the snake at time n , γ is the Euler

step size, and κ is a weight factor for the external force. \mathbf{I} is the identity matrix of the appropriate size and \mathbf{A} is a pentadiagonal banded matrix that depends on α and β . The sizes of the matrices \mathbf{A} and \mathbf{I} are determined by the number of points on the snake, which may change as the algorithm is executed.

4 Fitting self-organizing maps to UAMs

Another method used in map representation and in evaluating the different techniques is the self-organizing map introduced by Kohonen (1982), which is basically an artificial neural network that uses a form of unsupervised learning, and is suitable for applications where the topology of the data is to be learned. Robots that learn the environment structure using artificial neural networks are reported on in Kurz (1996) and Yamada (2004). SOMs for curve and surface reconstruction have been used in applications such as computer-aided design (CAD) modeling of objects having irregular shapes (Kumar et al. 2004; Knopf and Sangole 2004). In this paper, we use the SOM for fitting curves to the ultrasonic map points obtained with the different UAM processing techniques.

An SOM is an artificial neural network with two layers. We use a 1-D SOM, whose structure is illustrated in Fig. 1. The two neurons at the input layer are used to input the p_x and p_y coordinates of a map point. Each neuron at the output layer represents a point on the curve to be fitted, and the associated connection weights are the p_x and p_y coordinates of this point. The output neurons are arranged as a chain-like structure, where each neuron, except those at the two ends, has two neighbors. This neighborhood affects the weight updates described below. For each input map point, the winning neuron is determined to be the closest point on the curve to that input. Thus, for the input map point $\mathbf{p} = (p_x, p_y)^T$, output neuron weights $\mathbf{w}_i = (w_{1i}, w_{2i})^T$, and a total of N points on the curve, the index i^* of the

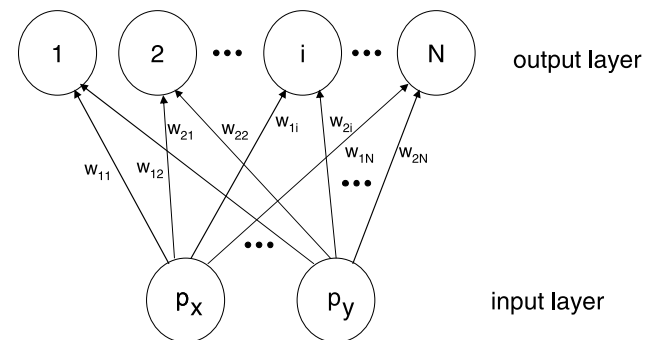


Fig. 1 Structure of the 1-D SOM

winning neuron is given by:

$$i^* = \arg \min_{i=1, \dots, N} \sqrt{(p_x - w_{1i})^2 + (p_y - w_{2i})^2} \quad (12)$$

Through the use of a Gaussian function $g_{0, \sigma(n)}(\cdot)$, updating the weights is done such that the weight update of one neuron also affects the neighboring neurons. Then, for all neurons $i = 1, \dots, N$, the weight update rule is

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \lambda(n)g_{0, \sigma(n)}(|i - i^*|)[\mathbf{p}(n) - \mathbf{w}_i(n)] \quad (13)$$

where n is the iteration step, $\mathbf{w}_i(n)$ is the 2×1 weight vector of neuron i , $\mathbf{p}(n)$ is the position vector of the input map point, $\lambda(n)$ is the time-dependent learning rate, and $g_{0, \sigma(n)}(\cdot)$ is a 1-D Gaussian function with zero mean and standard deviation $\sigma(n)$. Note that $\lambda(n)$ and $\sigma(n)$ are functions of the iteration number n and should be decreased at the end of each epoch as the iterations are performed, with respective decay coefficients k_λ and k_σ . An epoch is completed when all map points are given as input to the SOM once.

5 Parameter selection and optimization

Active snake contours and SOMs have a number of parameters that affect the convergence characteristics and performance of curve fitting. Since each problem or situation requires a different set of parameter values, universally accepted values for these parameters do not exist in the literature. For the map points extracted from an unknown environment, it is difficult to guess or estimate these parameter values beforehand. The parameters depend on the shape and the nature of the environment as well as on the mapping technique used. For example, a structured environment mostly composed of specularly reflecting surfaces may require a different set of parameters than an unstructured environment with some combination of specularly and diffusely reflecting surfaces. Optimization methods can be applied to select the best parameter values specific to the problem at hand. Below, we provide guidelines for selecting these parameters and suggest two alternative approaches that can be employed for this purpose.

For the active contour method, the parameters that must be selected are α , β (4), γ , and κ ((10) and (11)). As stated above, α penalizes elongation and β penalizes bending or sharpness of the snake curve. For example, selecting a small β value enforces the second derivative in the energy term to have smaller weight, thus allowing sharp corners in the snake. γ is the Euler step size of the discretization and κ is a weight factor for the external force. Some authors use $\kappa = 1$, as in the original definition, where it is also stated that the

step size γ should be reduced if the external force becomes large (Kass et al. 1987). Including a κ parameter (different than one) provides more user control of the problem. Determining these parameters depends on the initialization and the required curve features, such as length and curvature.

For the SOM method, the parameters that affect convergence and the performance of curve fitting are the learning rate $\lambda(n)$ and the standard deviation $\sigma(n)$. These are functions of the iteration number n and should be decreased as the iterations are performed. In our implementation, we multiply each with decay coefficients $0 < k_\lambda < 1$ and $0 < k_\sigma < 1$ at the end of each epoch. Thus, the parameters to be chosen are $\lambda(0)$, $\sigma(0)$, k_λ , and k_σ . The initial learning rate $\lambda(0)$ is usually chosen to be less than one in order not to “overshoot” the map points in fitting. If the curve is initialized close to the boundaries of the mapped region, $\lambda(0)$ and $\sigma(0)$ should be chosen smaller. The Gaussian function has a value of one at its peak point, where $i = i^*$, and decreases symmetrically and gradually. Using a Gaussian function in (13) to update the weights allows all neurons to be updated at the same time, resulting in a smoother fitting curve to the input data at each iteration. If we visualize the 1-D SOM as a chain-like structure, $\sigma(n)$ determines the tightness/looseness of the chain. The larger it is, the tighter the chain, and vice versa. Its initial value depends on the length and position of the initial curve and the total number of points on that curve.

Since finding the best-fitting parameters depends on many factors, it is not possible to determine and fix them for every purpose beforehand. In this work, we followed two different approaches to estimate the best parameter values for a given set of map points.

In the first approach, we sample the parameter space uniformly and search for parameter sets that minimize the error. For example, in snake fitting, since there are four parameters to be estimated, the parameter space is a 4-D hypercube. We divide this hypercube evenly into a sufficient number of smaller-sized hypercubes and use the parameter values that correspond to the center of each hypercube. In other words, each parameter is uniformly incremented and all possible combinations of the selected parameter values are considered. The parameter values that resulted in the minimum error for both approaches are presented in the first line of Table 2.

We used particle swarm optimization as the second approach (Kennedy and Eberhart 1995). This method is inspired by the idea of swarms in nature in order to find the minimum (or, without loss of generality, maximum) value of an objective function. Here, we outline the method as implemented in this study. For a more detailed overview of the original form of the algorithm and its many variants, we refer the reader to Poli et al. (2007).

