BAYESIAN RECONSTRUCTION OF SEAFLOOR SHAPE FROM SIDE-SCAN SONAR MEASUREMENTS USING A MARKOV RANDOM FIELD

Philipp Woock [a], Alexey Pak [b]

[a] Karlsruhe Institute of Technology (KIT), Adenauerring 4, 76131 Karlsruhe, Germany
[b] Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB), Fraunhoferstr. 1, 76131 Karlsruhe, Germany

E-mail: philipp.woock@kit.edu

Abstract: *To explore the seafloor, a side-scan sonar emits a directed acoustic signal and then records the returning (reflected) signal intensity as a function of time. The inversion of that process is not unique: multiple shapes may lead to identical measured responses.*
In this work, we suggest a Bayesian approach to reconstructing the 3D shape of the seafloor from multiple sonar measurements, inspired by the state-of-the-art methods of inverse raytracing that originated in computer vision. The space near the bottom is modelled as a grid of voxels, whose occupancies are represented by random binary variables. Any assignment of occupancies corresponds to some seafloor shape. A global multi-component energy potential describes how well the resulting surface agrees with the sonar data and with the a priori assumptions. Minimization of energy is equivalent to finding the maximum a posteriori (MAP) assignment to this Markov random field (MRF) and is done using the iterated belief propagation (BP) algorithm.
The critical step in this method is to compute messages from "factors" representing the sonar beams to voxels. Naïvely, its complexity scales exponentially with the number of voxels traversed by a beam. Unlike inverse raytracing, where a pixel value constrains voxels only along a single view ray, a sonar beam involves voxels within a relatively wide cone. Employing dynamic programming techniques and space-filling curves, we were able to develop a practical approximate solution to this problem.
The algorithm is not restricted to side-scan sonar reconstruction and could be applied to medical ultrasound or ultra wide-band (UWB) radar imaging.

Keywords: *Bayesian reconstruction, surface estimation, side-scan sonar, inverse raytracing, Markov random field, belief propagation*

1. INTRODUCTION

Reconstructing a seafloor 3D shape from side-scan sonar data is difficult since the recorded signal carries only indirect information about the distance from the sensor to bottom. Beam propagation in the water and reflection from the bottom can be modelled relatively accurately, see e.g., [1], [2], and [3] but the inverse problem is to our knowledge not yet solved. A similar problem in computer vision, namely the inverse raytracing, has recently seen an efficient solution [4], [5].

The latter works by Liu and Cooper focus on reconstructing a 3D scene from multiple camera views. To that end, they introduce a grid of voxels spanning a volume of interest. Each voxel corresponds to a binary random variable that describes its occupancy. The relations between voxels are encoded in terms of “factors’’ linked to those variables. The resulting structure known as a Markov random field (MRF) can be represented as a bi-partite graph. In essence, an MRF describes the joint probability of a simultaneous assignment to all the occupancy variables. A huge monolithic probability function is replaced with a long product of functions each depending only on a few variables, enabling efficient inference methods.

The RayMRF, as Liu and Cooper call it, contains three types of factors. A unit factor is linked to a single variable and describes the a priori occupancy probability for that voxel. Pair factors connect two adjacent voxels and encode, e.g., continuity or smoothness assumptions. The most interesting ray factors are linked to all voxels pierced by a camera view ray corresponding to a single pixel and describe the agreement between the observed pixel value and the given assignment of voxel values.

One practical way to “calibrate’’ such an MRF, or find the maximum a posteriori (MAP) assignment of voxel variables (or, equivalently, the scene shape), is known as loopy belief propagation (LBP). Each factor iteratively exchanges “messages’’ with all the linked variables. The messages express the “beliefs’’ about the occupancy of each voxel based on prior data and the messages from other variables. To compute a message from a factor, one has to consider all assignments to the linked variables and thus solve a local optimization problem. This is not a big problem for, e.g., pair factors where one only has to consider four assignments (for a binary variable), but a ray factor in practice can be linked to hundreds of voxels. The number of assignments, growing exponentially, prohibits a naïve brute-force solution.

RayMRF presents a novel method to compute ray factor messages. It exploits the fact that the ray “energy’’, or the assignment likelihood, depends only on the first occupied voxel on a ray. With dynamic programming, the complexity of this step is reduced to linear in the number of linked voxels. In this report we generalize this approach to apply to the (much more challenging) problem of the sonar data inversion.

