Multi-UUV Perimeter Surveillance

Mathieu Kemp, Andrea L. Bertozzi, and Daniel Marthaler

***Abstract*—We develop two algorithms for the surveillance of underwater perimeters by a group of unmanned vehicles, and compare their performance.**

***Index Terms*—multi-vehicle coordination, underwater vehicles, robustness, low-bandwidth.**

I. Introduction

We consider the general problem of finding and patrolling an underwater perimeter with N UUVs that use a scalar sensor. We define a perimeter as a curve of constant concentration, i.e. a curve $(X(s),Y(s))$ that satisfies $C(X(s),Y(s))=C_o$ where $C(X,Y)$ is a two-dimensional scalar field, s is the curve parameter, and C_o is the perimeter concentration. Starting from an initial configuration, a team of N UUVs must find the perimeter and maintain a lock on it. This problem is of practical importance for thin layer monitoring [1], harmful algae bloom monitoring [2], tactical oceanography [3]-[6], harbor protection [7], and possibly many other problems.

Underwater perimeter surveillance is challenging. First, the UUVs can only use a scalar sensor. Second, only low throughput (acoustic) or intermittent and asynchronous communication (surface RF) are available. Third, some of the UUVs are expected to die during the mission. A successful algorithm must therefore able to function given these constraints.

We develop two algorithms. The first uses an image segmentation technique called snakes. The second treats the vehicles as a gas of particles affecting each other's speed. We examine their respective stability, convergence, and robustness, and compare them. Of the two, we find that the UUV-gas algorithm works best because it does not rely on a gradient and because it loosely couples the vehicles.

Manuscript received July 9, 2004. This work was supported by the Army Research Office under Grants DAAD19-02-1-0055 and 19-03-C-0075, and by ONR under Grant N000140410054. We thank Dan Stilwell for his comments.

M. Kemp is with Nekton Research, Durham NC 27707 USA (phone: 919-405-3993; fax: 919-405-3994; e-mail: fruitbat@ nektonresearch.com).

A. L. Bertozzi is with the Mathematics Department, UCLA, Los Angeles CA 90095-1555 USA (e-mail: bertozzi@math.ucla.edu).

D. Marthaler is with the Mathematics Department, UCLA, Los Angeles CA 90095-1555 USA (e-mail: daniel@math.duke.edu).

II. Snake Algorithm

A. Formulation

Snakes were originally developed for image segmentation where the goal is to find a structure inside a noisy 2D image. A snake is a curve that, when placed near the structure, quickly wraps itself around it.

The core of the snake algorithm is a partial differential equation that controls the shape of the curve:

$$\begin{aligned} \partial_t Z &= \alpha \partial_{ss}^2 Z - Q(Z) \\ Q &= \partial_x P + i \partial_y P \end{aligned} \tag{1}$$

where $Z(s,t) = X(s,t) + iY(s,t)$ is the curve, t is the evolution variable, $s \in [0, 2\pi]$ is the curve parameter, α is called elasticity parameter, and $P \equiv (C - C_o)^2$ [8]-[9]. The first term on the right-hand side contracts the curve and the second ties it to the perimeter.

In practice, the snake is not a continuous curve but an array of N points approximating a curve. Its temporal evolution is also done in steps rather than continuously. The transition from the continuous to the discrete case is done with finite differencing:

$$Z(t + \Delta t) = Z(t) + \Delta t \left(\alpha D Z(t) - Q\left(Z\left(t\right)\right)\right) \tag{2}$$

where Z is the N-dimensional vector whose component Z_n is the position of point n, $Q_n \equiv Q(Z_n)$, and D is the second difference operator with periodic boundary conditions:

$$D = \left(\frac{N}{2\pi}\right)^2 \begin{bmatrix} -2 & 1 & 0 & \cdots & 1 \\ 1 & -2 & 1 & 0 & \cdots \\ 0 & 1 & -2 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 1 & -2 \end{bmatrix}. \tag{3}$$

where Δt is the time step.

In the image segmentation problem, each of the N points is a pixel-like element that moves in an image. Nothing limits the points to be pixel-like objects however. In particular, we could treat each of them as the location of a vehicle moving in a concentration field. $Z_n(t)$ would then be the location of vehicle n at time t, $Z_n(t + \Delta t)$ its next waypoint, Q_n the data measured by n, and the (DZ) term would be computed by each vehicle using its own position measurements and data received form the other vehicles.