The swarm is composed of a number of particles, each of which is a candidate for a minimum. At an arbitrary iteration

Table 2 Best parameter values found by (i) uniform sampling of the parameter space, and (ii) initializing PSO with the best parameters of (i)

Method	Snake curve fitting				SOM			
	α	β	γ	κ	$\lambda(0)$	$\sigma(0)$	k_λ	k_σ
(i) Uniform sampling	4.20	0.60	0.60	1.80	0.03	2.59	0.83	0.76
(ii) PSO (initialized by (i))	4.58	1.23	0.98	1.72	0.02	2.85	0.90	0.76

n of the algorithm, each particle in the parameter space has a position \mathbf{x}_n , a velocity $\dot{\mathbf{x}}_n$, a “previous best” value $pBest_n$, and the corresponding previous best position \mathbf{p}_n . The best value would be the lowest (highest) value of the objective function achieved by the particle so far, for a minimum (maximum). We denote the value of the objective function at a position \mathbf{x} by $J(\mathbf{x})$. Furthermore, the minimum (maximum) of $pBest_n$ among all particles is denoted as the global best, $gBest_n$, and the corresponding position as \mathbf{g}_n . The velocity and position of each particle is updated according to the equations:

$$\dot{\mathbf{x}}_{n+1} = w_n \dot{\mathbf{x}}_n + c_1 \varphi_1(\mathbf{p}_n - \mathbf{x}_n) + c_2 \varphi_2(\mathbf{g}_n - \mathbf{x}_n) \quad (14)$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \dot{\mathbf{x}}_{n+1} \quad (15)$$

where w_n is called the inertia weight, c_1 and c_2 are constants, and φ_1 and φ_2 are independent random variables uniformly distributed in the interval $[0, 1]$. In this study, we take $c_1 = c_2 = 2$, as suggested in Kennedy and Eberhart (1995). The inertia weight w_n serves as a memory coefficient and prevents the particles from taking “sharp turns” during the iterations, resulting in a smoother movement of the particles.

At each step of the algorithm, the values $pBest_n$ and $gBest_n$ are checked to see if they should be updated. In searching for a minimum, if the current value $J(\mathbf{x}_n)$ of a particle is less than its $pBest_n$ value, $pBest_n$ is set to the current value and \mathbf{p}_n is set to its current position \mathbf{x}_n . Similarly, if the current value $J(\mathbf{x}_n)$ of any particle is less than the $gBest_n$ value, $gBest_n$ and \mathbf{g}_n are updated accordingly.

We used the PSO algorithm summarized above to optimize the parameters for the active contour and SOM methods. The objective function to be minimized is selected as the average error between the fitted curves and the absolute reference for the map, obtained by a very accurate laser system. We initialized the parameter values in two different ways. First, random initialization is considered: For the active contour method, all four parameters of each of the 20 particles of PSO are initialized randomly in the interval $[0, 12]$. For the SOM method, $\lambda(0)$ parameter is initialized randomly in $[0, 0.5]$, k_λ and k_σ are initialized randomly in $[0, 1]$, and $\sigma(0)$ is initialized randomly in $[0, 20]$. However, random initialization did not result in the smallest error values. As another alternative, we used the best parameter values found by uniform sampling and added white Gaussian noise to them to initialize each particle of PSO. For

the active contour method, all parameters were added white Gaussian noise with standard deviation 0.2. For SOM, the standard deviations of the Gaussian noise added to the four parameters were 0.01, 0.4, 0.02, and 0.02, respectively. This approach reduced the errors considerably, as well as improving the convergence speed of PSO. The parameter values that resulted in the minimum error for active contours and SOM are presented in the second line of Table 2.

6 Experiments

We obtained our experimental results using the front three ultrasonic sensors and structured-light system of the Nomad 200 robot. We used a simple rule-based wall-following algorithm for the indoor environment shown in Figs. 2 and 3(a). The environment is a small room—approximately 2.75 m square—with four corners and an edge feature. There is a cater-corner opening in the lower left corner of the room from which no ultrasonic or structured-light data were received. Referring to Fig. 3, the environment is comprised of smooth wooden (top and left) and painted (right) walls, and a window shade with vertical slats of 15 cm width (bottom). Some of the corners of the room are not perfect (e.g., where the shade and the right wall make a corner).

To demonstrate our methodology, we used the mapping results of the different UAM processing techniques listed in Table 1 and described in detail in Barshan (2007). Each of these techniques results in a different set of map points, to which both a snake curve and an SOM are fitted in 2-D.

Let each set of processed UAM data points be denoted as M_k , where k corresponds to one of the UAM processing techniques indexed in Table 1. For compatibility, let the set of original laser data points be denoted as M_0 . Then, for the laser map ($k = 0$) and for the k th ultrasonic map ($k = 1, \dots, 8$), the potential function used in fitting the k th snake is selected as

$$U_k(\mathbf{p}) = D_{M_k}(\mathbf{p}) \quad k = 0, 1, \dots, 8 \quad (16)$$

for all points \mathbf{p} . Here, $D_{M_0}(\mathbf{p})$ denotes an element of the Euclidean distance map with respect to the original laser data and $D_{M_k}(\mathbf{p})$ denotes an element of the distance map with respect to the k th processed UAM. Note that the value of the potential function is zero for those points on the image corresponding to the extracted map, and increases gradually with increasing distance of the point \mathbf{p} from the map points.



(a)



(b)

Fig. 2 Views of the environment in Fig. 3(a): (a) looking towards the right, showing the top, right, and bottom walls; (b) looking towards the lower right corner, showing the right and bottom walls in Fig. 3(a). The cylinder is an additional feature

In the following, the set of curve points fitted to the map points resulting from the k th UAM processing technique will be referred to as S_k where $k = 1, \dots, 8$. That is, the k th curve is represented as a collection of points $\mathbf{p}_{ik}, i = 1, \dots, N_k$, where N_k is the total number of points on curve S_k . The set of points of the curve fitted to the original laser data will be referred to as S_0 .

The map of the environment acquired with a structured-light laser system is shown in Fig. 3(b). This is the original laser data, which is quite accurate, and it is used as the absolute reference. The corresponding Euclidean distance map is shown in Fig. 3(c), and is drawn by rescaling the values of the potential function to be between zero and 255. In the distance map, the darkest points of value zero correspond to a distance of zero and, after normalization, the lightest

Table 3 Values of some experimental quantities

Map size	525 × 525
Initial curve center	(30, 55)
Initial curve radius	185
Number of snake points	400–500
Number of snake iterations	250
Intensity range of distance map	0–255
Number of SOM epochs	20
Number of SOM input neurons	2
Number of SOM output neurons	160
Number of PSO particles	20
Number of PSO iterations	20
Number of samples of parameter space	1,296

points correspond to the value 255. The snake fitted to this laser data is shown in Fig. 3(d), superimposed on the data points. Figure 3(e) shows the SOM fitted to the same laser data. These will be used for comparison in Sects. 6.1 and 6.2. Values of some quantities used in the experiments are provided in Table 3.

