Image Algebra And Morphological Template Decomposition

Gader, Paul, Dunn, Elizabeth

Paul Gader, Elizabeth G. Dunn, "Image Algebra And Morphological Template Decomposition," Proc. SPIE 1098, Aerospace Pattern Recognition, (30 August 1989); doi: 10.1117/12.960432

Event: SPIE 1989 Technical Symposium on Aerospace Sensing, 1989, Orlando, FL, United States

Image algebra and morphological template decomposition

Paul Gader

Environmental Research Institute of Michigan
P.O. Box 8618, Ann Arbor, Michigan 48107-8618

Elizabeth G. Dunn

University of Wisconsin, Department of Agricultural Economics
Taylor Hall, 427 Lorch Street, Madison, Wisconsin 53706

ABSTRACT

Development and testing of image processing algorithms for real-time aerospace pattern recognition applications can be extremely time consuming and labor intensive. There is a need to close the gap between high-level software environments and efficient implementations. Image algebra is an algebraic structure designed for image processing that can be used as a basis for a high-level algorithm development environment. Systematic methods for mapping algorithms represented by image algebra statements to specific architectures are being studied.

In this paper we discuss template decomposition, a problem encountered in mapping image algebra statements to combinations of parallel and pipeline architectures. In particular, we show that the gray scale morphological template decomposition problem can be viewed as a linear problem, even though morphological transformations are nonlinear. We show how methods for solving linear programming problems and, in particular, the transportation problem can be applied to template decomposition.

1. INTRODUCTION

Image algebra is an algebraic structure specifically designed to reflect the types of computational environments commonly used in image processing.[1,2] It has been designed to serve both as a mathematical tool for image processing as well as the basis for a high-level software environment for image processing algorithm development. It has been demonstrated that the image algebra provides a succinct and general representational system for image processing algorithms; the image algebra is capable of expressing all image-to-image transformations involving a finite number of gray levels as well as a large variety of image-to-feature computations. The interaction of the image algebra with the symbolic processing domain is also being studied.

One of the next major steps involving the image algebra is the formation of a software algorithm development environment based on image algebra. One goal for creating such an environment is to increase programmer productivity. To meet this goal, the software environment must be effective at several levels. At the high level, it must be easy for the algorithm developer to quickly implement and test ideas. This implies the availability of a high-level representation, free from computational details. On the other hand, the computationally intensive nature of image processing demands that computations be efficient, especially during testing. This creates a conflicting demand on the system, since there is usually a tradeoff between an algorithm description that is more general and high level and one that is computationally efficient. For this reason, efficient mapping of image algebra statements and operations to processing environments is an ongoing research topic.[3,4] This work involves software techniques, such as data flow analysis, as well as mathematical techniques.

Automated template decomposition is an important consideration in the construction of an image processing algorithm development environment. The purpose of template decomposition is to increase the computational efficiency of image processing algorithms, either by reducing the number of arithmetic computations in an algorithm or by restructuring the algorithms to match the structure of a special purpose image processing architecture. Much of the front-end processing in a typical aerospace image processing application algorithm involves regular, translation invariant computations at each point in an image. Such computations often are performed over a "window" around each point in an image. For this reason, many special purpose architectures for image processing have been based on parallel, parallel-pipeline, or pipeline structures which provide means for performing computations over a small window extremely quickly. For example, the PIPE is designed to grab a 3x3 window and perform neighborhood operations such as linear convolutions, while the CYTO-HSS performs a variety of morphological and other operations over a 3x3 window.[5]

The hardware limits the size of the window that can be processed directly. If processing is required over a larger window, then the computation must be decomposed to fit the hardware restriction. Some algorithms exist for decomposing template operations, and others are proposed or under development, such as the technique to be described in this paper. Many involve fairly complex mathematical operations or heuristic strategies. Algorithm designers working in computer vision and image processing often do not have the expertise, the time, or the desire to perform such decompositions.

The image algebra offers a unified representation of window operations in terms of generalized templates. Thus, the problem of restructuring computations over a window to meet hardware restrictions can be cast in terms of reorganizing template operations in the image algebra. In an automated system, the algorithm designer would simply have to define a window operation via a generalized template and the system would automatically decompose the template into a form compatible with the computational environment. To implement such a system, a variety of decomposition criteria and methods are required in order to accommodate different hardware configurations and types of operations.

