An Approximation Algorithm for Computing the Mean Square Error Between Two High Range Resolution RADAR Profiles

John Weatherwax

***Abstract*— In this paper we present an approximation method for computing the one dimensional mean square error (MSE) between two High Range Resolution (HRR) RADAR profiles. In much of the signal processing literature the MSE is computed with a brute force exhaustive search or a cross-correlation technique. In this paper we wish to emphasize and develop an alternative approximate computational technique for its calculation. We found that the MSE calculation could be approximated well by two one-dimensional minimization and show how to optimally compute each minimization. The technique we present has the added features that it does all of its calculations in the spatial domain.**

I. Introduction

IN much of one–dimensional signal processing, two finite length signals are compared using a mean square error (MSE) metric. In this paper we envision that two length N signals ($x[\cdot]$ and $t[\cdot]$) are to be compared with the following mathematical expression

$$\text{MSE}(x[\cdot],\, t[\cdot]) = \text{Min} \sum_{i=0}^{N-1} (\alpha\, x[i-s] - t[i])^2 \,. \qquad (1)$$

Here the minimum, in the above expression, is taken over an index shift s, and a real valued scaling factor α. Physically, in this expression each vector (x and t) represents a signal corrupted with additive noise. Mathematically, a MSE score between two one–dimensional vectors represents how alike the two signals are, regardless of signal amplitude and position. A large MSE score indicates a large difference, while a small score indicates signals that are very similar.

A specific application where this MSE criterion is used in RADAR signal processing is that for template matching of two High Range Resolution (HRR) RADAR profiles [1]–[7]. As an example of the type of signals considered there, in figure 1 we present a sample Moving and Stationary Target Acquisition and Recognition (MSTAR) HRR profile taken from a M9 tank. In this paper our application of the MSE will be taken from HRR RADAR signal processing, but the analysis we do will be applicable to many other signal processing domains where similar situations hold.

Manuscript received January 1, 2005; revised December 30, 2005. This work was sponsored by the U.S. Government under Air Force Contract F19628-00-C-0002. Options, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

J. Weatherwax is with the Massachusetts Institute of Technology, Lincoln Laboratory.

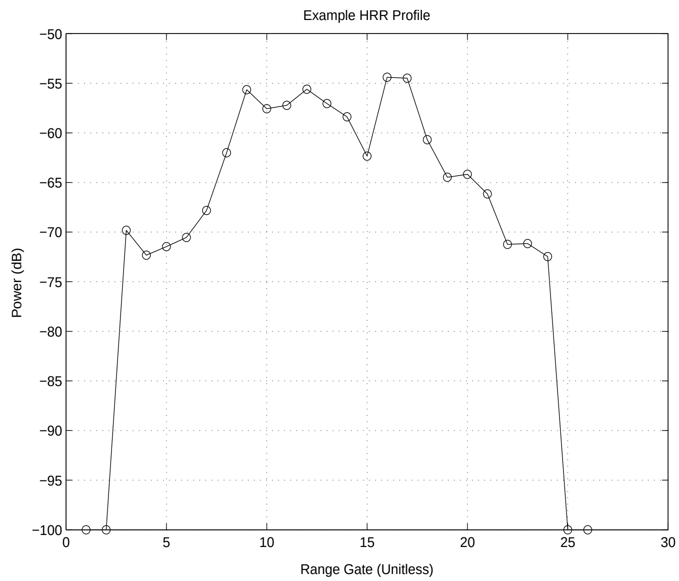

Fig. 1. Typical MSTAR RADAR High Range Resolution profi le.

In what follows, the vector $x[\cdot]$ will often be referred to as the "test" signal and the vector $t[\cdot]$ will often be referred to as the "template" signal. This choice of terminology is derived from one operational use of the MSE in template matching. In this application a new input test signal x would be compared with every template signal t in a template library. The template signal with the smallest MSE score would be chosen to most represent the test signal. Signal classification or additional processing could be done with this information.

Physically, the amplitude (α) in equation 1 is included to insure that the test signal $x[\cdot]$ has the same signal to noise ratio as the template signal. The shift s is to insure that both profiles have their signal properly centered. We envision that shifts of a profile only slide the signal portion of the vector along a fixed and infinite noise floor. Thus the range of the shift parameter is not bounded, but different MSE scores only result for $s \in (-N, N)$.