6.1 Results of fitting active contours

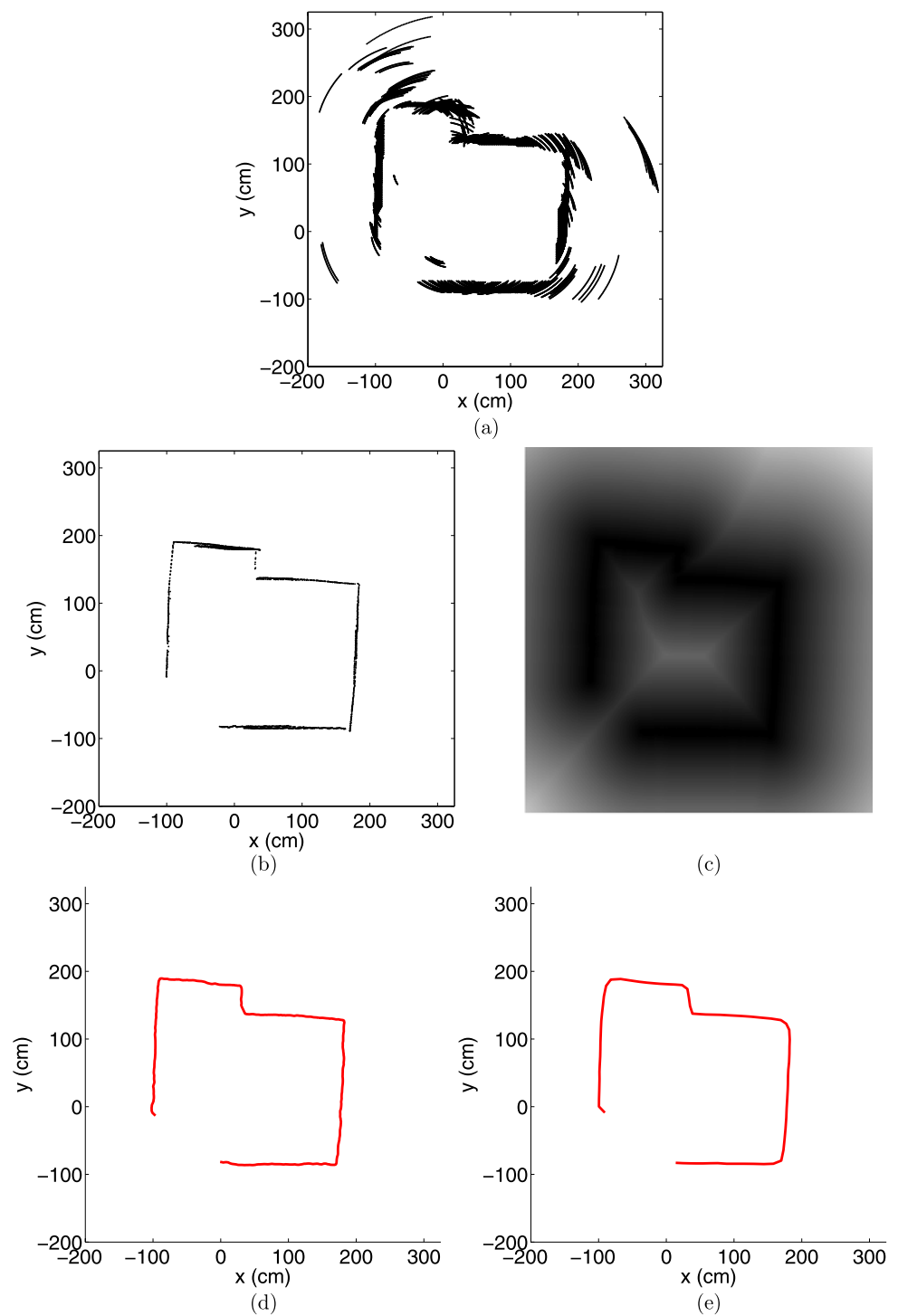
The snake is initialized as a circle whose center is at (30, 55) with a radius of 185 units to encompass the room boundary (Fig. 3). Then, the snake is evolved for a fixed number of iterations (250). After each iteration, the points on the snake are checked for uniformity. The distance between any two neighboring points is maintained between two and four units, determined experimentally. That is, after each iteration, the points are deleted or created as required by this constraint. We allow the snake to converge to outlier points caused by cross-talk, multiple, and higher-order reflections to provide a fair evaluation of the different techniques.

Using the two-sided error criterion defined in (2), we define the error of the fit at iteration n as follows:

$$\begin{aligned} \mathcal{E}_{(S_k(n)-M_k)}(n) &= \frac{1}{2} \left(\frac{1}{N_k(n)} \sum_{i=1}^{N_k(n)} D_{M_k}(\mathbf{p}_{ik}(n)) + \frac{1}{L_k} \sum_{j=1}^{L_k} D_{S_k(n)}(\mathbf{q}_{jk}) \right) \\ k &= 0, \dots, 8 \end{aligned} \tag{19}$$

Here, $n = 1, \dots, 250$ is the iteration step, $\mathbf{p}_{ik}(n)$ is the position vector of the i th point on snake k at iteration n , \mathbf{q}_{jk} is the position vector of the j th point of M_k , $N_k(n)$ is the number of points on snake k at step n , and L_k is the number of points of M_k . Note that if the snake fits perfectly to the ex-

Fig. 3 (Color online) **(a)** The raw UAM, **(b)** original laser map, **(c)** distance map with respect to the laser data, **(d)** snake fitted to the laser data, **(e)** SOM fitted to the laser data



tracted map points at any iteration (in other words, if every snake point corresponds to a map point and vice versa), this error becomes zero, since each of the summed distance values in (17) would be zero. We calculate and store this error for each iteration. Then, the snake curve that results in the minimum error is determined and selected as the snake that best represents the corresponding map points.

In this paper, we used a fixed number of iterations and selected the snake that results in the minimum error. Our observations reveal that the error decreases to a certain value between iterations 100 and 150 and then oscillates around that value, which does not affect the results significantly. In a practical application, it is also possible to take the error at the end of a fixed number of iterations or to set an error

threshold to stop the iterations when the error goes below the threshold (Altun and Barshan 2008).

To evaluate map accuracy based on the generic error criterion given in (2), we chose the point sets P and Q in three different ways.

In the first, we compare the snake fitted to processed UAM points with the originally acquired laser data points so that the two point sets are S_k and M_0 . Using the notation at the beginning of this section, the minimum distance of point i on snake S_k to the set M_0 is given by $D_{M_0}(\mathbf{p}_{ik})$. Then, the error is given as:

$$\mathcal{E}_{(S_k-M_0)} = \frac{1}{2} \left(\frac{1}{N_k} \sum_{i=1}^{N_k} D_{M_0}(\mathbf{p}_{ik}) + \frac{1}{L_0} \sum_{j=1}^{L_0} D_{S_k}(\mathbf{q}_{jk}) \right) \quad k = 0, \dots, 8 \tag{18}$$

For our example, when $k = 0$, $\mathcal{E}_{(S_0-M_0)} = 1.07$, indicating that the average error of the laser snake fit to the original laser data is about one pixel. The errors for the other k values are tabulated in Tables 4 and 5.

In the second criterion, we compare the processed UAM snake S_k with the snake curve S_0 fitted to the original laser data (Fig. 3(d)). We calculate the distance of every point on the snake S_k to the nearest point on the snake S_0 and average

these distances, and vice versa:

$$\mathcal{E}_{(S_k-S_0)} = \frac{1}{2} \left(\frac{1}{N_k} \sum_{i=1}^{N_k} D_{S_0}(\mathbf{p}_{ik}) + \frac{1}{N_0} \sum_{j=1}^{N_0} D_{S_k}(\mathbf{q}_{jk}) \right) \quad k = 0, \dots, 8 \tag{19}$$

$$\approx \frac{1}{2} \left(\frac{\int_{S_k} D_{S_0}(\mathbf{p}) ds}{\int_{S_k} ds} + \frac{\int_{S_0} D_{S_k}(\mathbf{q}) ds}{\int_{S_0} ds} \right) \tag{20}$$

Note that $\mathcal{E}_{(S_0-S_0)} = 0$ by definition. The summations in (19) are, in fact, discrete approximations of the line integrals of the distance map functions given in (20).

