Graphics and Image Processing
W. Newman Editor

# Reconstruction of Pictures from Their Projections

Richard Gordon* and Gabor T. Herman†
State University of New York at Buffalo

**There are situations in the natural sciences and medicine (e.g. in electron microscopy and X-ray photography) in which it is desirable to estimate the gray levels of a digital picture at the individual points from the sums of the gray levels along straight lines (projections) at a few angles. Usually, in such situations, the picture is far from determined and the problem is to find the "most representative" picture. Three algorithms are described (all using Monte Carlo methods) which were designed to solve this problem. The algorithms are applicable in a large and varied number of fields. The most important uses may be the reconstruction of possibly asymmetric particles from electron micrographs and three-dimensional X-ray analysis.**

**Key Words and Phrases: approximation, biomedical image processing, efficient encoding, image processing, linear programming, mathematical programming, Monte Carlo techniques, optimization, picture compression, picture description, picture processing, stereology, X-ray analysis**

**CR Categories: 3.12, 3.13, 3.15, 3.17, 3.63, 5.41, 5.6**



* Center for Theoretical Biology. † Department of Computer Science, SUNY, 4226 Ridge Lea Road, Amherst, NY 14226.
This project was supported in part by NASA Grant NGR-33-015-016 to the Center for Theoretical Biology, SUNY/Buffalo, and by NSF Grants GJ596 and GJ998.

## 1. Introduction

We have been investigating a picture processing problem originally arising from a problem in electron microscopy, but having applications in a wide range of fields, both within and outside biology. In particular, we have discovered a new set of algorithms for the reconstruction of three-dimensional objects from a set of transmission photographs taken through a translucent object at a few angles. As will be seen, the problem is reducible to the two-dimensional one of reconstructing a given planar section of the object. Such algorithms are of great importance in biology, for they allow us to discover the structure of macromolecules and macromolecular assemblies, such as ribosomes [3].

With Rosenfeld [21, p. 147] we define "a *picture* as being a real-valued, nonnegative function of two real variables; the value of this function at a point will be called the *gray level* of the picture at the point." We define a *projection* as an ordered set of values, each value being the integral or sum of the gray levels along one of a set of parallel *rays*, which are bands of finite width drawn across the picture at a known angle. *Our problem is to reconstruct a picture from a finite subset of its projections taken at distinct angles.*

We attempted solution for digital pictures only. A *digital* picture is one which is defined by specifying an $n \times n$ matrix of gray levels. An element of the matrix will be called a *picture element*. A digital picture is said to be *quantized* if each element can take on only finitely many values. Rosenfeld [22, Ch. 3] discusses digitization (sampling) and quantization and gives justifications for them. In general, a ray is now a subset of picture elements. Each picture element contributes to one and only one of the rays of a given projection. A simple example

of a projection is found in the ordered set of row sums $(r_1, \ldots, r_n)$, where

$$r_i = \sum_{j=1}^{n} \rho_{i,j}, \quad i = 1, \ldots, n, \tag{1}$$

and $\rho_{i,j}$ is the gray level of picture element $(i, j)$, i.e. the element in the $i$th row and $j$th column. The $n$ rays of a projection at an arbitrary angle are defined below.

For such pictures, a projection can be considered to be a set of $n$ linear equations in the $n^2$ unknowns $\{\rho_{i,j}\}$. If we take $m$ distinct projections, we have $mn$ simultaneous linear equations in $n^2$ unknowns. If $m < n$, there will generally be more than one solution, i.e. more than one picture with exactly the same projections. If one is satisfied with solutions accurate to a given number of digits, then all of the projections may be multiplied by an appropriate constant so that the gray levels become integers. All of our algorithms, although somewhat generalizable, were implemented in integers. In particular, we shall assume that the admissible gray levels are $0, 1, \ldots, l - 1$ (a quantized picture).

If we were to treat each of the rays as a linear equation in the $n^2$ unknowns $\rho_{i,j}$, and if we have $m$ projections, each with $n$ rays, then the matrix of these equations would be an $mn \times n^2$ matrix, requiring $mn^3$ locations in storage. If $n = 1000$ and $m = 100$, this would require $10^{11}$ locations, which is far from reasonable. Even in the case of $n = 100$, $m = 10$, the number of locations needed is on the order of $10^7$, which is still too large for the core memory of most present-day computers. Thus direct treatment of the simultaneous linear equations, applying techniques which are usual in linear programming (manipulation of the matrix of coefficients), does not appear to be feasible to implement. An entirely different approach seems to be warranted. Our three algorithms require storage proportional to $n^2$ only.