In this paper, we focus on one particular aspect of this problem--the gray scale morphological template decomposition problem. The morphological template decomposition problem is that of decomposing templates with respect to the operations of mathematical morphology. Mathematical morphology, which is sometimes called image algebra, is a mathematical system based on the operations of erosion and dilation. Algebraically, these operations can be thought of as nonlinear convolutions. The origins of mathematical morphology can be traced to the work of Minkowski at the turn of the century.[6] More recently, several groups of researchers have demonstrated that mathematical morphology can be a useful tool in image processing environments providing an alternative to some of the linear techniques that have been so successful in one-dimensional signal processing.[7,8,9,10,11,12,13,14] Indeed, the use of mathematical morphology constituted one of the first uses of an algebraic structure for image processing. The algebraic system defined by mathematical morphology is included in our image algebra as a subalgebra.

As mentioned previously, several special purpose architectures have the capability of performing morphological operations over a 3x3 neighborhood extremely quickly. For example, the CYTO-HSS can perform 3x3 neighborhood operations at the rate of 10 million pixels per second.[5] Standard morphological operators requiring larger supports have been decomposed by hand so that they can be run on such machines. This procedure works well for regularly shaped morphological filters. However, for more complicated morphological filtering operations, hand decomposition is too difficult for most algorithm developers to consider. The goal of gray scale morphological template decomposition is to provide a tool for algorithm developers that would automatically decompose gray scale morphological operations into morphological operations requiring only a given support (often 3x3).

In the last decade, several authors have considered the problem of decomposition of linear operators.[3,15,16,17,18,19,20,21,22] The linear problem is difficult; it is equivalent to factoring polynomials (for FIR filters) or rational functions (for IIR filters). The morphological problem, especially the gray scale problem, has only been considered more recently. Zhuang and Haralick have developed a method for decomposing binary morphological templates based on a constrained search technique.[23] Ritter and Li have also developed some methods for specific classes of templates.[24] At this time, we do not know of any general techniques for solving the gray scale morphological template decomposition problem.

2. IMAGE ALGEBRA BACKGROUND

The fundamental definitions of the image algebra are readily available in the literature. In fact, they are included in other papers in the same session of this conference. Therefore, we do not include them here. In this paper, we shall focus our attention on images and templates defined on a single, two-dimensional coordinate set X. The templates that we consider are all translation invariant. That is, their configuration and weights are the same for every point in the coordinate set X. The gray scale morphological operation is expressed in the image algebra using these particular types of templates as $r = s \boxed{\vee} t$. For example, if s and t are

1	2	3
2	1	5
3	5	1

2	1	2
1	2	3
1	0	2

then r is

3	7	6	7	5
6	6	7	8	6
5	6	7	7	5
5	5	6	5	4
4	3	5	4	3

This is an example of gray scale morphological dilation. It is similar to linear correlation except that at each point one adds the weights of the overlapping regions and then takes the maximum.

It can be seen from the example how two 3x3 templates can be composed to generate a 5x5 template. Template decomposition works in the reverse order. The 3x3 morphological template decomposition problem is the following: Given a translation invariant template t on X, find a set of 3x3 templates $s_1, s_2, \ldots, s_k$ such that

$$t = s_1 \boxed{\vee} s_2 \boxed{\vee} \ldots \boxed{\vee} s_k \quad .$$

This is a special case of the local decomposition problem. It is sometimes called the strong decomposition problem. By contrast, there is a weak formulation that involves pointwise operations.

3. LINEAR PROGRAMMING FORMULATION.

In this section, we demonstrate how the morphological decomposition problem can be formulated as a linear programming problem. We focus our attention on the dilation operation, that is, on the $\boxed{\vee}$ operation for translation invariant templates. Let t be a given translation invariant template with rectangular support of size m x

n. We want to find translation invariant templates x and y with rectangular supports, both smaller than that of t, such that $t = x \boxed{\vee} y$. Let us consider first the case the t is a 1 x 5 template and x and y both 1 x 3 templates to be determined as shown below for concreteness. We use the subscript notation for indexing the weights in the template.

$$\boxed{t_1}\boxed{t_2}\boxed{t_3}\boxed{t_4}\boxed{t_5} \;=\; \boxed{x_1}\boxed{x_2}\boxed{x_3} \;\boxed{\vee}\; \boxed{y_1}\boxed{y_2}\boxed{y_3}$$

Figure 1. A simple case of template decomposition. t is known; x and y are unknown.