The third way measures how well the k th snake fits to the ultrasonic map points of the k th technique, and is not related to the reference laser data. It corresponds to the minimum error that is obtained during the snake iterations using (17), and is given by:

$$\mathcal{E}_{(S_k-M_k)} = \frac{1}{2} \left(\frac{1}{N_k} \sum_{i=1}^{N_k} D_{M_k}(\mathbf{p}_{ik}) + \frac{1}{L_k} \sum_{j=1}^{L_k} D_{S_k}(\mathbf{q}_{jk}) \right) \quad k = 1, \dots, 8 \tag{21}$$

The errors of snake fitting for the different UAM processing techniques are tabulated in the third, fourth, and fifth columns of Tables 4 and 5. The results in Table 4 are obtained by using the best-fitting parameter values obtained by

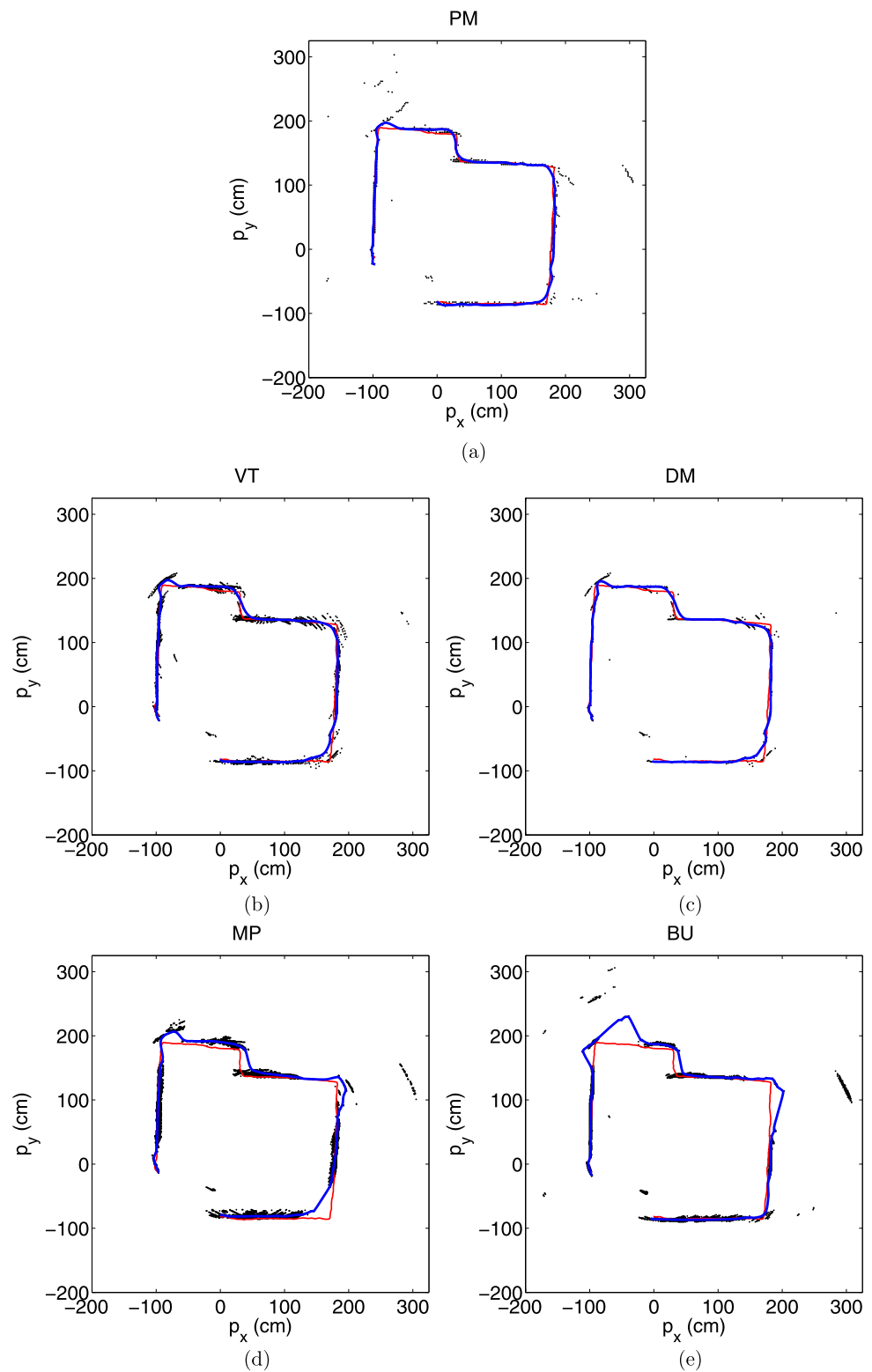
Table 4 Error values (in pixels) for snake curve fitting and SOM using parameters found by uniform sampling of the parameter space

k	Method	Snake curve fitting			SOM		
		$\mathcal{E}_{(S_k-M_0)}$	$\mathcal{E}_{(S_k-S_0)}$	$\mathcal{E}_{(S_k-M_k)}$	$\mathcal{E}_{(S_k-M_0)}$	$\mathcal{E}_{(S_k-S_0)}$	$\mathcal{E}_{(S_k-M_k)}$
1	PM	2.71	2.29	3.77	6.06	7.12	4.67
2	VT	2.81	2.51	1.87	2.91	4.13	2.25
3	DM	2.69	2.63	1.98	2.86	3.77	2.54
4	MP	4.82	5.14	2.95	7.09	8.17	2.69
5	BU	5.89	5.35	4.28	9.32	10.38	2.89
6	ATM-org	2.97	2.58	3.07	5.81	6.69	3.73
7	ATM-mod	3.11	3.02	2.70	4.21	5.56	2.79
8	TBF	4.00	4.63	4.62	5.22	6.59	4.60

Table 5 Error values (in pixels) for snake curve fitting and SOM using PSO parameters

k	Method	Snake curve fitting			SOM		
		$\mathcal{E}_{(S_k-M_0)}$	$\mathcal{E}_{(S_k-S_0)}$	$\mathcal{E}_{(S_k-M_k)}$	$\mathcal{E}_{(S_k-M_0)}$	$\mathcal{E}_{(S_k-S_0)}$	$\mathcal{E}_{(S_k-M_k)}$
1	PM	2.58	2.13	4.01	5.46	6.02	4.02
2	VT	2.78	2.45	1.91	2.85	3.65	2.18
3	DM	2.74	2.44	1.57	2.65	3.11	2.24
4	MP	5.02	5.39	2.90	6.88	7.46	2.63
5	BU	5.92	5.38	4.34	9.10	9.50	2.53
6	ATM-org	2.79	2.36	2.97	5.37	6.07	3.23
7	ATM-mod	2.97	2.91	2.39	3.99	4.79	2.54
8	TBF	4.28	4.93	4.68	4.81	5.67	4.21

