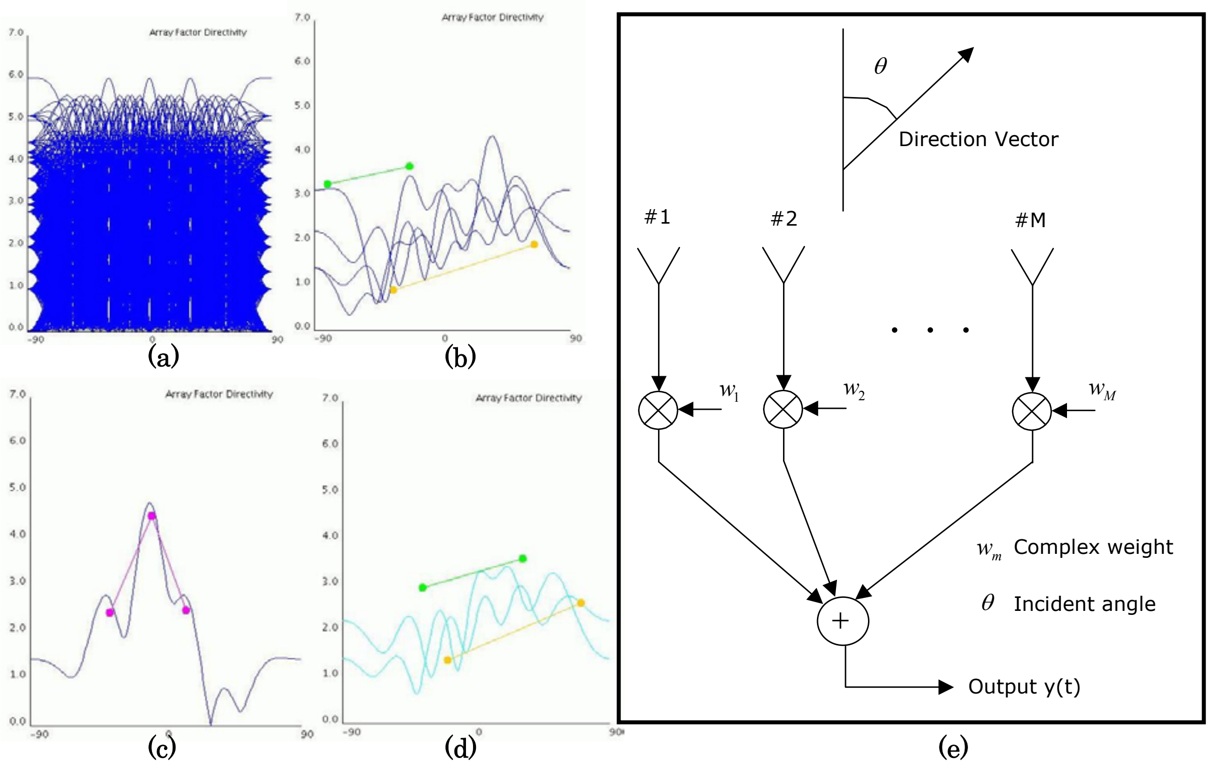
EXHAUSTIVE GENERATION AND VISUAL BROWSING FOR RADIATION PATTERNS OF LINEAR ARRAY ANTENNAS

Darren Leigh, Tom Lanning,
Neal Lesh, Kathy Ryall
Mitsubishi Electric Research Laboratories
201 Broadway, 8th floor
Cambridge, MA 02139
USA
{lanning, leigh, lesh, ryall}@merl.com

Hiroaki Miyashita, Shigeru Makino
Antennas Technology Department
Information Technology R&D Center
Mitsubishi Electric Corporation
5-1-1 Ofuna, Kamakura, Kanagawa 247-8501,
Japan
{miyas, makino}@isl.melco.co.jp

1. Introduction

Almost any obtainable radiation pattern can be achieved with a phased array antenna if the phases and amplitudes are chosen correctly. However, if these are quantized, it can be a time consuming and difficult process for a human expert to determine the best quantized excitation coefficients to produce a desired radiation pattern. In this paper, we explore the use of exhaustive generation of all possible permutations of quantized excitation coefficients. We provide visual query methods which allow a designer to quickly explore the large number of resulting designs. In particular, we demonstrate our techniques with a system for phase-only synthesis of uniform linear arrays (see diagram, Figure 1(e)) with quantized phase coefficients.



(a) (b) (c) (d) (e)

Figure 1: Example queries for a sample antenna analysis data set: (a) all the plots, (b) min and max hard constraints prune the matches, (c) a goal query, (d) an over-constrained query with two soft matches, and (e) a diagram of a uniform linear array for phase-only synthesis.