The template equation $t = x \boxed{\vee} y$ yields the following system of nonlinear equations:

$$\begin{aligned}
t_1 &= x_3 + y_1 \\
t_2 &= \boxed{\vee}\,[x_2 + y_1,\ x_3 + y_2] \\
t_3 &= \boxed{\vee}\,[x_1 + y_1,\ x_2 + y_2,\ x_3 + y_3] \\
t_4 &= \boxed{\vee}\,[x_1 + y_2,\ x_2 + y_3] \\
t_5 &= x_1 + y_3 \qquad .
\end{aligned} \tag{1}$$

These nonlinear equations in turn lead to consideration of the following linear system of inequalities:

$$\begin{array}{cccccccl}
 & & x_3 & + y_1 & & & & \le t_1 \\
\hline
 & x_2 & & + y_1 & & & & \le t_2 \\
 & & x_3 & & + y_2 & & & \le t_2 \\
\hline
x_1 & & & + y_1 & & & & \le t_3 \\
 & x_2 & & & + y_2 & & & \le t_3 \\
 & & x_3 & & & + y_3 & & \le t_3 \\
\hline
x_1 & & & & + y_2 & & & \le t_4 \\
 & x_2 & & & & + y_3 & & \le t_4 \\
\hline
x_1 & & & & & + y_3 & & \le t_5
\end{array} \tag{2}$$

which, when rearranged, yields

$$\begin{array}{cccccccl}
x_1 & & & + y_1 & & & & \le t_3 \\
x_1 & & & & + y_2 & & & \le t_4 \\
x_1 & & & & & + y_3 & & \le t_5 \\
 & x_2 & & + y_1 & & & & \le t_2 \\
 & x_2 & & & + y_2 & & & \le t_3 \\
 & x_2 & & & & + y_3 & & \le t_4 \\
 & & x_3 & + y_1 & & & & \le t_1 \\
 & & x_3 & & + y_2 & & & \le t_2 \\
 & & x_3 & & & + y_3 & & \le t_3
\end{array} \tag{3}$$

Thus, the problem of decomposing the template t can be formulated as finding a particular type of solution for a structured system of linear inequalities. This idea is completely general; the problem of decomposing any template with rectangular support with respect to morphological operations can be formulated similarly. If we divide the inequalities in system (2) into groups by requiring that two inequalities are in the same group if and only if they have the same right hand side for every template t, then our solution must have the property that at least one inequality in each group must be satisfied as an equality. We shall refer to this requirement as the group equality restriction.

There are many possible ways to solve this problem. For example, one could try all possible linear systems of equations constructed by taking one inequality from each group and treating it as an equality. This approach would work but it would be impractical.

Another approach is to formulate the problem as a linear programming problem, that is, as a problem of maximizing or minimizing a linear function, the objective function, subject to a system of linear constraints. Thus, to solve as a linear programming problem, we need to define an objective function to go with our system of linear constraints. The coefficients of the objective function are therefore parameters that must be chosen.

As an example, let t be the template

$$t = \begin{array}{|c|c|c|c|c|} \hline 6 & 5 & 4 & 3 & 2 \\ \hline \end{array} \quad .$$

We can solve the problem $t = x \boxed{\vee} y$ where x and y are each 1 x 3 by solving the linear program

$$\text{Maximize} \quad x_1 + x_2 + x_3 + y_1 + y_2 + y_3 \tag{4}$$

subject to the constraints given by (3) with the appropriate values of t plugged in. We obtain the solution

$$x = \begin{array}{|c|c|c|} \hline 0 & 1 & 2 \\ \hline \end{array} \quad \text{and} \quad y = \begin{array}{|c|c|c|} \hline 4 & 3 & 2 \\ \hline \end{array} \quad .$$

This objective function does not work for all decomposable templates t, however. Choice of an appropriate objective function is therefore an important problem that we shall discuss again later in the paper.

Another issue in using a linear programming approach for template decomposition is the size of the problem. The number of linear inequalities grows very rapidly as a function of the size of the template t. In fact, if t is a 5 x 5 template and x and y are 3 x 3 templates to be determined (the smallest two-dimensional decomposition problem), then the number of linear inequalities in the system is 81. In general, if t is 2k+1 x 2k+1 and x is 3 x 3, then there are $9(2k-1)^2$ linear inequalities. At first glance this large number of constraints might appear to rule out using the linear programming approach as a practical tool for template decomposition. However, the fact that the inequalities are extremely structured allows us to make very significant reductions in the size of the problem.