As an indication of the success of our approach, the reader is referred to Figures 5(g) and 5(h). Figure 5(h) is a 49 × 49 digitized picture (2,401 unknowns), and Figure 5(g) indicates the result of our reconstruction using eight projections (488 linear equations, some linearly dependent on others). The next three sections give the details of the three algorithms which have been used to obtain the reconstructions of Figure 5. Further details and discussion are given in an internal publication [14], which is available upon request. For a summary of our results and applications of our techniques, see Sections 5 and 6.

## 2. The First Algorithm

Since we will be dealing with $n \times n$ digitized pictures, it will usually be convenient to define the rays of a given projection as subsets of exactly $n$ of the picture elements $\rho_{i,j}$. Let $\theta$ be the angle associated with a given projection, $-90° < \theta \leq 90°$. We draw a line at angle $\theta$ to the horizontal through the position of the first matrix element and calculate the distance of each point $(i, j)$ from this line (Figure 1). Points above the line, in case $\theta$ is negative, are considered to be at a negative distance. The picture elements are then sorted according to these distances, and blocked into $n$ groups of $n$. The $k$th group is defined as ray $k$ of the projection at angle $\theta$ (Figure 2).

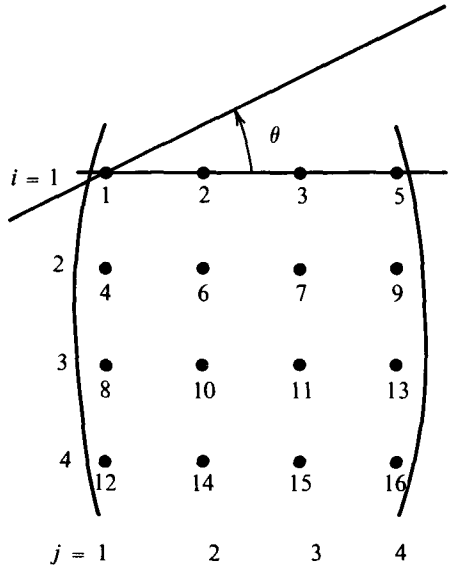
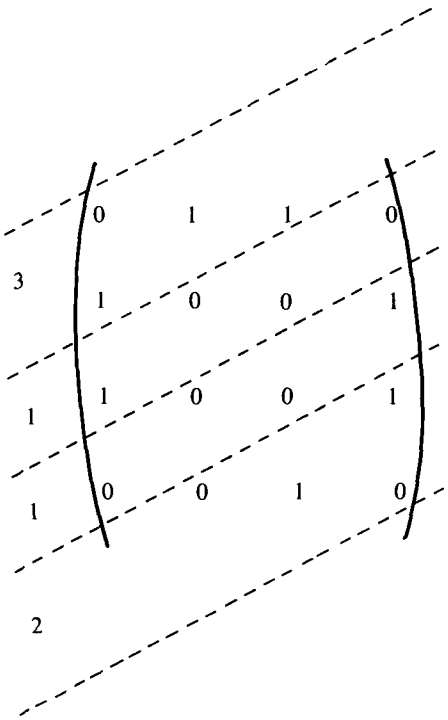



Fig. 1. Example of the sorting of the picture elements for Algorithm 1 with $\theta = \tan^{-1}(\frac{1}{2})$ for a 4 × 4 picture. The picture elements are labeled according to their distance from the line through element (1,1) at angle $\theta$ from the horizontal.

Algorithm 1 begins with an initially blank $n \times n$ matrix ($\rho_{i,j} = 0$). Each ray of the first projection is considered in turn. Bits (equal to 1 gray level unit each) are randomly added to the picture elements of the ray until their sum is correct. Next we scan the rays of the second projection. The sum of the picture elements of a given ray may be too high or too low, in which case bits are either subtracted or added at random from its picture elements. (Bits are not added to or subtracted from picture elements whose gray level is already $l - 1$ or 0, respectively, but otherwise each picture element is equally likely to have a bit added to it.) When all the rays of a projection are done, the total number of bits in the matrix is restored to the proper number. The third etc., projections are treated similarly.

Fig. 2. The four rays of the projection at angle $\theta = \tan^{-1}(\frac{1}{2})$ for a $4 \times 4$ picture, as in Figure 1. The values of the rays (sums of gray levels along the rays) are indicated on the left.




An unfortunate consequence of our first algorithm is that whenever a new projection is satisfied, the previous ones may be disturbed. The algorithm seems to work because the changes introduced into the picture are unbiased random noise, from the point of view of the old projections. The noise seems to disappear when we take the average picture of many independent individual reconstructions. So that no projection has more noise than any other, we consider them in random order for each individual reconstruction.

In summary, Algorithm 1 creates random individual reconstructions by satisfying (in a random order) each of the given projections. The average of several individual reconstructions was found to be a reasonably smooth picture more or less satisfying all the projections.