It can be seen from equation 1 that a given MSE calculation can be considered a two–dimensional minimization problem; that of minimizing the function

$$S(\alpha, s) \equiv \sum_{i=0}^{N-1} (\alpha x[i-s] - t[i])^2 \,, \qquad (2)$$

over the real variable α and the *discrete* variable s. See figure 2

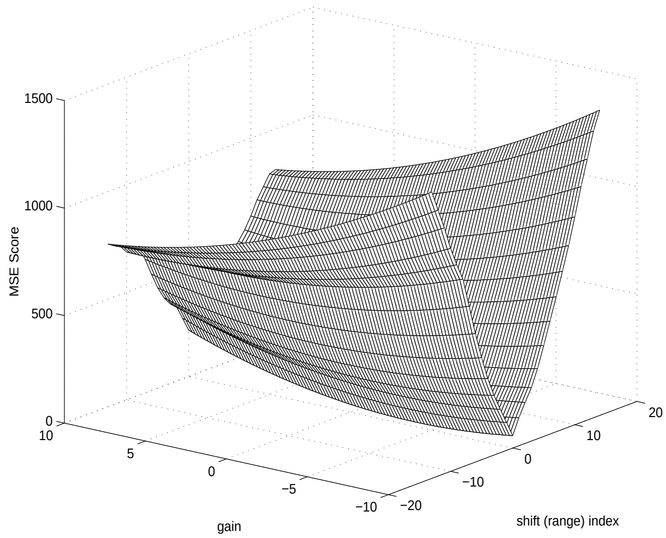

Fig. 2. Typical quadratic unimodal MSE surface behavior. The gain α and shift index s are varied along the x and y axes respectively. A unique minimum results at the vertex of this "parabola."

for a typical plot of the $S(\alpha, s)$ surface. In that figure one can see the nice quadratic – looking structure that results along each dimension. The rest of the paper presents the approximate minimization technique used in calculating an approximate value for the MSE.

II. An Approximate Technique at Calculating the MSE Minimum

For all of the cases considered in our research the two-dimensional minimization represented in figure 2 could be well approximated by *two* one-dimensional minimizations. While it is well known that this is not in general true for arbitrary functions of two variables we found that this procedure represented a very good approximation for computing the MSE in the following sense.

In the template matching problem considered here it is important that any procedure used to calculate the MSE truly select the global minimum (or something very close to it) when the two HRR profiles are from the *same* class and the resulting MSE small. If the two HRR profiles are from *different* classes it is less important that our procedure return the exact minimum MSE. For by returning something larger than the exact minimum MSE between two dissimilar HRR profiles we increase our probability of excluding this template profile as a match to our input test profile. While this argument is not rigorous, again, in practice with many comparisons between similar and different profiles this two minimization procedure worked very well.

In performing our two one-dimensional minimizations, we first minimize the function S with respect to the gain α over a fixed shift index ($s = 0$), and then with that value of α we minimize the function S over a the shift parameter s. The minimization over α can be done analytically. The result with $s = 0$ is given by standard calculus and is given by

$$\alpha^* = \frac{\sum_{i=0}^{N-1} t[i]\, x[i]}{\sum_{i=0}^{N-1} x[i]^2}\,. \tag{3}$$

We note that in the operational setting that motivated this work we found it advantageous to "mean match" the test signal to the template signal, before using equation 3. Mean matching, involves computing the mean of the test signal μ_{test} and the mean of the template μ_{temp} signal and adding to the test signal a constant equal to $\mu_{temp} - \mu_{test}$.

The method used to minimize over s is more interesting and can greatly affect algorithm performance. To the authors' knowledge, at this point in the MSE calculation many methods perform some sort of brute force exhaustive search over the discrete index s or use a cross-correlation technique [7]. Rather use either technique we sought a method that was computationally cheaper than brute force and that avoided the use of FFT's. The technique we use to arrive at a minimum of S with respect to s, is called a discrete Fibonacci search (also called a lattice search [8]). We were led to a discrete Fibonacci search by the following observations about the second minimization:

- The independent variable s is discrete and therefore Newton type methods that rely on derivatives cannot be used.
- Each function evaluation of S is very expensive involving $O(N)$ additions.

To optimize the MSE calculation with respect to speed we would like a method that uses as few function evaluations as possible; in fact, we would prefer a method that *reuses* function evaluations if possible in locating this unique minimum. The *provable optimal* algorithm for this task is the Fibonacci search [9]. In the next section we will briefly discuss this algorithm with respect to our problem of interest.