Furthermore, by considering the dual of the linear programming problem that results from direct formulation of the morphological template decomposition problem, we find that not only do we achieve a reduction in problem size, but we are able to cast the problem in terms of a well known setting--the transportation problem.

4. THE DUAL FORMULATION AS A TRANSPORTATION PROBLEM.

In this section, we show how the original, or primal, formulation of the morphological template decomposition problem can be transformed into a classical transportation problem by taking the dual.* Again, for concreteness, let us consider the case that t is 1x5 and x and y are both 1x3. The primal problem is then

$$\text{Maximize} \quad c_1 x_1 + c_2 x_2 + c_3 x_3 + c_4 y_1 + c_5 y_2 + c_6 y_3 \tag{5}$$

subject to the constraints given by (3), where the t's are given, the x's and y's are unknown, and the c's are parameters to be chosen. We have placed no nonnegativity constraints on the x and y variables. This implies, by linear programming theory, that the constraints in the dual problem will be equality constraints. If we choose to place the restriction that the x and y variables must be nonnegative, then the constraints in the dual will be $\geq$ constraints. From the standpoint of template decomposition considerations, the nonnegativity constraint is not necessary since, even if nonnegative weights are required, the results can be "shifted" after the solution is found because of the algebraic relationship that states that dilating by a structuring element and then adding a constant to every weight in the result is the same as adding the constant to every weight of the structuring element and then dilating. That is, if s and t are translation invariant and c is a constant, then

$$(s \boxed{\vee} t) + c = (s + c) \boxed{\vee} t \quad . \tag{6}$$

The dual of the above maximization problem (5) is:

$$\begin{array}{l} \text{Minimize} \\ t_3 z_1 + t_4 z_2 + t_5 z_3 + t_2 z_4 + t_3 z_5 + t_4 z_6 + t_1 z_7 + t_2 z_8 + t_3 z_9 \end{array} \tag{7}$$

Subject to

$$\begin{array}{ccccccccccccccccccc}
z_1 & + & z_2 & + & z_3 & & & & & & & & & & & & & = & c_1 \\
 & & & & & & z_4 & + & z_5 & + & z_6 & & & & & & & = & c_2 \\
 & & & & & & & & & & & & z_7 & + & z_8 & + & z_9 & = & c_3 \\
z_1 & & & & & + & z_4 & & & & & + & z_7 & & & & & = & c_4 \\
 & & z_2 & & & & & + & z_5 & & & & & + & z_8 & & & = & c_5 \\
 & & & & z_3 & & & & & + & z_6 & & & & & + & z_9 & = & c_6
\end{array}$$

$$z_1,\ z_2,\ z_3,\ z_4,\ z_5,\ z_6,\ z_7,\ z_8,\ z_9 \geq 0 \quad .$$

---

*For more information concerning the dual, one can consult a linear programming or operations research text. The fundamental theorem of duality theory is that the optimal solution to the dual problem yields a complementary optimal solution to the primal problem.

Any linear programming problem of this form is called a transportation problem. There are several extremely efficient methods for solving transportation problems that exploit the special structure of the problem.[25] In fact, several categories of linear programming problems are often recast as transportation problems with more variables and constraints to make use of the more efficient techniques. Rather than use a simplex tableau to solve the problem as in the case of a general linear programming problem, another tableau, called the transportation tableau is used. In the case of template decomposition, the transportation tableau has far fewer entries than the corresponding simplex tableau for the primal problem. For example, the transportation tableau for the dual problem given by (7) is:

t_3	t_4	t_5
t_2	t_3	t_4
t_1	t_2	t_3

The (i,j) element of the box is the coefficient of the variable z_k in the dual objective function where $k = 3(i-1) + j$. In the case that t is 5x5 and x and y are 3x3, the transportation tableau is a 9x9 array:

t_{33}	t_{34}	t_{35}	t_{43}	t_{44}	t_{45}	t_{53}	t_{54}	t_{55}
t_{32}	t_{33}	t_{34}	t_{42}	t_{43}	t_{44}	t_{52}	t_{53}	t_{54}
t_{31}	t_{32}	t_{33}	t_{41}	t_{42}	t_{43}	t_{51}	t_{52}	t_{53}
t_{23}	t_{24}	t_{25}	t_{33}	t_{34}	t_{35}	t_{43}	t_{44}	t_{45}
t_{22}	t_{23}	t_{24}	t_{32}	t_{33}	t_{34}	t_{42}	t_{43}	t_{44}
t_{21}	t_{22}	t_{23}	t_{31}	t_{32}	t_{33}	t_{41}	t_{42}	t_{43}
t_{13}	t_{14}	t_{15}	t_{23}	t_{24}	t_{25}	t_{33}	t_{34}	t_{35}
t_{12}	t_{13}	t_{14}	t_{22}	t_{23}	t_{24}	t_{32}	t_{33}	t_{34}
t_{11}	t_{12}	t_{13}	t_{21}	t_{22}	t_{23}	t_{31}	t_{32}	t_{33}