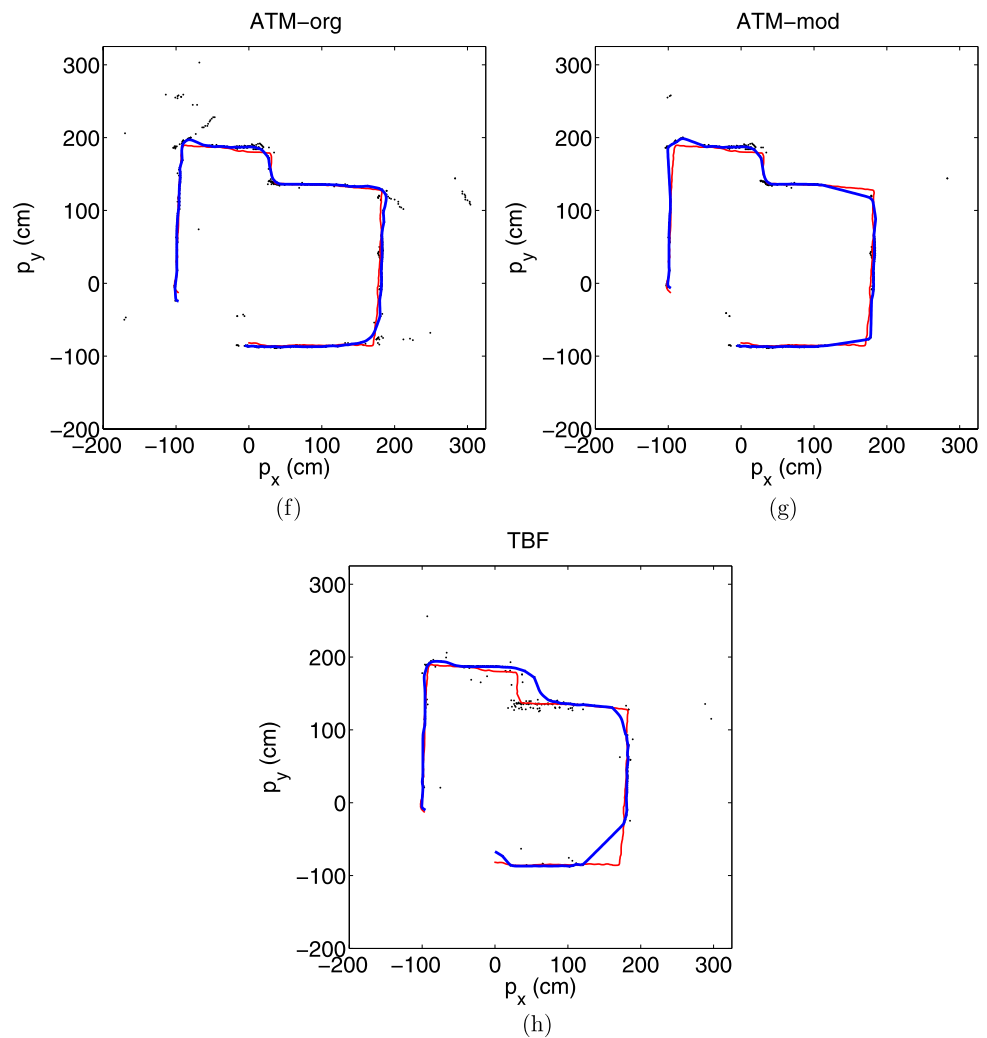
Fig. 4 (Color online) Results of snake fittings for (a) PM, (b) VT, (c) DM, (d) MP, (e) BU, (f) ATM-org, (g) ATM-mod, and (h) TBF



uniform sampling of the parameter space (first line of Table 2). Similarly, the parameter values found by initializing PSO with the uniform sampling parameters (second line of Table 2) are used to obtain the results in Table 5. In both

tables, it can be observed that $\mathcal{E}_{(S_k-M_0)}$ and $\mathcal{E}_{(S_k-S_0)}$ values are mostly comparable with each other since both take the laser data as reference, in original and snake-fitted forms, respectively. According to the results, PM, DM, and VT tech-

Fig. 4 (Continued)



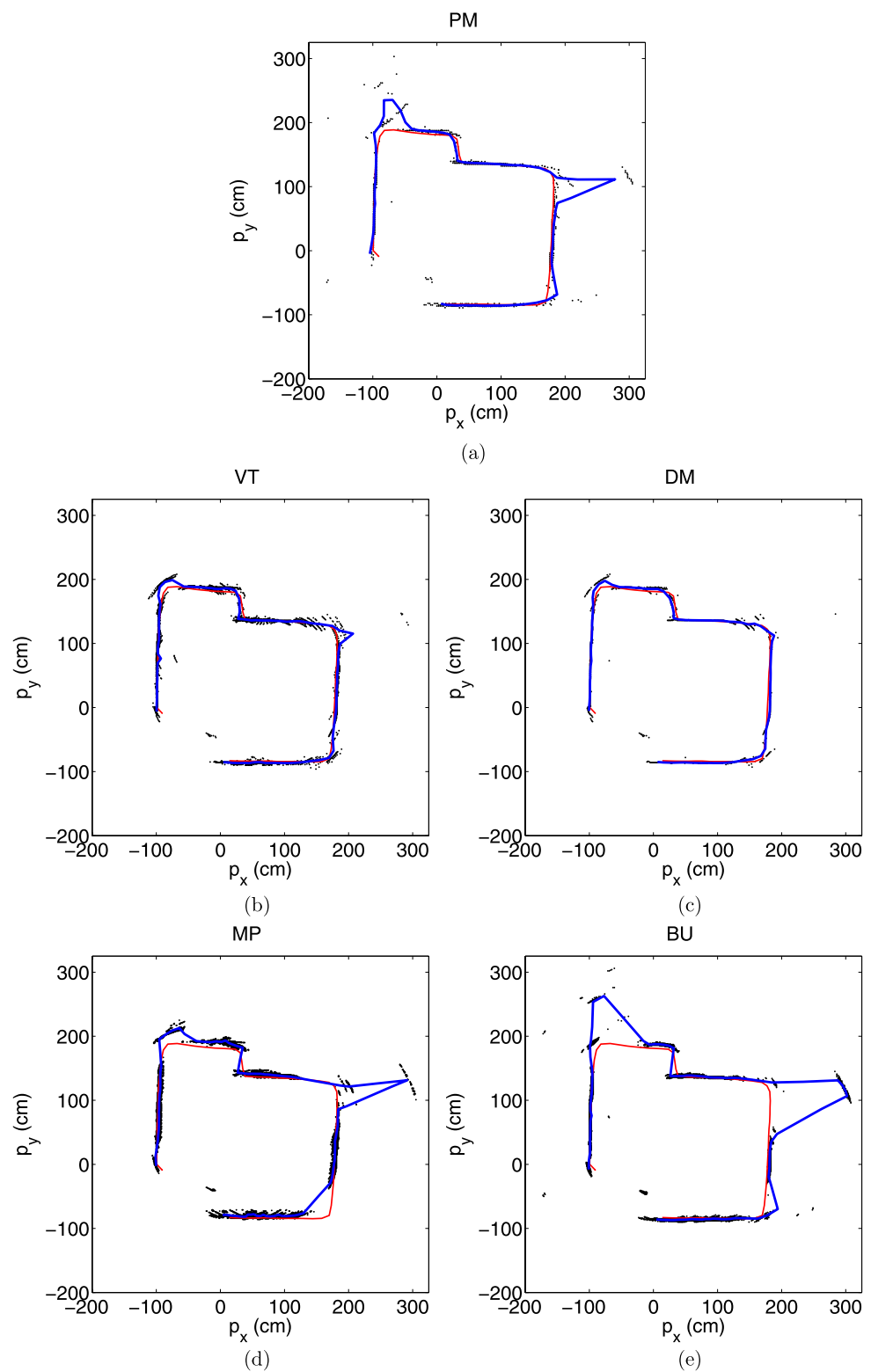
niques have the smallest errors, and MP and BU perform the worst because these latter processing techniques result in more spurious points on the extracted map. The remaining techniques are comparable with each other. It can also be observed that PM and ATM-org are the only methods for which the $\mathcal{E}_{(S_k-M_k)}$ value is larger than $\mathcal{E}_{(S_k-M_0)}$ and $\mathcal{E}_{(S_k-S_0)}$, indicating that the snake curves for these methods fit better to the laser data than the processed UAMs. This is because unlike most of the other methods, many outlying points remain after processing the UAM with these methods; the snake fits poorly to these outliers and this causes the error $\mathcal{E}_{(S_k-M_k)}$ to increase.