## 3. The Second Algorithm

Our second algorithm satisfies all of the projections *exactly* in each individual reconstruction. We have only implemented this up to four projections, so that the first algorithm still had to be retained. We have restricted our FORTRAN implementation to four particular projections: (1) vertical (90°); (2) horizontal (0°); (3) 45° diagonal; and (4) −45° diagonal. For the diagonal projections a ray is redefined as a set of picture elements on a line parallel to a principal diagonal of the matrix. Thus the diagonal projections have $2n - 1$ rays each, and each ray consists of 1 to $n$ picture elements. The rays for the horizontal and vertical projections are just the rows and columns of the matrix, so that they have $n$ rays each with $n$ picture elements in each ray, as in Algorithm 1.

The rays of the vertical projection are satisfied by adding bits at random, as was the first projection in Algorithm 1. Then we have

$$c_j = \sum_i \rho_{i,j} \tag{2}$$

where $c_j$ is the value of the $j$th ray of the vertical projection.

Suppose $r_1, \ldots, r_{i-1}$ (the horizontal row sums, eq. (1)), are already satisfied, and $i < n$. If the reconstruction so far is such that

$$r_i < \sum_j \rho_{i,j}, \tag{3}$$

then we randomly select a $j$, $1 \leq j \leq n$, and add 1 to $\rho_{i,j}$. But this causes an error in the vertical column sum:

$$\sum_i \rho_{i,j} = c_j + 1. \tag{4}$$

To restore the first projection, we randomly select $u > i$ $(u \leq n)$ and subtract 1 from $\rho_{u,j}$. The first projection is restored and the first $i - 1$ rays of the second projection are left undisturbed. We repeat the process until ray $i$ is satisfied, and then move on to $i + 1$. The case

$$r_i > \sum_j \rho_{i,j} \tag{5}$$

is analogous.

Since

$$\sum_{i,j} \rho_{i,j} = \sum_j c_j = \sum_i r_i \tag{6}$$

is fixed and was already correct after we satisfied the first projection, when $r_1, \ldots, r_{n-1}$ are satisfied, $r_n$ will be automatically satisfied.

What if a picture element we have selected for subtraction of a bit is already 0? We do not allow a negative gray level. Similarly, we may want to add to a picture element whose value is already the maximum, $l - 1$. In such cases we make at most $f(n)$ independent random attempts of the kind described above, where $f(n)$ is a monotonically increasing function of $n$. It is possible that even after $f(n)$ attempts we cannot satisfy this ray

of the second projection. For instance, if the original matrix was

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

(with only two gray levels allowed, $l - 1 = 1$), then the first projection may have provided us with

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

and there is no way of manipulating the second and third rows to satisfy the second projection. In such cases we try to satisfy the second projection in reverse order, i.e. starting with its last ray. We call this *flipping*, since our program flips the matrix rather than reverse the indices. We may have to flip the matrix several times. In practice we never found any need to flip at this stage of the algorithm. Note also that one flip will allow precise reconstruction of the matrix of the example above.

We found situations where the method of flipping will not provide us with a solution. Such examples had to be artificially created and we never came across one in practice, but the algorithm must check for this possibility.

With projections $\{c_j\}$ and $\{r_i\}$ satisfied, we now wish to satisfy the 45° diagonal

$$d_k = \sum_{i+j=k+1} \rho_{i,j}, \quad k = 1, \ldots, 2n - 1. \tag{7}$$

Suppose that $d_1, \ldots, d_{k-1}$ are already satisfied and that $k < 2n - 2$. If the reconstruction so far is such that

$$d_k < \sum_{i+j=k+1} \rho_{i,j}, \tag{8}$$

then we randomly select a picture element $(i, j)$ on ray $k = i + j - 1$, and add 1 to $\rho_{i,j}$. But this disturbs both $c_j$ and $r_i$. To counteract this we randomly select a $u$ and a $v$ such that $i < u \leq n, j < v \leq n$, and subtract 1 from both $\rho_{i,v}$ and $\rho_{u,j}$. This restores $c_j$ and $r_i$ but disturbs $c_v$ and $r_u$. But by adding 1 to $\rho_{u,v}$, the first two projections are again satisfied (Figure 3(a)). We repeat this process until $d_k$ is satisfied. (If inequality (8) is reversed, the process is analogous.) We then move on to $d_{k+1}$, etc., until all of $d_1, \ldots, d_{2n-3}$ are satisfied. This implies that $d_{2n-2}$ and $d_{2n-1}$ will be automatically satisfied. A mathematical proof can be found in [14, Appen. II].