Antenna performance data (e.g., radiation patterns) and other data sets involving linear ordered sequences (e.g., finance, weather, genetics, census data) are especially well-suited for visual query techniques; the graphical nature of the input and output methods support users in their information-seeking tasks, in this case designing a phased array antenna with particular performance

ISBN: 4-88552-207-2 C3055©IEICE

characteristics. Exploring these large data sets, however, raises a number of issues. The first issue is the question of how to best examine the data. Looking at the plots (i.e., radiation patterns) sequentially, one at a time, is too time consuming. Looking at all the data in parallel (e.g., overlaying all of the plots in to a single display) reveals little if any information; the result is often an indistinguishable blob as shown in Figure 1(a)[1]. One approach is to use hard constraints to prune the data set, as shown in Figure 1(b). Unfortunately pruning the space before one knows its content may remove some items of interest.

When using visual queries, a second issue is what types of queries to permit a user to compose and how best to help users refine their queries. In many cases when looking for patterns in the data, constraints are typically too rigid; users might be looking for general shapes or trends, without knowing the exact values (e.g., looking for minimums or maximums). Under our approach, users may set a "goal" query, an example of which is shown in Figure 1(c). Here an antenna designer is looking for graphs that have a peak near the center. Under-constraining and/or over-constraining the query is another common difficulty encountered by users. In Figure 1(d) the designer has set max and min constraints that are not met by any of the data in the series. Using our approximate presentation technique, two "soft matches" (near misses) are shown; both violate the min constraint specified by the designer, but are close to satisfying the query. Our approximate techniques permit the user to explore the data set by posting constraints and preferences that are then used to select and sort a subset of the instances.

2. System Overview

We have developed a prototype system for exploring the domain of phased-array antennas[1]. Phased-array antenna designers explore trade-offs in design by evaluating a variety of input parameters (e.g., number of elements, element amplitude and phase-shift choices) and performance metrics (e.g., radiation patterns). Our goal is to support the designer in quickly examining and evaluating a large number of possible designs. Our system has two components, a *generator* and a visual browser or *visualizer*. The generator produces a set of possible designs to be explored using the visualizer. Our system is modular – any generator that can produce the data in an appropriate format may be used with our visualizer. Likewise, any visualizer can be used with our data, although to our knowledge, our visual browser is unique in its functionality.

We consider the problem of a system for phase-only synthesis of uniform linear arrays. The generator takes as input the number of elements in the array and the set of quantized values for phase and amplitude coefficients. It then uses a brute-force exhaustive process to compute the radiation patterns for all possible combinations of excitation parameters, and produces a set of correlated graphs (excitation coefficient parameter values and radiation patterns) for each design. The data is stored in the common CSV (comma-separated value) format that is used by many applications (e.g., a commercial spreadsheet/graphing package such as Excel, or the free Gnuplot plotting package). The visualizer takes as input the data file from the generator and a style sheet indicating some display parameters for presenting the data. The style sheet is a simple text file that is easily edited by the designer, and indicates things such as colors, spacing, titles, scale and so forth.

Query Lines are the graphical representation of a designer's query. The Query Line *penalty* indicates the relative positioning information for how to compare a data point with a particular Query Line. There are three types of penalties: (1) **minimum**: examines the data relative to the Query Line and penalizes anything that falls below this point (2) **maximum**: examines the data relative to the Query Line and penalizes anything that falls above this point (3) **goal**: examines data relative to the Query Line and penalizes any deviation. Min and max preferences are commonly found in many query systems, and are useful for pruning data by setting absolute bounds. A goal Query Line provides designers with a mechanism to look for patterns in the data. When looking for shapes in the data, it is often useful to incorporate some fuzziness into the pattern description – it is less likely that data will exactly match a given pattern for which the designer is looking.