Because smaller errors are obtained by uniform sampling of the parameter space followed by PSO, the best-fitting parameter values found with this approach are used for the given illustrations. The snake curves fitted to the processed UAMs and the laser data are illustrated in Fig. 4. In the different parts of the figure, the black features correspond to map points obtained with a particular UAM processing technique. The thick lines (blue lines in the color ver-

sion online) are the snakes fitted to these map points. The curve in thin lines (red curve in the color version online) is the snake fitted to the laser data, which is the same in each part of the figure and is included as a reference for visual comparison. This is also the same curve as in Fig. 3(d).

Note that in this study, the snake curves are initialized outside the boundaries of the room because the curve tends to shrink rather than expand due to the first derivative term in the energy expression. This fact should be taken into account in determining the initial location of the snake curve. Initializing the snake within the boundaries of the environment is also a possibility, as the spurious points outside the boundaries would not affect the snake curve as much, allowing it to follow the boundaries of the room more closely. However, inside initialization would not result in a fair comparison between the techniques in terms of the amount of spurious points left after UAM processing. In addition, in some mapping applications one may not be completely free to choose the initial location. For example, in a room with

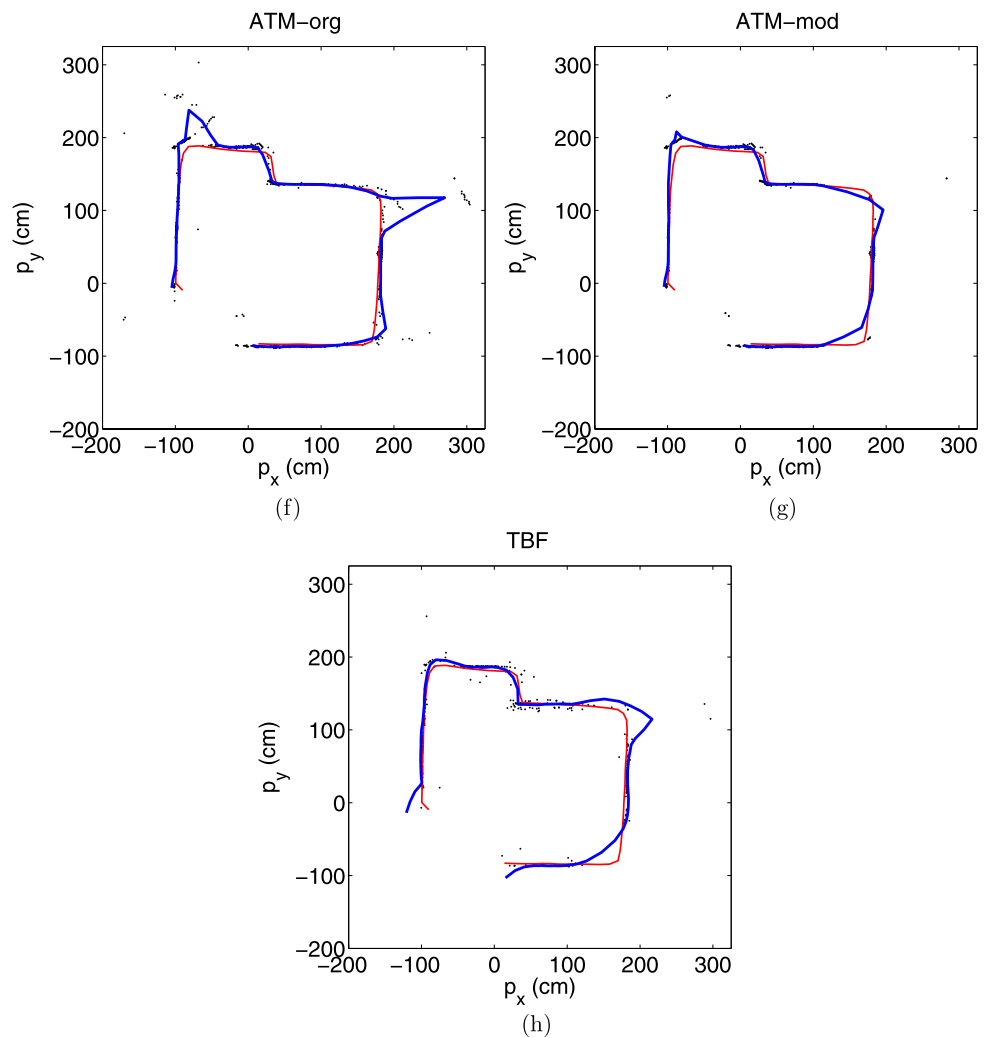
Fig. 5 (Color online) Results of SOMs for (a) PM, (b) VT, (c) DM, (d) MP, (e) BU, (f) ATM-org, (g) ATM-mod, and (h) TBF



many obstacles close to the room boundaries, it would be essential to initialize the snake curve outside in order to represent the boundaries of the room correctly. However, if detecting obstacle boundaries is more important, one would

initialize the snake inside the boundaries. In fact, some applications may require both. The choice for the initial location should depend on the configuration of obstacles and the free space.

Fig. 5 (Continued)



6.2 Results of fitting self-organizing maps

We initialize an SOM with 160 neurons as a circle outside the boundaries of the room, with the same center and the same radius as in the snake curve fitting procedure. The iterations are stopped after 20 epochs. Similar to the case with the snake curve, the distance between neighboring points on the curve is maintained between 10 and 16 units. An SOM fitted to the laser data using the parameters of uniform sampling can be seen in Fig. 3(e), superimposed on the data points. The curves fitted to the processed UAMs can be seen in Fig. 5.

The previously defined error criterion is employed for this method as well, using the curves fitted by the SOM instead of snakes. The results of the SOM method are given in the sixth, seventh, and eighth columns of Tables 4 and 5. With a few exceptions, SOM errors are larger than snake errors. The DM and VT methods demonstrate the best performance. In Fig. 5(f), it can also be observed that the SOM curve fitted to the map obtained by the ATM-org technique is

highly affected by the outlier points, unlike the snake curve fitted for that technique (Fig. 4(f)). Curves generated by fitting an SOM are not constrained by length or curvature as the snake curves are, thus they are more likely to fit to the outlier points. This results in larger error values in general.

7 Discussion

Looking at the error values in Tables 4 and 5 and observing Figs. 4 and 5, it can be concluded that DM, VT, and ATM-mod methods eliminate most of the artifacts in the ultrasonic data resulting from multiple and higher-order reflections, cross-talk, and erroneous measurements. The PM, MP, BU, and ATM-org methods cannot eliminate those artifacts as much, resulting in larger errors. This can be observed more clearly in Fig. 5.