2. MRF APPLICATION TO SIDE-SCAN SONAR DATA

2.1. Geometry and observables of side-scan sonar beams

Unlike thin camera view rays, a sonar beam has typically a wide and flat power distribution. If by analogy with RayMRF we devise an MRF-based model for sonar-based reconstruction (a “BeamMRF’’) with unit, pair, and beam factors, each of the latter will be connected to a large number of voxel variables (larger than that for a narrow ray). Next, instead of a single RGB value, the observed quantity for a beam is a function $I(t)$ that represents the reflected acoustic energy that has reached the receiver at the time moment t. In order to relate the time to geometry, we assume isovelocity sound propagation, i.e., a constant and isotropic speed of sound in the water. This translates to straight conical sonar beams. In order to describe the relation between the function $I(t)$ and the bottom shape it is necessary to discretize the beam geometry. First, we split the wide beam into infinitesimally narrow cones. Second, we sub-divide each cone into individual slices as shown in Figure 1.

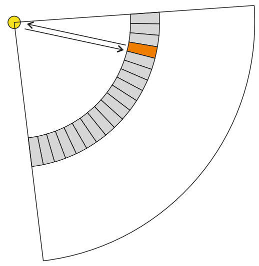

Figure 1: Discretized sonar beam geometry for coinciding source and receiver positions.

The recorded infinitesimal energy $\Delta R(t_i)$ at some time slice t_i is a sum of contributions ΔR_s from ray cones (indexed by s) in some set $S(t_i)$ which encompasses all surface elements that can be reached by the signal (i.e., which are not shaded) and located at a distance determined by the return travel time t_i (here we neglect multiple reflections):

$$\Delta R(t_i) = \sum_{s \in S(t_i)} \Delta R_s \, .$$

This model can be further described using a two-dimensional grid (Figure 2). Given D directions inside the beam and T discernible time slices, the sonar response can be computed based on the occupancies of $D \cdot T$ voxels[1]:

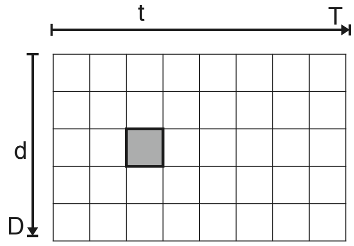

Figure 2: Logical voxel structure of a sonar beam

Any assignment of variables on this grid corresponds to some bottom shape. Let us denote the first occupied voxel in direction d by $t^*(d)$. That means that all voxels before $t^*(d)$ (i.e., those with $t < t^*(d)$) on the d-th ray are empty while voxels after $t^*(d)$ may be either empty or occupied. If we assume that a voxel at (t, d) reflects back an amount of energy given by $R(t, d)$ (depending on, e.g., surface normal, its material, and the sonar characteristics), we may compute the energy corresponding to the depth/time slice t as

$$\Delta \mathrm{R}(t) = \sum_{d=1}^{D} R(t, d) \cdot \delta_{t^*(d)}^{t},$$

where δ is the Kronecker delta symbol. Denoting the voxel occupancy variables as $o(t, d) \in \{0,1\}$, we may further define $t^*(d)$ as follows:

[1] Here we skip the relation between the beam and the world voxels which in the RayMRF model is known as interpolation. It suffices here to say that this problem is solved using the common raytracing methods.

$$t^*(d) = \begin{cases} 1, \text{if } o(1,d) = 1 \\ 2, \text{if } o(1,d) = 0 \ \wedge o(2,d) = 1 \\ 3, \text{if } o(1,d) = 0 \ \wedge o(2,d) = 0 \ \wedge o(3,d) = 1 \\ \quad \text{etc.} \end{cases}$$

For each beam B_i, the agreement between the observation and the model is given by a quadratic functional ("beam energy", not to be confused with the acoustic energy!):

$$E_{B_i} = \sum_{t=1}^{T} \left(\Delta R^{\text{estim}}(t) - \Delta \mathrm{R}^{\text{observed}}(t)\right)^2 \cdot \varrho(t),$$

with $\varrho(t)$ being some time-dependent weight that accounts, e.g., for lower signal-to-noise ratio for later readings. The full MRF energy (which can be thought of as the negative logarithm of the joint probability function) contains then the prior terms E_u and E_p (unit and pair energies), and the sum of beam energies from the set of observations $\mathcal{B}$. The goal of the reconstruction is to find an assignment O to voxel occupancies that minimizes the total energy functional:

$$O_{\text{opt}} = argmin\left(E_u + E_p + \sum_{B_i \in \mathcal{B}} E_{B_i}\right).$$