Rays of the $-45°$ diagonal should have values

$$e_k = \sum_{j-i=n-k} \rho_{i,j}, \quad k = 1, \ldots, 2n - 1. \tag{9}$$

To satisfy $e_k$ without disturbing $\{c_j\}$, $\{r_i\}$, and $\{d_k\}$, we use essentially the same technique, but now it is somewhat more complicated to restore previous projections. If

$$e_k < \sum_{j-i=n-k} \rho_{i,j} \tag{10}$$

we randomly select a $u$ and a $v$ such that $i < u < n$, $1 < v < j$, $u + j - v \leq n$, and $i + v - u \geq 1$. We add 1 to $\rho_{i,j}$, $\rho_{u+j-v,v}$, $\rho_{u,i+v-u}$, and subtract 1 from $\rho_{u,j}$, $\rho_{i,v}$, $\rho_{u+j-v,i+v-u}$. These six picture elements form a hexagonal cycle which will not disturb the three previous projections (Figure 3(b)).

In this fashion we satisfy $e_1, \ldots, e_{2n-5}$. In [14, Appen. II] we proved that $e_{2n-4}, \ldots, e_{2n-1}$ are automatically satisfied. We found that in all examples we tried, flipping handles problems caused by the bounds on the gray levels with three or four projections as well as with two projections.

The techniques described above for simultaneously satisfying a number of projections can be extended to any number of projections, provided only that all but one of them are taken at angles whose tangents are rational numbers. The details of this extension of Algorithm 2 are discussed in [14, Appen. III]. Figure 4 shows a suitable polygon, which preserves eight projections while changing the ninth.

## 4. The Third Algorithm

As will be discussed below, the average of an increasing number of individual reconstructions from Algorithms 1 and 2 causes the entropylike function

$$S = -\sum_{i=1}^{n} \sum_{j=1}^{n} \rho_{i,j} \ln \rho_{i,j} \tag{11}$$

to approximate a certain value asymptotically from below. (The sign in eq. (11) is the one usually used by information theorists. For a discussion of the nature of the sign of this function, see Brillouin [4, p. 161]. The sense in which the $\rho_{i,j}$ are proportional to probabilities in a probability distribution is discussed in [14, Appen. I].) This value is nearly (but not quite) the maximal value for $S$. It is shown in [14, Appen. II] that all solutions for four projections obtained by Algorithm 2 are accessible to one another by repeated applications of the following simple transformation.

A *double hexagonal transformation* of a matrix is a change in the values of its elements brought about in the following way. Take a ray of the fourth special projection (a $-45°$ diagonal). Choose two different points on it such that it is possible to draw from each a hexagon of the type described in the last section and Figure 3 (b). From each draw such a hexagon with $u = i + 1$ and $v = j - 1$. Add 1 to one of the points, subtract 1 from the other. Make appropriate changes on both hexagons to restore the first three projections. Clearly, the fourth projection will also remain undisturbed. (In getting from one solution to another we may on the way encounter picture elements outside the range 0 to $l - 1$.)

We attempted to make use of double hexagonal transformations to directly maximize the function $S$.

Consider ray $k$ of the fourth projection at $-45°$. Consider the small hexagonal cycle $(i, j)$, $(i + 1, j)$,

Fig. 3(a). The method of satisfying the third projection in Algorithm 2.

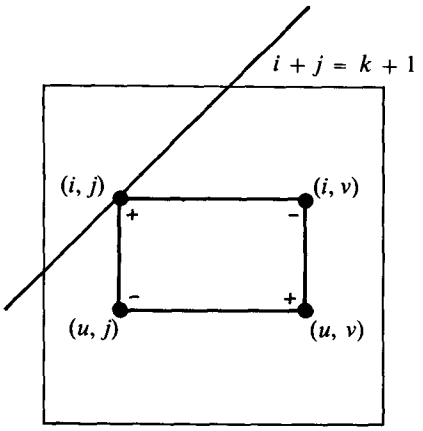



Fig. 3(b). The method of satisfying the fourth projection in Algorithm 2.

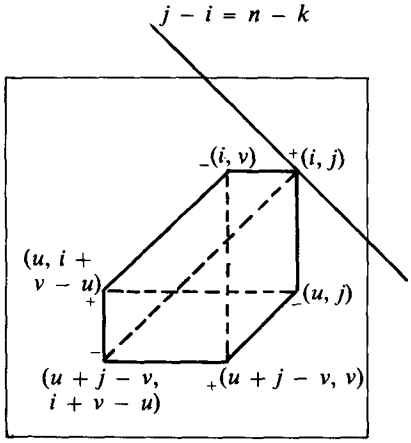
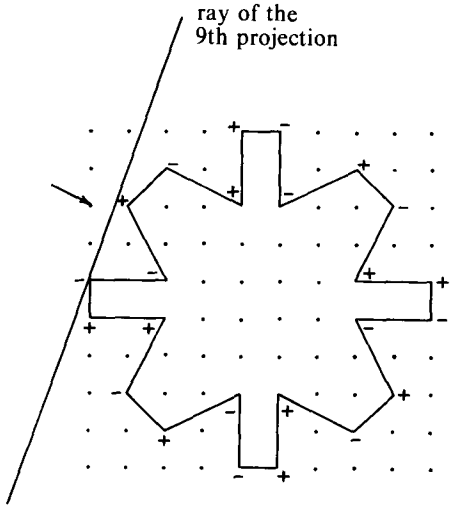



Fig. 4. A method of satisfying the ninth projection in Algorithm 2.