Two examples are shown in Figure 1. The first is a circular perimeter (derived form a Gaussian concentration field) patrolled by 5 UUVs. Under the combined effect of Q and curve elasticity, the vehicles converge to the perimeter and wrap themselves around it. The second example is a "figure 8" perimeter patrolled by 15 UUVs. As in the first example, the vehicles are able to successfully find the perimeter .

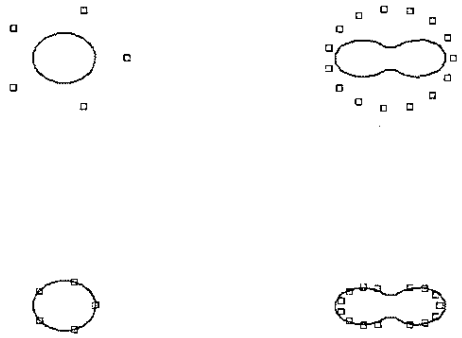

Figure 1. Time lapse sequence of simple snake dynamics. Left column: Gaussian field monitored by 5 vehicles. Right column: "figure 8" pattern monitored by 15 vehicles. The elasticity parameter is 0.05.

The snake algorithm is a decentralized algorithm. Each vehicle uses data from its sensors and data from the other vehicles to determine where to go next but no guidance is received from a central controller. The algorithm is therefore naturally immune to central controller failures.

In vehicle language, equation (2) is a guidance law. It specifies where each vehicle should go next. The guidance law of vehicle n has two terms, one that depends on the concentration gradient (at the vehicle's current location), and one that depends on vehicle n's current location and the current locations of its two neighbors. (We note that equation (2) presupposes synchronous vehicle updates; in [10] we show how asynchronous updates can be handled.) Different guidance laws can be used besides (2). Implicit temporal discretization of equation (1) gives:

$$Z(t+\Delta t) = Z(t) + \Delta t\left(\alpha DZ(t+\Delta t) - Q\left(Z\left(t+\Delta t\right)\right)\right) \quad (4)$$

and semi-implicit discretization gives:

$$Z(t+\Delta t) = Z(t) + \Delta t\left(\alpha DZ(t+\Delta t) - Q\left(Z\left(t\right)\right)\right). \quad (5)$$

The guidance law can also be augmented with additional terms. The term $i\omega\left(\partial_x C + i\partial_y C\right)$, where ω is a constant, adds patrolling, i.e. vehicle rotation around the perimeter [9]-[10]. A two-body repulsive term can also be added to create curve inflation.

B. Stability

To examine the stability of the algorithm, we consider the simpler field $C = C_{-} - ZZ^{*}/2$ where C_{max} is an arbitrary constant, and the circular perimeter $Z(s) = \mathrm{R}e^{is}$ (corresponding to $C_{\cdot} = C_{-} - R^{2}/2$).

The steady-state solutions of the equation

$$\partial_t Z = \alpha\partial_{ss}^2 Z + \left(R^2 - ZZ^{*}\right)Z \quad (6)$$

are discussed in [11]. The solution that matches the perimeter is $Z(s) = Ae^{is}$ where $A^2 = R^2 - \alpha$. Linearization about this solution gives

$$\partial_t z = \left(\alpha\partial_{ss}^2 + \alpha - A^2\right)z - A^2 e^{2is} z^{*} \quad (7)$$

where z is the perturbation. The eigenvalues are:

$$\lambda_k = -\left[\alpha k^2 + A^2 \pm \sqrt{4\alpha^2 k^2 + A^4}\right]. \quad (8)$$

The condition for stability is $\alpha < 2/5\,R^2$. There is therefore a critical value of the elasticity above which the algorithm is unstable.

Analysis of the discrete-space discrete-time problem proceeds along similar lines:

$$\alpha < KR^2$$

$$K = \frac{1}{\left(\frac{N}{2\pi}\right)^2\left[3-2\cos\left(\frac{2\pi}{N}\right)-\cos\left(\frac{4\pi}{N}\right)\Big/\cos\left(\frac{2\pi}{N}\right)\right]}. \quad (9)$$

Stability is again conditioned to the amount of elasticity. Additionally however, the discrete-time nature of the algorithms creates an extra stability condition on the time-step:

$$\Delta t < \frac{1}{R^2 + 2\alpha\left(N/2\pi\right)^2\left(2\cos(2\pi/N)-1\right)}. \quad (10)$$

For the implicit scheme, we find the same stability condition $\alpha < 2/5\,R^2$ but find no time-step constraints.