2.2. Messages from beam factors to voxel variables

The details and the justification of the LBP method can be found elsewhere [6]. We also assume that the unit and pair factor updates are performed as in the RayMRF model. Here we focus only on the non-trivial problem of computing the messages from a beam factor to the linked variables. Given the energy functional E_{B_i}, we formally define the needed differential message from the beam factor to voxel at (t,d) as

$$w(t,d) = M_{f\to td}(o(t,d) = 1) - M_{f\to td}(o(t,d) = 0),$$

$$M_{f\to td}(o(t,d) = 1) = \min_{\{o(t',d') | o(t,d) = 1\}} \left(E_{B_i}(\{o(t',d')\}) + \sum_{(t',d')\neq(t,d)} M_{td\to f}\big(o(t,d)\big)\right),$$

$$M_{f\to td}(o(t,d) = 0) = \min_{\{o(t',d') | o(t,d) = 0\}} \left(E_{B_i}(\{o(t',d')\}) + \sum_{(t',d')\neq(t,d)} M_{td\to f}\big(o(t,d)\big)\right).$$

The incoming messages $M_{td\to f}\big(o(t,d)\big)$ are known, and the minimum in each case is taken over all assignments to variables other than (t,d). The above formula requires thus roughly $\mathcal{O}(2^{T\cdot D})$ steps, which is too expensive for any reasonable T and D.

2.3. Exact beam front-based solution

Following Liu and Cooper, we notice that the beam energy in fact only depends on $t^*(d)$. The summations above can therefore be split into three regions (Figure 3). The voxels at $(t^*(d), d)$ belong to region A, those at $(t,d), t > t^*(d)$ to region B, and those at $(t,d), t < t^*(d)$ to region C. Note that in region A all voxels are necessarily occupied, and in region C necessarily empty.

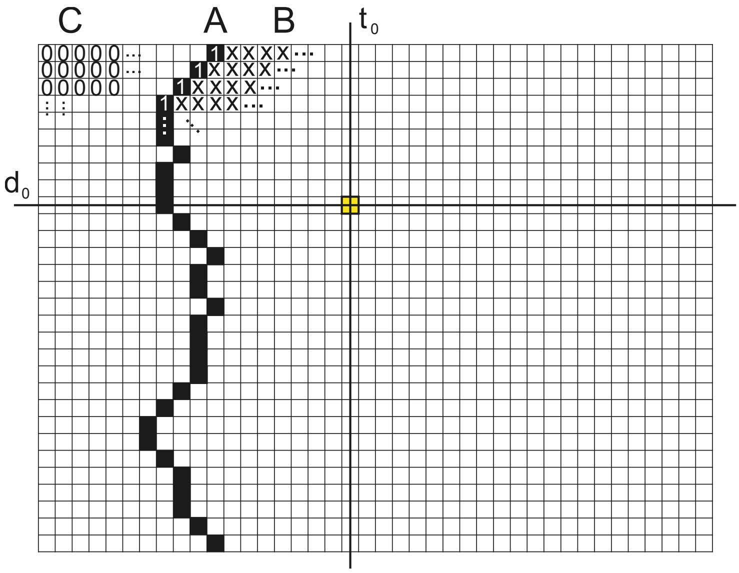

Figure 3: Beam factor occupancy profile

Simplifying terms in each region, we may compute the messages with a minimum taken over all assignments to $t^*(d) = (a_1, \dots, a_D) = \vec{a}$ instead of the binary grid:

$$M_{f\to td}(o(t,d)=1) = \min_{\{\vec{a}|a_d \le t\}} \left(E_{B_i}(\{\vec{a}\}) + \sum_{d'=1}^{D} m(a_{d'}, d') + \sum_{d'=1}^{D} \sum_{t'=a_{d'+1}}^{T} \min\big(0, m(t', d')\big) - CT_1 - CT_2 \right) + \xi$$

$$CT_1 \coloneqq \text{ if } t > a_d \text{: } \min\big(0, m(t,d)\big), \text{ else } 0$$

$$CT_2 \coloneqq \text{ if } t = a_d \text{: } m(t,d), \text{ else } 0.$$

Since an unoccupied voxel at (t,d) can only belong to regions C or B, we have:

$$M_{f\to td}(o(t,d)=0) = \min_{\{\vec{a}|a_d \ne t\}} \left(E_{B_i}(\{\vec{a}\}) + \sum_{d'=1}^{D} m(a_{d'}, d') + \sum_{d'=1}^{D} \sum_{t'=a_{d'+1}}^{T} \min\big(0, m(t', d')\big) - CT_3 \right) + \xi$$

$$CT_3 \text{: if } t > a_d \text{: } \min\big(0, m(t,d)\big), \text{ else } 0$$

The constant ξ cancels in the final formula for $w(t,d)$ and is thus not important. This representation is exact and has complexity $\mathcal{O}(D^T)$. The double sums under minima would naïvely require more steps but the iteration over $t^*(d)$ can be re-organized so that each sum can be incrementally updated in constant time on each step. The same applies to beam energies of each configuration. Moreover, messages from a beam factor to all variables may be computed at one pass. This, again, is possible due to dynamic programming.