$(i+2, j-1)$, $(i+2, j-2)$, $(i+1, j-2)$, $(i, j-1)$, where $(i, j)$ is a picture element of ray $k$. We define the entropy of the cycle as

$$s_{i,j} = -\sum_{\text{small hexagon}} \rho_{i',j'} \ln \rho_{i',j'} \tag{12}$$

where the $(i', j')$ are the coordinates listed above.

Let $(i, j)$ and $(u, v)$ be two points on a ray at which it is possible to perform a double hexagonal transformation. If

$$s'_{i,j} + s'_{u,v} > s_{i,j} + s_{u,v} \tag{13}$$

where the primes denote the cycle entropies after the $\rho$'s have been changed, then the total value of $S$ for the picture will increase.

In our implementation of Algorithm 3, we scanned each picture element $(i, j)$ of each ray of the $-45°$ diagonal in turn, tested if its cycle entropy increased on adding or subtracting 1, and if so, then tried to find the next picture element $(u, v)$ of the ray such that with the complementary operation on it inequality (13) was satisfied. If there was none such, scanning just continued. After all the rays had been scanned, we calculated $S$. (We only allowed transformations which kept the gray levels in the permitted range.)

In practice we discovered that during successive iterations of Algorithm 3 the value of $S$ converged to a value far below the maximum. In order to get away from such a local maximum, we added some random noise to the picture, restored the four projections by passing the picture back to Algorithm 2, and then tried Algorithm 3 again. The noise decreased $S$, but then it generally climbed above the previous local maximum. When it did not, we tried the same operations on the (stored) previous maximum, but with twice as much noise. If $S$ improved, the noise for the next attempt was cut in half.

We shall compare the success of this direct search algorithm with the others in Section 5.

## 5. Implementation, Examples, Evaluation

All three algorithms have been implemented in FORTRAN, which was chosen because it does not have any obvious disadvantages over other high level languages. Also in its favor was the high level of support at the computing center where most of the work was carried out (as well as at most other installations) and the fact that it is the best known language among scientists, who are potential users of the program. (See Section 6 on applications.)

The algorithms have been tested on a large number of examples. Here we report on one of the least trivial ones, a picture of a little girl, Judy (Figure 5). For lack of other equipment, Judy's photograph was digitized by hand (using a simple densitometer) into a $49 \times 49$ matrix using 16 gray levels. (See Figure 5(h) for the original digitized version of the photograph.) A human

Fig. 5. Pictures of Judy.

(a) Average of 20 independent reconstructions with two projections (0°, 90°) using Algorithm 2.

(b) Picture with maximum entropy satisfying two projections (0°, 90°).

(c) Reconstruction with four projections (0°, 90°, ±45°) using Algorithm 3.