III. UUV GAS

A. Formulation

The snake algorithm has a few drawbacks: it is conditionally stable and it relies on knowledge of the gradient. We have shown in the previous section that stability can be restored by reducing the elasticity parameter (although the critical elasticity parameter decreases as more vehicles are lost). Loss of stability is directly attributable to the tight coordination imposed by the algorithm. The dependence on the gradient is more problematic. Real ocean signals are rarely smooth and it is not clear how to construct a reliable gradient estimator from real data. Noise and limited resolution of real sensors will dominate the gradient estimator in regions of low gradient. Finally, some sensors can only detect the presence or absence of a signal – an example is an alarm detector. There are good reasons to design an alternative algorithm that uses looser coordination and that does not require a gradient.

B. Removal of dependence on gradient

We first introduce the single-vehicle case before discussing the multi-vehicle algorithm. The method is shown in Figure 2: whenever the UUV is inside the perimeter, it turns clockwise and whenever it is outside, it turns counterclockwise:

$$\frac{d\theta}{dt} = \begin{Bmatrix} +\omega & inside \\ -\omega & outside \end{Bmatrix} \tag{11}$$

where ω is a constant and θ is the heading. This simple procedure allows the UUV to, using only knowledge of whether it is inside or outside the perimeter, patrol the perimeter.

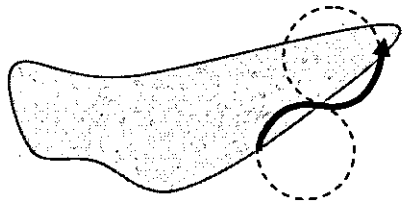

Figure 2. Description of gradient-free method.

Figure 3 shows three examples. The first is a perimeter of constant curvature, the second is a perimeter of variable and sign-changing curvature, and the third is a "city-block" perimeter. In all cases, the UUV is able to patrol the perimeter.

1) Stability

The algorithm creates a necklace-like sequence of connected circular segments of radius r. While on a segment, the UUV is at most within $2r$ of the perimeter. Because this always holds, the algorithm is stable.

2) Convergence

We assume that the vehicle is given a-priori knowledge of a point X_p inside the perimeter. To achieve convergence, we implement a two-state transition machine. While in state 1 (*searching*), the vehicle moves towards X_p if it is outside the perimeter and away from it if it is inside. Transition to state 2 (*patrolling*) occurs when a crossing is detected. Transition back to *searching* is initiated if the time since the last crossing exceeds a timeout (set to a few times the expected delay between crossings). Starting from any point, this state machine guarantees that the UUV will find the perimeter. This shows that the algorithm is convergent.

3) Coverage

Stability and convergence do not guarantee that the entire perimeter will be searched (Figure 4). We now show that coverage is complete provided that the perimeter is smooth.

Each UUV trajectory intersects the perimeter at $s_1, s_2, ..., s_n, ...$ To show coverage, we need to show that (i) $s_{n+1} > s_n \forall n$, and (ii) the entire perimeter is contained within the necklace.

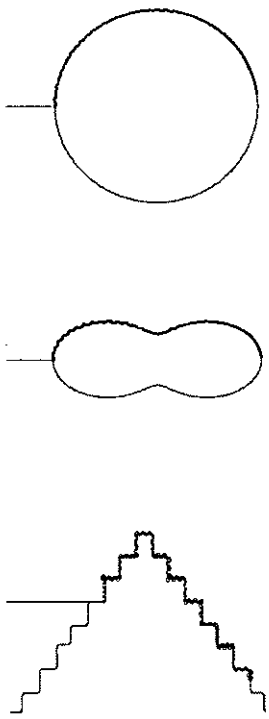

Figure 3. Gradient-free algorithm on three perimeters. The UUV arrives from the left in the *searching* state and switches to the *patrolling* state when it crosses the perimeter. While patrolling, the UUV trajectory is a series of circular segments around the perimeter. The gray line is the perimeter and the black line is the UUV trajectory. Top: circular perimeter. Middle: "figure 8" perimeter. Bottom: city block perimeter.