2.4. Approximate front-based solution

As stated above, the iteration over all front shapes takes $\mathcal{O}(D^T)$ steps. This may still be too expensive for wide beams. We further notice then that most of the shapes are highly improbable, and the corresponding fronts do not contribute to (most) factor messages. We thus limit ourselves with exploring only the front shapes near some "most plausible" variants. Technically, an assignment $\vec{a}$ to the front shape $t^*(d)$ is equivalent to an integer number with D digits in base T. Given some starting number, we may thus simply consider a few values in its vicinity such that the number of steps is determined by some "search depth" parameter. However, as discussed above, numbers base T are inconvenient for dynamic programming: on each step, many entries in $\vec{a}$ may change by more than a single unit. We thus adopt a parameterisation of the D-dimensional search volume with a space-filling Hilbert curve [7]. By following this curve, we are guaranteed that each step changes a single entry by plus or minus one.

The starting front shapes ("seeds") can be obtained from additional sensors or with any heuristic method such as that of [8]. Finally, an efficient message calculation scheme for a beam factor is as follows. First, we select some starting front assignment ("seed") $\vec{a}$ and compute the value

$$X(\vec{a}) = E_{b_i}(\vec{a}) + \sum_{d'=1}^{D} m(a_{d'}, d') + \sum_{d'=1}^{D} \sum_{t'=a_{d'}+1}^{T} \min\left(0, m(t', d')\right).$$

We also initialize all messages to plus infinity. The following assignments $\vec{a}$ are selected according to the Hilbert curve parameterization. Therefore, $X(\vec{a})$ and the messages can be updated at each step in constant time as

$$M_{f\to td}(o(t,d) = 0) = \min_{\{\vec{a}\}} \begin{cases} X(\vec{a}), \text{if } a_d > t \\ \infty, \text{if } a_d = t \\ X(\vec{a}) - \min(0, m(t,d)), \text{if } a_d < t \end{cases}$$

$$M_{f\to td}(o(t,d) = 1) = \min_{\{\vec{a}\}} \begin{cases} \infty, \text{if } a_d > t \\ X(\vec{a}) - m(t,d), \text{if } a_d = t \\ X(\vec{a}) - \min(0, m(t,d)), \text{if } a_d < t \end{cases}$$

For an efficient implementation, caching of minimum values for each cell (t, d) is advised. The complexity of this method depends only on the "search depth" near the seed value and must be determined based on the desired inference accuracy. The remaining infrastructure of the reconstruction framework can be directly inherited from the RayMRF model. We expect that the resulting algorithm will for the first time deliver accurate Bayesian shape estimations based on sonar data.

3. CONCLUSION

In this contribution, we suggest a novel method to reconstruct seafloor shape from side-scan sonar data that is based on the RayMRF model borrowed from the domain of computer vision. For the introduced beam factors, we discuss the core inference step and demonstrate a practically feasible approximate solution that uses dynamic programming and space-filling curves to drastically reduce the message update complexity. In the future, we plan to present the applications of the method to synthetic and real sonar data and quantify its accuracy.

The suggested method does not rely of heuristic treatment of the side-scan sonar data but honors the physical origins of the signal. Employing that method, AUVs equipped only with classical side-scan sonars might reconstruct the bottom surface. Additionally, AUVs equipped with sensors that produce distance measurements (like multi-beam echo-sounders (MBES) or interferometric sonars) can use those measurements as prior knowledge and refine their output with the (typically higher-resolution) data from the imaging side-scan sonar. As a means for a more exact environment mapping, the method should also facilitate the application of simultaneous localization and mapping (SLAM) methods for AUVs to improve the navigation accuracy.

REFERENCES

[1] J. M. Bell, „A Model for the Simulation of Sidescan Sonar,“ Heriot-Watt University, Department of Computing and Electrical Engineering, Edinburgh, 1995.

[2] J. Groen, „Adaptive motion compensation in sonar array processing,“ TU Delft, PhD thesis, 2006.

[3] E. Coiras und J. Groen, „Simulation and 3D Reconstruction of Side-looking Sonar Images,“ IN-TECH, Vienna, Austria, 2009.

[4] S. Liu, Statistical Inverse Ray Tracing for Image-Based 3-d Modeling, Providence: Brown University, 2011.

[5] S. Liu and D. B. Cooper, "Ray Markov Random Fields for image-based 3D modeling: Model and efficient inference," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[6] D. Koller and N. Friedman, Probabilistic Graphical Models - Principles and Techniques, Cambridge, Massachusetts: The MIT Press, 2009.

[7] J. Skilling, „Programming the Hilbert curve,“ Jackson Hole, Wyoming, 2004.

[8] P. Woock und J. Beyerer, „Seafloor shape estimation by raytraced kernels,“ Taipei, Taiwan, 2014.