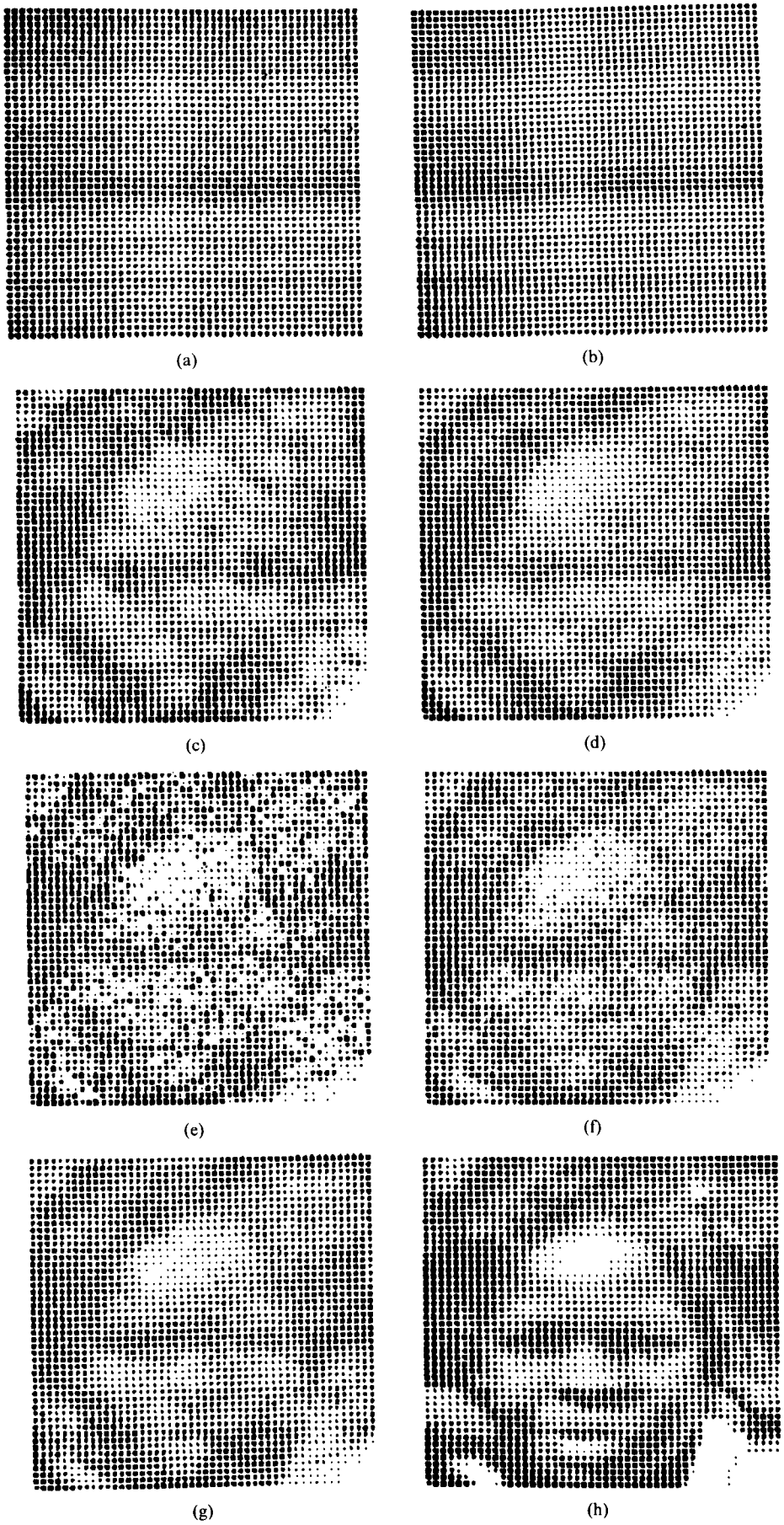
(d) Average of 20 independent reconstructions with four projections using Algorithm 2.

(e) Individual reconstruction with eight projections using Algorithm 1 (±22°, ±68°) and Algorithm 2 (0°, 90°, ±45°).

(f) Average of four independent reconstructions with eight projections using Algorithms 1 and 2.

(g) Average of 20 reconstructions with eight projections.

(h) Original picture of Judy.



face has been chosen, since it has a fairly complicated structure, which is, however, familiar to us all, and so it should be relatively easy to evaluate the success of our reconstruction. Because of the large number of white and dark patches on Judy's picture, some of them quite small, we felt that a successful reconstruction of this picture would demonstrate the validity of our algorithms.

Let us observe the most successful reconstruction (Figure 5(g)). Before objecting to its imperfections, the reader should note that Judy's picture has $49 \times 49 = 2{,}401$ points in it and the gray level had to be determined at each of these points from nothing but eight projections. Thus we had the problem of finding the value of 2,401 unknowns from 488 linear equations, some of which were linearly dependent on the others. It would be absolutely unreasonable to demand the reconstruction of the exact gray level at each point. Looking at the picture from a more structural point of view, we consider our reconstruction extremely successful. For instance, we have identified 19 regions of brightness in the original (some very small) and every one of these can clearly be detected in the reconstruction, although some of them are joined together.

Figures 5(e), 5(f), and 5(g) give a visual indication of how our method of averaging multiple individual reconstructions is working out in practice. The average of a large number of pictures is both smoother and nearer to the (smooth) original. Note in particular the great improvement by taking the average of only four individual reconstructions.

This basic aspect of our program is even better demonstrated by Figures 6 and 7. Figure 6 shows the root mean square distance between the original picture and reconstructions. The root mean square distance between pictures $\{\rho_{i,j}\}$ and $\{\rho'_{i,j}\}$ is defined to be

$$\delta = \left[ (1/n^2) \sum_{i=1}^{n} \sum_{j=1}^{n} (\rho_{i,j} - \rho'_{i,j})^2 \right]^{1/2} \tag{14}$$

and is considered to be a reasonable measure of the difference between two pictures (Rosenfeld [22]). We see that, for any number of projections, there is a point after which making extra individual reconstructions does not give a significant improvement in approximating the original. At this point the computation should be terminated. However, this criterion cannot be used in practice, since we do not know what the original picture is and so

Fig. 6. Distance $\delta$ of reconstructions from the original picture of Judy, versus the number of reconstructions averaged, for two, four, and eight projections.