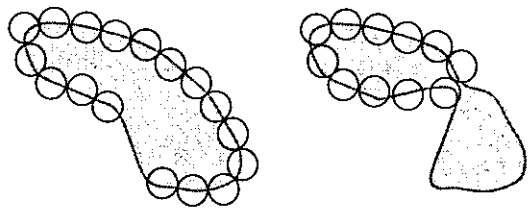

Figure 4. Necklace. Left: complete coverage. Right: incomplete coverage.

We consider a smooth perimeter with minimum radius of curvature R_c and a trajectory with $r<R_c$. The circle that completes the circular segment at A (Figure 5) contains the perimeter segment from A to B. To prevent this circle from containing other sections of the perimeter, we impose $r < R^* \equiv \min\left(D_{\min}/2, R_c\right)$ where $D_{\min}$ is the minimum distance between any two perimeter points C and D subject to $\left|s(C) - s(D)\right| > 2\pi R_c$.

Within this restriction, the trajectory crosses at A and exits at B. The vector that joins A to B has a positive projection on the vector tangent to the perimeter at A. This implies that s increases from A to B. Since the argument can be repeated at

every crossing point, $s_{n+1} > s_n \forall n$.

Because a unique perimeter segment is contained within each circle and because the parametrization is continuous, all perimeter points between s_n and s_{n+1} are contained within circle n. Because circles n and $n+1$ touch at s_{n+1}, there are no gaps in s. Because the increase of s within a circle has a lower bound ($s_{n+1} - s_n > \pi r/2$) and since nothing limits the length of the sequence, the entire perimeter is therefore contained in the necklace. Coverage is therefore complete.

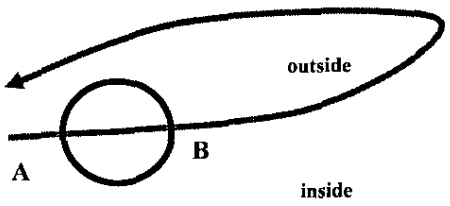

Figure 5. The UUV covers the perimeter without skipping sections if the radius of the circle is small enough.

C. Coordination

If N UUVs are launched, each running the gradient-free algorithm, they will find the perimeter and patrol it but will not spread themselves evenly. UUV-gas coordination spreads the vehicles by changing their speed as a function to their proximity (in contrast to changing their velocity, as was done in [9]):

$$\left|\frac{dZ_n}{dt}\right| = U_o\left(1 + gt_n \bullet \sum_{m \neq n} f\left(Z_m - Z_n\right)\right) \qquad (12)$$

$$f(Z) = Z/|Z| \exp\left(-|Z|/\lambda\right)$$

where t_n, the vector tangent to the perimeter, is dotted with the repulsion vector f, g is a speed gain, U_o is the speed in the absence of other vehicles, and λ is the interaction decay length. The gain g is typically of order 0.1-0.5, higher values producing faster spreading. The interaction decay length is of order 0.1-0.5 times the size of the perimeter, lower values tending to emphasize local interaction. The tangent vector is estimated from the mean vehicle heading. The speed update law is applied on a timescale long compared with the time between perimeter crossings.

An example is shown in Figure 6. Initially the UUVs form a cluster. Because of the repulsion, the head and tail vehicles move faster and slower respectively. Eventually the UUVs cover the perimeter homogeneously.

UUV-gas coordination preserves the stability and convergence properties of the single vehicle algorithm because speed only affects bead radius and because convergence and stability are independent of bead radius.

Regardless of the number of vehicles or their distribution, the algorithm functions. The algorithm is therefore robust to loss of vehicle.

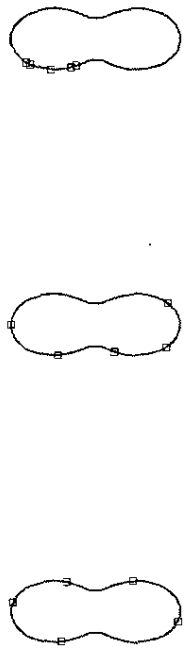

Figure 6. Time lapse sequence of UUV-gas algorithm on a "figure 8" perimeter. The UUVs are initially clustered. Through pair repulsion, they are able to spread around the perimeter.

IV. Discussion

A. Comparison

Table 1 compares the properties of the two algorithms. Snake properties are based on the circular perimeter.

The main differences are type of signal, convergence, and stability. The UUV-gas algorithm works with either a gradient or a binary signal; the snake requires a gradient. Convergence for UUV-gas is guaranteed; the snake has multiple solutions and the final state depends on initial conditions. The UUV-gas algorithm is stable; the snake is conditionally stable.