In general, if a UAM processing technique cannot eliminate artifacts well, the resulting errors are larger. DM, VT, and ATM-mod are superior to the other techniques in eliminating artifacts, therefore they result in smaller errors. In

Table 6 Computation times for fitting snake curves and SOM for a given parameter set. Overall computation time for snake fitting is the sum of the computation times in the Distance map and Snake fitting columns

Method	Number of map points	Computation time (sec)			
		Distance map	Snake fitting	Overall	SOM
PM	697	5.6	69.0	75	44
VT	2634	20.3	74.6	95	168
DM	863	7.1	78.2	85	55
MP	4692	35.1	110.6	146	299
BU	2994	21.9	105.9	128	191
ATM-org	920	5.8	66.6	72	59
ATM-mod	788	5.0	78.9	84	50
TBF	387	3.5	57.1	61	25

fact, DM can be considered an improved version of VT, where directional processing of the map points is incorporated in the algorithm (Barshan 2007). The modified ATM is also quite accurate and eliminates the artifacts better than the original ATM. In the UAM, if there are arcs with no intersections, these are removed with the modified ATM but not with the original ATM. Generally speaking, the ATM technique creates accurate yet sparsely filled maps. ATM is found to require a denser UAM to begin with in order to produce a map with approximately the same number of points as the other techniques.

The PM technique reduces each arc to a single point mark in the middle of the arc. ATM-org places a more accurate point mark on arcs with transversal intersections (except those with two and none), reducing many of the arcs to single points also. It treats arcs with no intersections in the same way as PM. For this reason, the number of extracted map points from these two techniques is quite similar.

The errors with TBF are usually larger than ATM-org and ATM-mod errors. The TBF obtains the fewest number of map points among all the techniques compared and results in more gaps in the resulting map. This result is, however, expected because apart from the fact that fewer arcs are used at a given time to begin with (due to the sliding window), TBF eliminates arcs with no meaningful and accurate correspondence. In addition, planar wall locations found with this method are not very accurate. This is also observed in Wijk and Christensen (2000a)'s Figs. 15 and 16 as many outlying points extracted by the algorithm. A major advantage of TBF is that it is very fast and takes about the same time as the simplest PM method (Barshan 2007) because it does not divide the environment into grids but processes the information geometrically.

Among the eight approaches considered, DM produces relatively low errors and the associated computation time is small (Table 6). Considering that DM also has high range accuracy and is superior in eliminating the artifacts and outliers of the UAM, it can be considered as one of the best methods in terms of the overall performance.

In Barshan and Başkent (2000, 2001a, 2001b), VT and MP were investigated in detail based on simulations and experimental studies for different transducer configurations (linear, circular, random), different beamwidths (5° to 105°), different surface curvatures, roughness, distance, and different noise levels on time-of-flight measurements. The best results were obtained with a random configuration of transducers, followed by circular and linear ones. For both methods, the errors were shown to increase with increasing beamwidth, increasing surface distance, curvature, and roughness. Although such detailed studies for the other methods have not been performed, we expect similar results for the remaining techniques because varying these parameters primarily affects the quality of the information inherent in the ultrasonic arc map. This also leads us to expect that for a given choice of these parameters, the results of comparing the methods will not be altered significantly.

The active contour and SOM methods used for map representation differ in many aspects. Snake curves minimize an energy function that is the sum of their internal energy (basically length and curvature) and a potential function that is at minimum on the map points. As illustrated in our example, snake curves are more robust to outliers than SOMs (compare Figs. 4(f) and 5(f)). SOMs, on the other hand, are not constrained by length or curvature and try to adjust the curve to encompass all data points. For this reason, SOMs are more likely to fit to outlying map points and may not converge to the actual borders of the environment, resulting in larger errors in general. This is the case when larger values of k_λ and k_σ are used with the effect that the decay is slower. Experimenting with different k_λ and k_σ values, we have observed that decreasing $\lambda(n)$ and $\sigma(n)$ more rapidly causes the SOM to be more robust to the outlier points of the ultrasonic map. When smaller values of k_λ and k_σ are used, it is more likely that the SOM will fit to the topology of the data rather than the outliers and the resulting errors will be smaller. However, even in this case, active snake contours are still superior in terms of the resulting errors.

The evolution equations (10) and (11) are used to update the position of the snake. The parameters involved in these

equations control the final shape of the snake such as determining its length and curvature. Selecting the best parameters is less critical for the convergence of snakes than for SOMs. For snakes, there may be more than one parameter set that fit the given data points well. However, the convergence of snakes is more sensitive to the initialization of the curve than are SOMs, whereas the convergence of SOMs is sensitive to the order in which the map points are input to the neural network. In our implementation of the SOM, the points were input randomly.

The running times of active contours and SOMs are tabulated in Table 6 for each of the UAM processing techniques, together with the number of data points in the corresponding processed UAM. These running times are obtained using MATLAB on a computer with a 1.80 GHz dual core processor. It can be observed that the running times of both methods increase with increasing data size, however, the running time of the SOM is more dependent on data size. The computation time of the SOM increases roughly linearly with the number of data points, with an average processing time of 64 ms per map data point. In the active snake contour method, the data size mainly affects the time for calculating the distance map. On average, forming the distance map takes 7.6 ms per map data point. Once the distance map is formed, the running time of the snake fitting algorithm is mainly determined by the number of points comprising the snake curve. For our example, the average running time is about 80 sec. The overall average running time is about 93 sec. The slight increase in the snake fitting time for the BU and MP techniques is because the corresponding snake curves are longer, resulting in more points to process at each iteration (Fig. 4). Furthermore, the overall running time and accuracy of the snake fitting technique change depending on the resolution of the image that represents the map. More generally, the computation times for both methods depend on the selected parameters because these parameters control the convergence characteristics of the curves. Even though a fixed number of iterations is used in this study, parameters can be tuned for a specific application and the computation times can be reduced by using an application-specific number of iterations. However, regardless of a fixed or variable number of iterations, the results given in Table 6 can be viewed as an example of what to expect for processing times in an application in terms of the map size and the curve-fitting method used.

8 Conclusion

We have presented two approaches to compactly and efficiently represent the maps obtained by processing UAMs with different techniques. Representing the map points with snake curves or self-organizing maps makes it possible to

compare maps obtained with different techniques among themselves, as well as with an absolute reference. For this purpose, we have defined and employed an error criterion that can be used to compare two different sets of points based on the Euclidean distance measure. The two sets of points can be chosen as (i) two different sets of map points acquired with different mapping techniques or different sensing modalities (e.g., ultrasonic data and laser data), (ii) two sets of curve points (e.g., two snake curves) fitted to maps extracted by different mapping techniques or different sensing modalities, or (iii) a set of extracted map points and a set of curve points fitted to those points. Our purpose in this paper was to represent ultrasonic maps efficiently and to evaluate the performance of UAM processing techniques through the use of a demonstrative example. Among the eight UAM processing techniques considered, the directional maximum method can be considered as one of the best in terms of the overall performance.

Although both active contours and SOMs can be employed to fill the erroneous gaps in discrete point maps, active contours are superior in eliminating outliers in the data, resulting in smaller errors. Active contours are more capable of fitting accurately to environmental features with high curvature, such as edges and corners, as well as smoother features, such as planar walls, in typical indoor environments.

A physical interpretation of the parameters and guidelines on parameter selection are provided. The best-fitting parameter values are found by uniform sampling of the parameter space and by particle swarm optimization. For snake curve fitting, the error values are comparable for the two approaches, whereas for SOM, PSO consistently results in smaller errors. Another alternative is to use non-parametric snakes that utilize a kernel density estimation approach (Özertem and Erdoğan 2007). However, this formulation does not completely eliminate the parameter optimization procedure, as a number of parameters must be determined for both fixed- and variable-bandwidth kernels, especially for noisy images. Furthermore, an additional heuristic procedure needs to be used to assure convergence to concavities in the map.