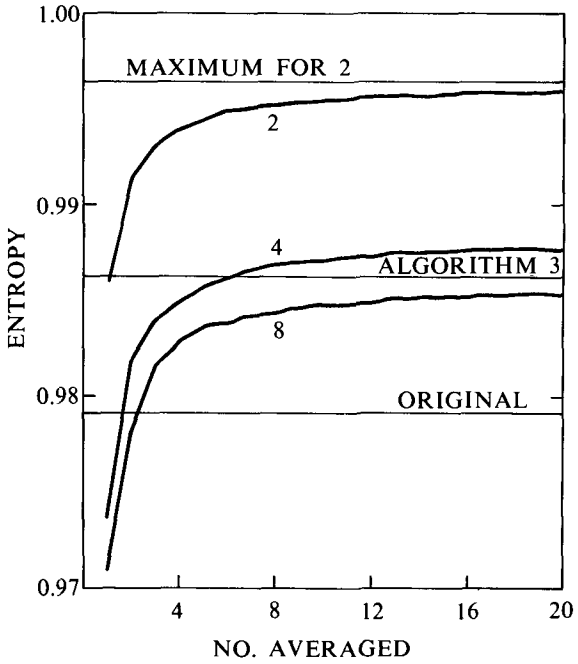
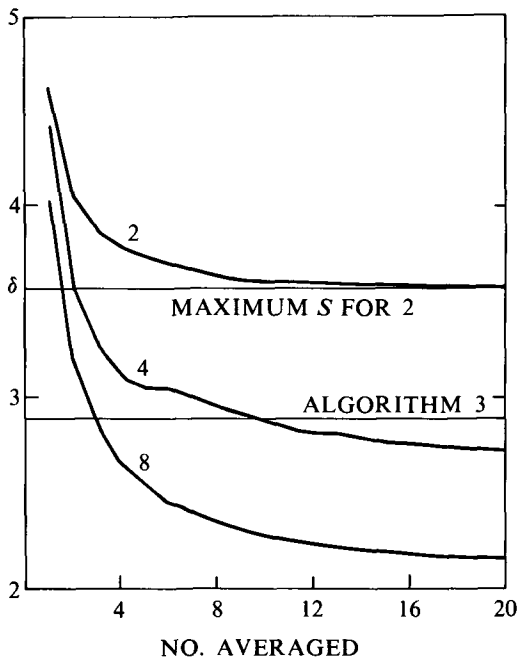



Fig. 7. Value of the normalized entropy function in reconstructions of Judy's picture, versus the number of reconstructions averaged.



we have no way of calculating the root mean square distance. However, by comparing Figures 6 and 7, we see that the root mean square distance from the original and the normalized entropy level out at about the same point. The normalized entropy $H$ is defined by

$$H = S/M \tag{15}$$

where

$$M = -\left(\sum_{i,j} \rho_{i,j}\right) \ln\left[(1/n^2) \sum_{i,j} \rho_{i,j}\right]; \tag{16}$$

i.e. $M$ is the maximum possible value of $S$ for the given total density. Thus $0 \leq H \leq 1$. The relation of $H$ and $S$ to the traditional notion of entropy of a picture is not entirely straightforward. A discussion is given in [14, Appen. I].

The value of $H$ can easily be calculated from the reconstruction (and this is done in our program). Thus we obtain a good heuristic guide as to when we have reached the point of diminishing returns.

In Figure 7, the line marked "maximum for 2" shows the maximum normalized entropy which a picture can have if it satisfies the horizontal and vertical projections of Judy. (This value is calculated according to standard techniques, see, e.g. Brillouin [4].) We see that the average of the pictures produced by Algorithm 2 using these projections very nearly approaches this maximum value. The average of 20 reconstructions (Figure 5(a)) is hardly distinguishable from the picture with the maximum normalized entropy (Figure 5(b)).

Using only four projections, we get a reasonable reconstruction of Judy's picture (Figure 5(d)). In particular, the average of 20 reconstructions compares very favorably with the picture produced by Algorithm 3 using the same amount of computer time (Figure 5(c)). In spite of the fact that Algorithm 3 directly optimizes the function $S$, Algorithm 2 produced a picture with a higher value (Figure 7). Also, the picture of Judy produced by Algorithm 2 is nearer to the original (Figure 6). Possibly one could improve the speed of Algorithm 3 by using some more sophisticated direct search self-optimization technique (see, e.g. Schmitt [25]), but we have not yet succeeded in doing so.

All in all, our experience with Judy's picture fitted our initial intuition as to what our Algorithms 1 and 2 would do. The average picture we produced seemed to tend to the picture which maximizes the value of the $S$ function for those projections, and (with four or more projections) they were good reconstructions of the original.