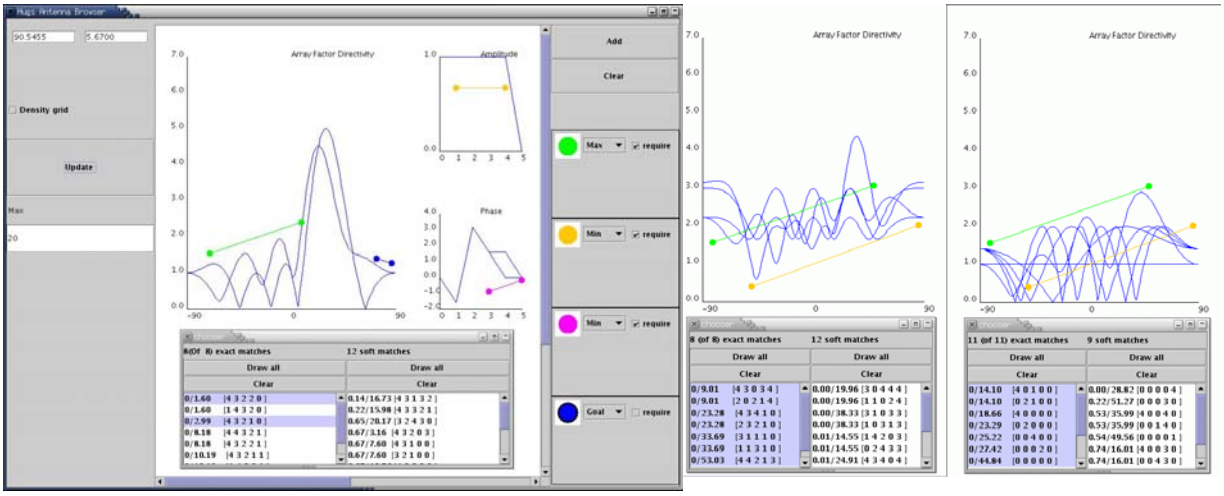
[1] In the radiation pattern figures used in this paper, the x-axis is observation angles (degrees) and y is the array factor directivity.

The Query Line *strength* tells us how to weight a penalty when a preference is not met. For simplicity, we currently define two categories: (1) **required**: these are hard constraints, and are expected to be met by the designer. Required Query Lines may be refined by the designer if they over-constrain a query. (2) **idealized**: these are user preferences, and are not necessarily expected to be fully met. Idealized Query Lines are used in ranking and sorting plots, most useful for an under-constrained query.

Based on the two Query Line strengths, we divide plots into *exact matches* and *soft matches*. Exact matches are those plots that satisfy all of the required Query Lines (i.e., constraints), and are equivalent to the results of a traditional constraint-based query. Soft matches violate one or more of the required Query Lines. In both cases, matches are scored and sorted for how much they satisfy or violate idealized user preferences.

3. Sample Application

Our generator enumerates all unique permutations of the excitation coefficients and computes their corresponding radiation patterns so that our system can permit an exhaustive search of every pattern obtainable by the given array. For phase-only array synthesis, the excitation amplitude coefficient is restricted to a value of either 0 or 1. We have assumed the use of an n-bit phase shifter which provides 2^n possible phase shifts. To avoid redundant and trivial cases, we have assumed that the element at the far left side of the array always has an amplitude of 1 and a phase shift of 0. For an m-element array, there are therefore $(2^n + 1)^{(m-1)}$ unique permutations of the excitation parameters. For the remainder of the paper we consider the particular problem of a six-element array (shown in Figure 2(a)) with half wavelength spacing between elements and a two-bit phase shifter. In this case there are 3125 unique excitations whose radiation patterns we need to generate and search. The radiation patterns contain 181 sample points (one per degree). The data set was generated in about one second, and requires approximately 4.7 megabytes of storage. It was selected for ease of demonstration and to fit on a laptop computer. Our system supports much larger examples as well.



```
8(Of  8) exact matches          12 soft matches
0/1.60   [4 3 2 2 0 ]           0.14/16.73 [4 3 1 3 2 ]
0/1.60   [1 4 3 2 0 ]           0.22/15.98 [4 3 3 2 1 ]
0/2.99   [4 3 2 1 0 ]           0.65/20.17 [3 2 4 3 0 ]
0/8.18   [4 4 3 2 1 ]           0.67/3.16  [4 3 2 0 3 ]
0/8.18   [4 3 2 2 1 ]           0.67/7.60  [4 3 1 0 0 ]
0/10.19  [4 3 2 1 1 ]           0.67/7.60  [3 2 1 0 0 ]
```

```
8 (of 8) exact matches          12 soft matches
0/9.01   [4 3 0 3 4 ]           0.00/19.96 [3 0 4 4 4 ]
0/9.01   [2 0 2 1 4 ]           0.00/19.96 [1 1 0 2 4 ]
0/23.28  [4 3 4 1 0 ]           0.00/38.33 [3 1 0 3 3 ]
0/23.28  [2 3 2 1 0 ]           0.00/38.33 [1 0 3 1 3 ]
0/33.69  [3 1 1 1 0 ]           0.01/14.55 [1 4 2 0 3 ]
0/33.69  [1 1 3 1 0 ]           0.01/14.55 [0 2 4 3 3 ]
0/53.03  [4 4 2 1 3 ]           0.01/24.91 [4 3 4 0 4 ]
```

```
11 (of 11) exact matches        9 soft matches
0/14.10  [4 0 1 0 0 ]           0.00/28.82 [0 0 0 0 4 ]
0/14.10  [0 2 1 0 0 ]           0.22/51.27 [0 0 0 3 0 ]
0/18.66  [4 0 0 0 0 ]           0.53/35.99 [4 0 0 4 0 ]
0/23.29  [0 2 0 0 0 ]           0.53/35.99 [0 0 1 4 0 ]
0/25.22  [0 0 4 0 0 ]           0.54/49.56 [0 0 0 0 1 ]
0/27.42  [0 0 0 2 0 ]           0.74/16.01 [4 0 0 3 0 ]
0/44.84  [0 0 0 0 0 ]           0.74/16.01 [0 0 4 3 0 ]
```