This is the data structure used to solve the transportation problem with the streamlined approach. Note that because of the special nature of template decomposition, the tableau has the structure of a block Toeplitz matrix with Toeplitz blocks. The Toeplitz structure could offer further savings over the conventional approaches; although there are 81 elements in the tableau, they are obtained directly from the 25 weights of the template. By contrast, the simplex tableau required to solve the primal problem has 81 rows and

99 columns (including the necessary slack variables). The special structure and available algorithms for the transportation problem indicate that it is the preferred formulation for template decomposition.

The group equality restriction must still be taken into account, however. To understand how the group equality restriction manifests itself in the dual formulation it is helpful to use the idea of complementary slackness. To solve a linear programming problem such as (4), one adds a variable, called a slack variable, to each of the inequalities in the constraint set. When the optimal solution is reached, the inequality will be satisfied as an equality if and only if the slack variable that was added equals zero. Thus, the group equality restriction can be restated as the requirement that in the optimal solution of the linear programming problem used to solve the template decomposition problem, the value of at least one slack variable in each group must be zero. More specifically, we require that at least one slack variable from each group must be a nonbasic variable in the solution (except in the degenerate case).

The theory of complementary slackness states that there is a one-to-one correspondence between slack variables in the primal problem and decision variables (the z's in (7) above) in the dual problem and that basic variables in one problem correspond to nonbasic variables in the other problem. Thus, if we divide the dual variables into groups by defining two variables to be in the same group if and only if they have the same objective function coefficient for every template t, then the group equality restriction can be interpreted as requiring that a solution to the problem must have the property that at least one variable from each group be a basic variable.

5. SOLUTION APPROACHES

The dual formulation of our problem provides us with a reasonably sized problem that is expressed in terms of a special class of linear programming problems. In order to find a solution, we must still modify traditional algorithms in order to satisfy the group equality restriction. The ideal case would be that we could choose a set of parameters (the coefficients of the objective function in the primal problem or the right-hand-sides in the dual problem) that would yield an optimal solution that satisfies the restriction in all cases. Although this may be possible, it seems unlikely.

One approach that we believe to be promising is to use a type of sensitivity analysis. The idea is as follows: Choose an initial set of coefficients using some criteria. Solve the linear programming problem using that set of coefficients and check to see if the solution satisfies the group equality restriction. If so, we are done. If not, then choose another set of coefficients by increasing the coefficient of a variable that appears in an inequality from a group not satisfying the group equality restriction. Solve the linear programming problem again and continue the procedure until a stopping criteria is reached or a solution is found.

We have performed some initial experiments using this approach and have had success. For example, we tried to decompose the template

$$t = \boxed{7}\boxed{9}\boxed{7}\boxed{8}\boxed{4}$$

using the initial objective function (for the primal problem)

$$\text{Maximize} \quad x_1 + x_2 + x_3 + y_1 + y_2 + y_3 \quad . \tag{7}$$

The solution to the problem did not satisfy the group equality restriction. We then went through the above procedure which resulted in the use of the objective function

$$\text{Maximize} \quad 5x_1 + x_2 + 5x_3 + 3y_1 + 4y_2 + 5y_3 \tag{8}$$

which yielded the solution

$$x = \boxed{1}\boxed{0}\boxed{2} \qquad y = \boxed{5}\boxed{7}\boxed{3} \quad .$$

It may be that there is some method for choosing the initial coefficients based on the characteristics of the template t that will start the procedure "close" to a solution, or that there is another methodology that can be used. These are questions that we intend to pursue.

6. CONCLUSIONS

Image algebra provides a high-level representation of image processing operations. In order for image algebra to provide a basis for an algorithm development environment, it is highly desirable to have a tool for automatically decomposing template operations.