We note, before presenting the details of the Fibonacci search, that the cross-correlation technique is also a computationally efficient method at producing the required range shift [10]. In addition, we will show in the appendix below, that our proposed search technique has the *same* computational complexity as the cross-correlation technique. Thus from a computational viewpoint there is no difference between the two techniques. A proof of this statement is presented in the appendix below.

III. The Discrete Fibonacci Search as Applied to the MSE Calculation

The Fibonacci search algorithm in the context of our problem will be explained in this section. However, before we can fully explain this search algorithm we need a few definitions and some background.

A discrete function f is *unimodal*[1] on an interval $[a, b]$ (with a and b integers) if f has a minimum i_0 in the interval, and f is strictly decreasing to the left of i_0 and strictly increasing to the right of i_0. See figure 3 for a graphical representation of a unimodal function. Consider now the problem of locating

[1] Some authors designate function with one minimum as uni**n**odal [11]

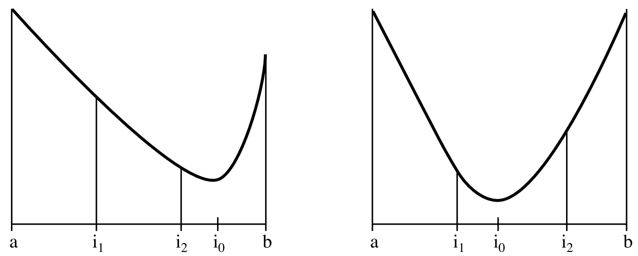

Fig. 3. Examples of unimodal functions. In the function on the left the new search interval should be $[i_1, b]$, while in the function on the right the new search interval should be $[a, i_2]$.

this extrema. Standard iterative "bracketing" algorithms [12], [13] attempt to reduce the interval under consideration by choosing two arbitrary intermediate points $i_1, i_2\,(a < i_1 < i_2 < b)$, examining the value of the function at i_1 and i_2, and then discarding one of the subintervals, $[a, i_1)$ or $(i_2, b]$. For example, if $f(i_1) > f(i_2)$, the interval $[a, i_1)$ can be excluded (see figure 3) and all our attention can be focused on the reduced interval $[i_1, b]$, which still bracket our minimum i_0.

The functions to which such bracketing algorithms are typically applied have rather complex non-analytical forms, are very expensive to evaluate, or are integer valued. If they were not of these types, standard newton type algorithms that utilize derivative information could be used (with quadratic convergence). Because the functional form of the MSE problem we are minimizing, each functional evaluation is expensive (on the order of N additions) *and*, the search space is discrete, newton methods are of no use. Now in continuing our bracketing routine we would select two new points $\hat{i}_1$ and $\hat{i}_2$ inside $[i_1, b]$ and based on function evaluations of these points, eliminate more interval as done above. In selecting the two *new* points from the reduced subinterval $[i_1, b]$ at which we will evaluate f, it is natural to hope that one of them could be the previous chosen i_2 at which f has *already* been evaluated. Thus we get information on the location of the minimum of our function without any additional functional evaluations. The problem, of course, is that an unfortunate original choice for i_2 could lead to an insignificant reduction at this stage in the size of our interval bracketing i_0.

The answer to the question as to how to choose the i points to reuse information and to maximize the amount of interval reduction is given by the Fibonacci search. The minimization technique is named such because it uses the Fibonacci sequence, defined by

$$F_0 = 0 \qquad F_1 = 1 \qquad \text{and} \qquad F_{m+1} = F_m + F_{m-1}\,, \quad (4)$$

for $m > 1$. The first 13 Fibonacci numbers are given as follows

$$0 \quad 1 \quad 1 \quad 2 \quad 3 \quad 5 \quad 8 \quad 13 \quad 21 \quad 34 \quad 55 \quad 89 \quad 144 \quad (5)$$

With this quick background the algorithm is as follows, please see any of the references at the end of this paper for a more thorough description. For a discrete search with $a = 1$ to $b = N$ (all discrete problems can be shifted to this form), select as i_2 the largest Fibonacci number less than the total number of shift indices (here N). This means that one finds the largest integer m such that

$$F_m < N\,, \quad (6)$$

where F_m are the Fibonacci numbers. Given the previously computed optimal $\alpha = \alpha^*$, from equation 3, evaluate the function S at the two points $i_2 = F_m$, and $i_1 = F_{m-1}$. The function values at our two internal test points will be called $S_1 \equiv S(\alpha^*, i_1)$ and $S_2 \equiv S(\alpha^*, i_2)$ points. With these two values the main iterative loop begins. We loop until $|i_2 - i_1| = 1$. If

$$S(\alpha^*, i_1) < S(\alpha^*, i_2) \quad (7)$$

then the new interval of search should be $[a, i_2]$, and the following assignments are made for the next pass through the loop,

$$\begin{aligned} a &= a & (8)\\ b &= i_2 & (9)\\ i_2 &= i_1 & (10)\\ S_2 &= S_1 & (11)\\ i_1 &= a + (b - i_2) & (12)\\ S_1 &= S(\alpha^*, i_1) & (13) \end{aligned}$$