TABLE I
COMPARISON BETWEEN ALGORITHMS

Property	Snake	UUV-gas
Signal	Gradient	Gradient or binary
Convergence	Depends on initial conditions Multiple solutions	Convergent
Stability	Up to maximum elasticity	Stable
Noise	Robust	Robust
Comm	Synchronous or asynchronous	Synchronous or asynchronous

B. Extension

Can the snake algorithm be analyzed for a general perimeter? We believe that in general, at most a semi-quantitative theory is possible. Consider for example the problem of finding the steady state solution. When α=0, the steady-state coincides with the perimeter. Expanding in powers of α around the perimeter gives $Z = perimeter + z(s)$, where $|z(s)| = \alpha / R(s)|\nabla C(s)|^2$ and $R(s)$ is the radius of curvature at s. To first order, we find that the steady-state snake is separated from the perimeter by an amount that depends on the radius of curvature. Compared with Section II, this is only an approximate result, but however approximate, it does provide important insight into the problem.

Similarly with stability. Our analysis of the circular perimeter showed that stability cannot in general be understood in terms of local properties which in retrospect can be attributed to the extended nature of the eigenstates.

An analogy with electronic states in 1D solids might be useful. Electronic eigenstates obey an equation similar to the linearized snake equation (Schrodinger's equation, $E\psi = -\partial_{ss}^2\psi + V(s)\psi$). Because of its importance in the theory of disordered solids, the properties of the eigenstates when the potential V is irregular have been well studied. It is now understood that the eigenstates are delocalized when the potential V has translational symmetry and localized when it doesn't (Anderson theorem, [12]). The similarity of the two equations suggests that for complex perimeters, the eigenmodes might be localized, in which case a local theory of stability might be possible.

C. Coordination

UUV coordination has been discussed by a number of authors [13]-[22]. It is often recognized that limited communication bandwidth and vehicle loss are crucial factors. Often, it is also tacitly held that efficiency, for example mission time optimization is an equally important ojective. We argue that in comparison with the first three, efficiency is relatively unimportant.

A perfectly efficient algorithm would spread UUVs evenly around the perimeter. One could then compare algorithms on the basis of how well they spread the vehicles. Imagine two algorithms, one that guarantees a perfectly even distribution but that is not robust, and one that spreads the vehicles "acceptably" but is robust. The second algorithm is clearly better regardless of its optimality. It is better because the most important mission determinants are getting the vehicles back and surviving any loss of communication. Efficiency is secondary, unless one redefines it to explicitly incorporate these factors.

REFERENCES

[1] S. MacIntyre, K. M. Flynn, R. Jellison and J. R. Romero, 1999, *Boundary mixing and nutrient flux in Mono Lake, CA*. Limnol. Oceanogr. **44**, 512.

[2] W. R. Boynton, L. Murray, J. Hagy, C. Stokes, and W. M. Kemp, 1996. *A comparative analysis of eutrophication patterns in a temperate coastal lagoon*, Estuaries **19(2B)**, 408.

[3] J. G. Bellingham, H. Schmidt, C. Chryssostomidis, 1998, *Real-time oceanography with autonomous ocean sampling networks*, ONR annual report, available on-line at http://auvlab.mit.edu/MURI/annual-report98.htm.

[4] B. Fletcher, 2001, *Chemical plume mapping with the remus AUV*, Proceedings of Unmanned Untethered Submersible Technology.

[5] J. A. Farrell, W. Li, S. Pang, and R. Arrieta, 2003, *Chemical Plume Tracing Experimental Results with a REMUS AUV*, Proceedings of Oceans.

[6] E. Fiorelli, P. Bhatta, N. Leonard, I. Shulman, 2003, *Adaptive sampling using feedback control of an autonomous underwater glider fleet*, Proceedings of Unmanned Untethered Submersible Technology.

[7] S. T. Makrinos, 2004, *Port security in the war on terrorism*, Sea Technology **45**(3), 33.

[8] T. McInerney, D. Terzopoulos, 2000, *T-snakes: topology adaptive snakes*, Medical Image Analysis **4**, 73.

[9] D. Marthaler and A. Bertozzi, 2003, *Tracking environmental level sets with autonomous vehicles*, Proceedings of Conference on Cooperative Control and Optimization.