The two methods are sufficiently general that they can also be applied to map data points acquired with other mapping techniques and sensing modalities. The results can be extended to 3-D data by fitting 3-D shapes. Another possible extension of this work would be the automatic determination of the appropriate number of curves or shapes to be fitted to a given set of extracted map points using clustering techniques. Determining whether the curves should be open or closed, or the shapes convex or concave, and initializing the multiple curve parameters are other challenging issues.

References

- Altun, K., & Barshan, B. (2008). Performance evaluation of ultrasonic arc map processing techniques by active snake contours. In H. Bruyninckx, L. Preucil, & M. Kulich (Eds.), *Springer tracts in advanced robotics (STAR) series: Vol. 44. European robotics symposium 2008* (pp. 185–194). Berlin: Springer.
- Barshan, B. (1999). Ultrasonic surface profile determination by spatial voting. *Electronics Letters*, 35(25), 2232–2234.
- Barshan, B. (2007). Directional processing of ultrasonic arc maps and its comparison with existing techniques. *International Journal of Robotics Research*, 26(8), 797–820.
- Barshan, B. (2008). Objective error criterion for evaluation of mapping accuracy based on sensor time-of-flight measurements. *Sensors*, 8(12), 8248–8261.
- Barshan, B., & Başkent, D. (2000). Comparison of two methods of surface profile extraction from multiple ultrasonic range measurements. *Measurement Science and Technology*, 11(6), 833–844.
- Barshan, B., & Başkent, D. (2001a). Morphological surface profile extraction with multiple range sensors. *Pattern Recognition*, 34(7), 1459–1467.
- Barshan, B., & Başkent, D. (2001b). Map building from range data using mathematical morphology. In P. P. Smith (Ed.), *World scientific series in robotics and intelligent systems: Vol. 26. Active sensors for local planning in mobile robotics* (pp. 111–135). New Jersey: World Scientific (Chapter 7).
- Başkent, D., & Barshan, B. (1999). Surface profile determination from multiple sonar data using morphological processing. *International Journal of Robotics Research*, 18(8), 788–808.
- Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3), 344–371.
- Choset, H., Nagatani, K., & Lazar, N. (2003). The arc-transversal median algorithm: a geometric approach to increasing ultrasonic sensor azimuth accuracy. *IEEE Transactions on Robotics and Automation*, 19(3), 513–522.
- Cohen, L. D. (1991). On active contour models and balloons. *Computer Vision, Graphics, and Image Processing (CVGIP): Image Understanding*, 53(2), 211–218.
- Cohen, L. D., & Cohen, I. (1993). Finite element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11), 1131–1147.
- Crowley, J. L. (1985). Navigation for an intelligent mobile robot. *IEEE Transactions on Robotics and Automation*, RA-1(1), 31–41.
- Drumheller, M. (1987). Mobile robot localization using sonar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(2), 325–332.
- Elfes, A. (1987). Sonar based real-world mapping and navigation. *IEEE Transactions on Robotics and Automation*, RA-3(3), 249–265.
- Gex, W., & Campbell, N. (1987). Local free space mapping and path guidance. In *Proceedings of IEEE international conference on robotics and automation* (pp. 424–431).
- Jacob, M., Blu, T., & Unser, M. (2004). Efficient energies and algorithms for parametric snakes. *IEEE Transactions on Image Processing*, 13(9), 1231–1244.
- Kass, M., Witkin, A., & Terzopoulos, D. (1987). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4), 321–331.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks* (Vol. 4, pp. 1942–1948).
- Knopf, G. K., & Sangole, A. (2004). Interpolating scattered data using 2D self-organizing feature maps. *Graphical Models*, 66(1), 50–69.
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1), 59–69.
- Kuc, R., & Siegel, M. W. (1987). Physically-based simulation model for acoustic sensor robot navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(6), 766–778.
- Kumar, G. S., Kalra, P. K., & Dhande, S. G. (2004). Curve and surface reconstruction from points: an approach based on self-organizing maps. *Applied Soft Computing*, 5(1), 55–66.
- Kurz, A. (1996). Constructing maps for mobile robot navigation based on ultrasonic range data. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 26(2), 233–242.
- Leonard, J. J., & Durrant-Whyte, H. F. (1992). *Directed sonar sensing for mobile robot navigation*. Boston: Kluwer Academic.
- Liang, J., McInerney, T., & Terzopoulos, D. (2006). United snakes. *Medical Image Analysis*, 10(2), 215–233.
- Menet, S., Saint-Marc, P., & Medioni, G. (1990). Active contour models: overview, implementation and applications. In *Proceedings of the IEEE international conference on systems, man and cybernetics* (pp. 194–199).
- Özertem, U., & Erdoğan, D. (2007). Nonparametric snakes. *IEEE Transactions on Image Processing*, 16(9), 2361–2368.
- Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization—an overview. *Swarm Intelligence*, 1(1), 33–57.
- Ribas, D., Ridao, P., Neira, J., & Tardós, J. D. (2007). Line extraction from mechanically scanned imaging sonar. In *Lecture notes in computer science: Vol. 4477. Pattern recognition and image analysis* (pp. 322–329). Berlin: Springer.
- Rosenfeld, A., & Pfaltz, J. L. (1968). Distance functions on digital pictures. *Pattern Recognition*, 1(1), 33–61.
- Tardós, J. D., Neira, J., Newman, P. M., & Leonard, J. J. (2002). Robust mapping and localization in indoor environments using sonar data. *International Journal of Robotics Research*, 21(4), 311–330.
- Wijk, O., & Christensen, H. I. (2000a). Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data. *Robotics and Autonomous Systems*, 31, 31–42.
- Wijk, O., & Christensen, H. I. (2000b). Triangulation-based fusion of sonar data with application in robot pose tracking. *IEEE Transactions on Robotics and Automation*, 16(6), 740–752.
- Xu, C., & Prince, J. L. (1998). Snakes, shapes and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3), 359–369.
- Yamada, S. (2004). Recognizing environments from action sequences using self-organizing maps. *Applied Soft Computing*, 4(1), 35–47.



Kerem Altun received his BS and MS degrees in mechanical engineering from Middle East Technical University (METU) in Ankara, Turkey, in 1999 and 2002, respectively. He is currently working towards his PhD degree in the Department of Electrical and Electronics Engineering at Bilkent University, Ankara. He received a PhD scholarship from The Scientific and Technological Research Council of Turkey in 2006, and the Professor Güven Aksu award from METU's Mechanical Engineering Department in 1998 for outstanding performance in an undergraduate course. His research interests include intelligent sensing, sensor signal processing, ultrasonic and inertial sensing, pattern recognition, system dynamics and control, and sensor data fusion.



Billur Barshan received BS degrees in both electrical engineering and physics from Boğaziçi University in Istanbul, Turkey in 1986 and her MS and PhD degrees in electrical engineering from Yale University in the USA in 1988 and 1991, respectively. Dr. Barshan was a research assistant at Yale from 1987 to 1991 and a postdoctoral researcher at the Robotics Research Group at the University of Oxford in the UK from 1991 to 1993. In 1993, she joined the faculty of Bilkent University in Ankara, Turkey, where she

is a professor in the Department of Electrical and Electronics Engineering. Dr. Barshan is also the founder of the Robotics and Sensing Laboratory in the same department. She is the recipient of the 1994 Nakamura Prize for the most outstanding paper at the 1993 IEEE/RSJ Intelligent Robots and Systems International Conference, the 1998 TÜBİTAK Young Investigator Award, and the 1999 Mustafa N. Parlar Foundation Research Award. Dr. Barshan's current research interests include sensor-based robotics, intelligent sensing, ultrasonic, optical, and inertial sensing, sensor signal processing, multi-sensor data fusion, and human activity recognition and classification.