On the other hand if

$$S(\alpha^*, i_1) > S(\alpha^*, i_2) \quad (14)$$

then the new interval of search should be $[i_1, b]$, and the following assignments are made for the next pass through the loop,

$$\begin{aligned} a &= i_1 & (15)\\ b &= b & (16)\\ i_1 &= i_2 & (17)\\ S_1 &= S_2 & (18)\\ i_2 &= b - (i_1 - a) & (19)\\ S_2 &= S(\alpha^*, i_2) & (20) \end{aligned}$$

Each time through the loop we decrease the interval surrounding our minimization, and it can be shown that our internal test points i_1 and i_2, are Fibonacci numbers. In addition, after m loops this routine will terminate. Thus, in general, the minimum can be found in m function evaluations with m equal to the index on the largest Fibonacci number less than or equal to N. All of these facts are demonstrated and discussed in [8]. In the next section we compare this search algorithm computationally against a more naive exhaustive search.

IV. ALGORITHM COMPARISONS

Here we compare our approximate technique against an exhaustive search algorithm for computing the MSE. In the exhaustive search α is discretized between -5 and 5 in steps of 0.5, and the shift index s is taken from within the range of $(-N/2, +N/2)$. Timing results are shown in figure 4. Both algorithms were developed in MATLAB and run on a Pentium PC running Linux.

On the x-axis we plot the number of randomly chosen HRR profiles and on the y-axis we plot the time (in seconds)

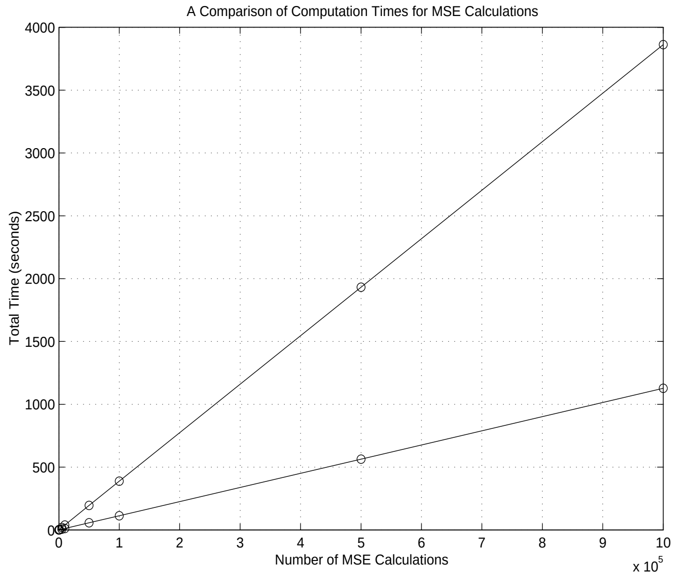

Fig. 4. Comparison of algorithmic timings between an exhaustive search technique and the Fibonacci search.

required to compare a fixed test profile with all profiles. The timing was done with the MATLAB "tic" and "toc" function. These timing results clearly show that a Fibonacci search speeds up the MSE computation, by about 350%. Using a Fibonacci search, one can do the computation of the MSE for about three and a half templates in the time it takes to do one template in the exhaustive search technique.

V. Conclusion

In this paper we have motivated the fact that a very good method for computing the MSE between two HRR signals of interest can be approximated well by two one–dimensional minimization techniques. After that motivation, we have presented a computationally efficient method at evaluating the minimization along each direction. This two minimization decomposition technique (as an approximation technique for MSE calculations) does not seem to have been made elsewhere. We hope that this observation will motivate other researchers to try similar techniques on other one–dimensional template matching problems that use the MSE.

Appendix
Computational Complexity of the Fibonacci Search

In this appendix we show that the computational complexity of the Fibonacci search technique on HRR profiles of size N, is equal to that of a cross-correlational technique used for the same purpose.