(a) (b) (c)

Figure 2: Prototype system: (a) The main viewing window displays three sets of plots; the chooser window shows the results of query, and controls which plots to display. Comparing idealized and required Query Lines: (b) idealized max Query Line and required min Query Line, and (c) required max Query Line and idealized min Query Line.

In this paper we focus primarily on searching the radiation patterns. A screenshot of the interface is shown in Figure 2(a). The main viewing window displays three tightly-coupled plots for the underlying antenna data (radiation pattern, amplitude, and phase-shift). The chooser window (the smaller inset window) provides access to the results of the query. It also controls which plots to display in the main viewing window by permitting designers to select one or more plots, along with information on how many of each match were found. Exact matches are listed on the left, soft matches on the right. For this example the designer has specified that 20 total matches be shown ('Max' 20, left part of Figure 2(a)). In this case the designer has selected two of the exact matches to display. On the right hand side of the main window are the controls for the Query Lines. The designer has composed a query using four Query Lines, two in the radiation pattern plots, one in the amplitude, and one in the phase-shift data. The largest plot in the main viewing window shown in Figure 2(a) displays radiation patterns, which are the primary performance metric of interest when analyzing antenna performance. The interface also provides support to the designer for drawing Query Lines. Initially the system provides controls for the designer to specify up to three Query Lines. Traditional GUI components (e.g., pull-down menus, check boxes and text inputs) permit the designer to change the penalty of each Query Line (e.g., min, max, goal), its strength (e.g., required or idealized). The designer may add as many additional Query Lines as needed.

Designers specify a query by drawing one or more Query Lines in any of the viewing windows. A designer may specify any number of Query Lines, and may add, delete or modify any particular Query Line at any time. Visually, each Query Line consists of a series of line segments, with control points for easy access and manipulation. Matches are divided into "exact" matches (displayed on the left), and "soft" matches (displayed on the right). Each match includes its two scores, indicating how well it meets the required and idealized preferences specified by the designer. A designer may scroll through the results individually, or compare several (or all) in parallel. Selecting one or more matches in the chooser window causes the corresponding plots to be displayed in the main viewing window. Exact matches are shown in a darker blue color. Soft matches are shown in a lighter color.

Figure 2(b-c) shows how the designer can combine different types of Query Lines using both required and idealized strengths. In this example the query is composed of two Query Lines (min and max), in which we swap the idealized and required parameters. In 2(b), the min Query Line is required while the max Query Line is idealized. In 2(c) the min Query Line has been "softened" to idealized, while the max has been changed to required. The results of these two queries are quite different sets of plots. The idealized Query Line in each example ranks the matches permitted by the required Query Line. This example illustrates the expressive power of our approach.

4. Conclusion

Our visual query techniques have similarities to that of TimeSearcher [2,3]. In TimeSearcher, the designer draws boxes rather than the lines used in our system. In our terminology, a simple Timebox represents two hard constraints: the top of the box operates exactly like a required max Query Line; the bottom of the box operates exactly like a required min Query Line. More importantly, the main focus of our work is on approximate query and presentation techniques that are not present in TimeSearcher.

With respect to the domain of antenna design, the key feature of our approach is that the data sets contain all possible permutations of excitation coefficients, and thus the antenna designer is guaranteed that the exact solution is in the data set he is browsing. Furthermore, the flexibility of our visual querying techniques allow the designer to quickly and easily find the designs that most closely satisfy their target performance, and then choose the best one from that set.

5. ACKNOWLEDGMENTS

We would like to thank Kent Wittenburg, Joe Marks, and David Wong for their valuable suggestions and support of this work.

6. REFERENCES

[1] R. C. Hansen Ed., "Microwave Scanning Antennas," Peninsula Publishing,, Los Altos CA, 1985.

[2] H. Hochheiser and B. Shneiderman, "Visual Queryies for Finding Patterns in TimeSeries Data," HCIL-2002-06.

[3] E. Keogh, H. Hochheiser, and B. Shneiderman, "An Augmented Visual Query Mechanism for Finding Patterns in Time Series Data," Proc. Fifth International Conference on Flexible Query Answering Systems, 2002. Also as HCIL-2002-15.