We have shown how the gray scale morphological template decomposition problem can be formulated as a linear programming problem with an additional restriction, the group equality restriction. Furthermore, we have shown that the dual of the initial formulation is an instance of a special type of linear programming problem called the transportation problem. This formulation is practical since several efficient algorithms exist for solving transportation problems. Current work involves extending conventional solution methods to accommodate the group equality restriction.

7. REFERENCES

1. G.X. Ritter, J.N. Wilson, J.L. Davidson, "Image Algebra: An Overview," UF-CIS Technical Report TR-88-05, Dept. of Comp. and Info. Sci., Univ. of Florida, Gainesville, FL (May 1988).

2. G.X. Ritter, J.N. Wilson, J.L. Davidson, "Image Algebra Application to Image Measurement and Feature Extraction," Proc. SPIE OELASE, Los Angeles, CA (Jan. 1989) (to appear).

3. G.X. Ritter and P.D. Gader, "Image Algebra Techniques for Parallel Image Processing," Journal of Parallel and Distributed

Computing 4(5), 7-44 (March 1987).
4. J.N. Wilson and L.A. White, "A New Approach to Program Optimization," preprint.
5. R.M. Lougheed, "Architectures and Algorithms for Digital Image Processing II," Proc. SPIE 534, 22-33 (1985).
6. H. Minkowski, Gessammelte Abhandlungen, Teubner, Leipzig-Berlin (1911).
7. S.R. Sternberg, "Grayscale Morphology", CVGIP 35, 333-355 (Sept. 1986).
8. J. Serra, Image Analysis and Mathematical Morphology, Academic Press, London (1982).
9. P.A. Maragos and R.W. Schafer, "Morphological Skeleton and Representation and Coding of Binary Images," IEEE Trans. Acoustics, Speech, and Signal Processing (34)5, 1228-1244 (October 1986).
10. R.M. Haralick, S.R. Sternberg, and X. Zhuang, "Image Analysis Using Mathematical Morphology," IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI-9(4),532-550 (July 1987).
11. J. Serra, "Introduction to Mathematical Morphology," CVGIP 35, 283-305 (Sept. 1986).
12. C.R. Giardina and E.R. Dougherty, Morphological Methods in Image and Signal Processing, Prentice-Hall (1988).
13. T. Crimmins and W. Brown, "Image Algebra and Automatic Shape Recognition," IEEE Trans. on Aerospace and Electronic Systems AES-21(1), 60-69 (Jan. 1985).
14. D. McCubbrey, "Real-Time Bin Picking Using Cellular Neighborhood Operations," SME Applied Machine Vision Conference, Cleveland, Ohio (1982).
15. D.P. O'Leary, "Some Algorithms for Approximating Convolutions," CVGIP 41, 333-345 (1988).
16. A.N. Venetsanopoulos and B.G. Mertzios, "A Decomposition Theorem and Its Implications to the Design and Realization of Two-Dimensional Filters," IEEE Trans. Acoustics, Speech, and Signal Processing ASSP-33(6), 1562-1575 (December 1985).
17. W.K. Pratt, "Intelligent Image Processing Display Terminal," Proc. SPIE 199, 189-194 (1979).
18. J.-F. Abramatic and O.D. Faugeras, "Sequential Convolution Techniques for Image Filtering," IEEE Trans. Acoustics, Speech, and Signal Processing, ASSP-30(1), 1-10 (Feb. 1982).
19. P.D. Gader, "Necessary and Sufficient Conditions for the Existence of Local Matrix Decompositions," SIAM J. Matrix Anal. Appl. 9(3), 305-313.
20. P.D. Gader, "Tridiagonal Factorizations of Fourier Matrices and Applications to Parallel Computations of Discrete Fourier Transforms," Linear Alg. Appl. 102, 169-209 (April 1988).
21. P.D. Gader, "Bidiagonal Factorization of Fourier Matrices and Systolic Algorithms for Computing Discrete Fourier Transforms," IEEE Trans. Acoustics, Speech, and Signal Processing (August 1989) (to appear).
22. P.D. Gader, "Displacement Operator Based Decompositions of Matrices Using Circulants or Other Group Matrices," Linear Alg. Appl., submitted for publication.
23. X. Zhuang and R.M. Haralick, "Morphological Structuring Element Decomposition," CVGIP 35, 370-382 (1986).

24. G.X. Ritter, personal correspondence.
25. F.S. Hillier and G.J. Lieberman, Introduction to Operations Research, pp. 118-146, Holden-Day, San Francisco (1980).