A cross-correlation technique requires $O(N\log(N))$ calculations for the FFTs on each profile. A component wise product and a maximization each require $O(N)$ additional calculations. Thus, the cross-correlation technique in total requires

$$O(2N\log(N) + 2N) = O(N\log(N)) \tag{21}$$

calculations to obtain its solution s.

For the Fibonacci search, it can be shown [14] that the Fibonacci numbers F_m can be written as

$$F_m = \frac{1}{\sqrt{5}}(\phi^m - \hat{\phi}^m) \tag{22}$$

where ϕ and $\hat{\phi}$ are defined to be

$$\phi = \frac{1+\sqrt{5}}{2} \approx 1.6180339 \tag{23}$$

$$\hat{\phi} = \frac{1-\sqrt{5}}{2} \approx -.6180339 \tag{24}$$

Since the Fibonacci search requires m function evaluations [8] where m is given in terms of N by the implicit equation

$$F_m \approx N\,. \tag{25}$$

Each function evaluation requires $O(N)$ additions so we get a total of $O(mN)$. Solving equation 25 for m as a function of N, using equation 22 gives

$$m = \frac{\log(\sqrt{5}N)}{log(\phi)}\,. \tag{26}$$

together this gives a total complexity of the Fibonacci search technique of $O(N\log(N))$, the same as that of the cross-correlation technique.

Acknowledgment

The author would like to thank Dr. John Kay for his help and encouragement while this research was carried out.

References

[1] D. H. Nguyen, G. R. Benitz, J. H. Kay, B. Orchard, and R. H. Whiting, "Superresolution HRR ATR with high definition vector imaging," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, pp. 1–19, 2001.
[2] D. H. Nguyen, G. R. Benitz, J. H. Kay, B. J. Orchard, and R. H. Whiting, "Improving HRR ATR performance at low SNR by beamspace HDI," in *Proceedings of the SPIE Conference on Automatic Target Recognition XI*, no. 11, Orlando, Fl, 2001.
[3] ——, "Improving HRR ATR performance at low SNR by multi-look adaptive weighting," in *Proceedings of the SPIE Conference on Automatic Target Recognition XI*, no. 11, Orlando, Fl, 2001.
[4] D. H. Nguyen, G. R. Benitz, J. H. Kay, and R. H. Whiting, "Super-resolution HRR ATR performance with HDVI," in *Proceedings of the SPIE Conference on Automatic Target Recognition X*, no. 10. Orlando, Fl: SPIE, 2000, pp. 418–427.
[5] ——, "Super-resolution high range resolution ATR with HDVI," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, 2001.
[6] D. H. Nguyen, J. H. Kay, B. J. Orchard, and R. H. Whiting, "Classification and tracking of moving ground vehicles," *MIT Lincoln Laboratory Journal*, pp. 1–50, 2002.
[7] A. Zyweck and R. Bogner, "Radar target classification of commercial aircraft," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, pp. 598–606, 1996.
[8] D. J. Wilde, *Optimum Seeking Methods*. Englewood Cliffs, N.J.: Prentice-Hall, Inc, 1965.
[9] J. Kiefer, "Sequential minimax search for a maximum," *PAMS*, vol. 4, pp. 502–506, 1953.
[10] S. W. Smith, *Digital Signal Processing: A Practical Guide for Engineers and Scientists*. McGraw-Hill N.Y.: Newnes, 2002.
[11] D. J. Wilde, *Globally Optimal Design*. New York, NY: Wiley–Interscience Publication, 1969.
[12] R. P. Brent, *Algorithms for Minimization Without Derivatives*. Englewood Cliffs, N.J.: Prentice-Hall, Inc, 1973.

[13] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*. New York, NY: Cambridge University Press, 1988.

[14] C. TH, L. CE, and R. RL, *Introduction to Algorithms*. McGraw-Hill N.Y.: MIT Press, 1990.

PLACE PHOTO HERE

John L. Weatherwax received a B.S. with honors in mathematics and a S.B. with honors in physics from the University of Missouri–Columbia in 1996. In September 1996 he attended Massachusetts Institute of Technology, Cambridge Ma. with a National Science Foundation Fellowship. In 2001 he graduated from M.I.T. receiving a doctorate in mathematics in the area of non–linear hyperbolic systems. From 2001 on, he has been a member of the technical staff working at M.I.T. Lincoln Laboratory, Lexington Ma. Currently he is in the ballistic missile division working on discrimination algorithms.