[10] A. Bertozzi M. Kemp, and D. Marthaler, 2004, *Determining environmental boundaries: asynchronous communication and physical scales*, Proceedings of Block Island Workshop on Cooperative Control, edited by V. Kumar, N. Leonard, and S. Morse.

[11] M. Argentina, O. Descalzi, E. Tirapegui, 2002, *Periodic nucleation solutions of the real Ginzburg-Landau equation in a finite box*, Int. Jour. Bifurcations and Chaos **12**, 2219.

[12] P. Anderson, 1958, Absence of diffusion in certain random lattices, Physical Review **109**, 1492.

[13] D. J. Stilwell, B. E. Bishop, 2001, *Platoons of underwater vehicles: Communication feedback, and decentralized control*, IEEE Control Systems Magazine **20**(6), 45.

[14] A. J. Healey and Y. Kim, 2000, *Control and random searching with multiple robots*, Proceedings IEEE CDC.

[15] H. Voicu and N. Schmajuk, 2001, *Three dimensional cognitive mapping with a neural network*, Robotics and Autonomous Systems **35**, 23.

[16] N. E. Leonard and E. Fiorelli, 2001, *Virtual leaders, artificial potentials, and coordinated control of groups*, Proceedings of IEEE CDC.

[17] A. Martins, J. M. Almeida, and E. Silva, 2003, *Coordinated maneuvers of gradient search using multiple AUVs*, Proceedings of Oceans.

[18] R. Kannan and B. Bourgeois, 2003, *Efficient protocol for integrated communication and formation control in UUV task forces*, Proceedings of Unmanned Untethered Submersible Technology.

[19] E. Honary, D. McFarland, and C. Melhuish, 2003, *Mapping the oceans using flock distortions*, Proceedings of Unmanned Untethered Submersible Technology.

[20] K. Lee, 2003, *Optimal control for multiple unmanned underwater crawling vehicle*, Proceedings of Unmanned Untethered Submersible Technology.

[21] C. Duarte, G. Martel, E. Eberbach, C. Buzzel, 2003, *Control language for dynamics tasking of multiple autonomous vehicles*, Proceedings of Unmanned Untethered Submersible Technology.

[22] B. Cook, D. Marthaler, C. Topaz, A. Bertozzi, and M. Kemp, 2003, *Fractional Bandwidth Reacquisition Algorithms for VSW-MCM*, Proceedings of International Workshop on Multi-Robot Systems.

Mathieu Kemp is Director of Physics at Nekton Research. He earned the Ph. D. in physics from University of North Carolina at Chapel Hill in 1992, the MScA in applied physics from Ecole Polytechnique de Montreal in 1988, and the B. Ing in engineering physics from Ecole Polytechnique de Montreal in 1986.

He previously held appointments at Duke University, Northwestern University, Universidad de Los Andes, and Ecole Polytechnique de Montreal.

Dr. Kemp is a member of the American Association for the Advancement of Sciences and of the Institute of Navigation.

Andrea Bertozzi is Professor of Mathematics at UCLA and Professor of Mathematics and Physics at Duke University. She earned the Ph. D., M. A. and A. B. degrees in mathematics from Princeton University in respectively 1991, 1988, and 1987.

Previous titles include L. E. Dickson Instructor of Mathematics at the University of Chicago (1991-1995) and Maria Geoppert-Mayer Distinguished Scholar at Argonne National Lab (1995-6). She also participated as a summer fellow at Bell Labs Murray Hill during the years 1987-9.

Professor Bertozzi is a member of the American Mathematical Society, American Physical Society, the Society of Industrial and Applied Mathematics, and the Association for Women in Mathematics. She currently serves on the editorial board of SIAM Review, Interfaces and Free Boundaries, and Applied Mathematics Research Express. Her honors include a Sloan Research Fellowship and the Presidential Early Career Award for Scientists and Engineers.

Daniel Marthaler is Assistant Researcher of Mathematics at UCLA. He earned the Ph. D. and M. A. degrees at Arizona State University, and the B. S. degree in mathematics from the University of Arizona in respectively 2002, 2000, and 1998.

He served in the United States Army as a machine gunner in the 101st Airborne before being honorably discharged in 1994. He was a Research Assistant in the Department of Mathematics at Duke University prior to his current position.

Dr. Marthaler is a member of the American Mathematical Society and the Society of Industrial and Applied Mathematics.