A rather important property of the graphs in Figures 6 and 7 is that they are to a large extent independent of the random numbers generated during the execution of our algorithm. For example in Judy's picture, the normalized entropy of individual reconstructions had the following ranges.

| | *Minimum* | *Maximum* |
|---|---|---|
| 2 projections | 0.9857 | 0.9866 |
| 4 projections | 0.9714 | 0.9741 |
| 8 projections | 0.9687 | 0.9714 |

When we consider that any picture which satisfies eight projections also satisfies two, this may appear to be an odd result. The reasons for it are discussed in [14, Appen. I].

This stability of the graphs is a strong argument in favor of our algorithm, since it shows that even though we have used a Monte Carolo technique our final result does not depend much on the actual outcome of the random choices.

As far as the cost of applying our algorithms is concerned, to obtain the final reconstruction (average of 20) of Judy with eight projections took approximately 5 minutes of central processor time on the CDC 6400, which costs approximately $50. Our timing here did not include an overhead, which comes from sorting the array elements into rays of a projection for Algorithm 1. Since this can be time-consuming, it is important to do it once and for all, and permanently store the results for later runs.

Another property of our method is that it is capable of handling experimental errors in the measurement of the projections. First of all, it eliminates noise automatically (this is in the basic nature of our algorithm). Secondly, inconsistent measurements will result in some dirt in the corners only, since our technique satisfies all but the last few linearly dependent rays of the projections in Algorithm 2, and all of them (at different times) in Algorithm 1. Thus we shall get a picture which is just about as good as it is possible to get with the available information.

Based on the algorithms described in this paper, we have since found some considerably faster iterative techniques for reconstructing pictures from their projections (Gordon, Bender, and Herman [13]). However, those techniques have the disadvantage that the reconstructed picture will not satisfy the projections *exactly*, although they can be approximated arbitrarily closely by iterating long enough.

Our algorithms have the ability to reconstruct an object adequately from relatively few projections. This is important, because taking too many photographs of an object with an electron microscope can decompose it. Our algorithms can be used to ascertain the structure of completely asymmetric objects, such as ribosomes (Bender, et al. [3]). Many fewer photographs are needed for a reasonable reconstruction of ribosomes, than the 30 which DeRosier and Klug [9] claim is needed by the Fourier method. (See also Crowther, et al. [7, 8].)

Since our algorithms make no assumptions about

the object, other than that it is stable, they are less biased than the parametric approach of Hookes, Randall, and Hopkins [16], which presumes that the object belongs to a particular subset of all pictures.

Our other results (Gordon, Bender, and Herman [13]) suggest that a narrow range of angles (within $\pm 30°$) will also give an adequate reconstruction, so that the object need not be rotatable a full 360°. This is a practical advantage, since most electron microscopes allow limited tilting of the stage.

Some entirely new methods, using techniques of communication theory, are being developed at present. A preliminary report can be found in Gaarder and Herman [11].

## 6. Applications

Our algorithms are directly applicable to the problem of finding the three-dimensional density in space of an object being viewed with an electron microscope. Since the object can be rotated around a single axis, each plane through the object, perpendicular to the axis, becomes an unknown picture. Its projections are taken photographically, using electrons which pass through the object, perpendicular to the axis, and thus across the plane. (The number of electrons arriving at a point on the photographic plate, and therefore the darkness of the plate at the point, depends on the total density of matter within the object along the corresponding ray.) Each plane may be reconstructed in turn, and the results stacked to give the three-dimensional reconstruction. The separation of the planes should be no greater than the resolution of the microscope.

Each "plane" is thus a layer of small thickness. If the volume of space corresponding to picture element $(i,j)$ transmits a fraction $T_{i,j}$ of an impinging beam of electrons, then the fraction of the beam left after it has passed along a given ray through the whole specimen is

$$T = \prod_{(i,j) \in \text{ray}} T_{i,j} \tag{17}$$

(assuming Beer's law holds). We define the value of the ray, $R$, and the optical densities or gray levels $\rho_{i,j}$ by

$$-\eta R \equiv \ln T = \sum_{\text{ray}} \ln T_{i,j} \equiv -\eta \sum_{\text{ray}} \rho_{i,j} \tag{18}$$

or

$$T_{i,j} = e^{-\eta \rho_{i,j}} \tag{19}$$

where $\eta > 0$ is a scale factor. The electron density of the object at $(i,j)$ is then

$$E_{i,j} = \kappa(1 - T_{i,j}) = \kappa(1 - e^{-\eta \rho_{i,j}}) \tag{20}$$

where $\kappa$ is a constant. $R$ is, in general, a nonlinear, monotonic function of the optical density of the film at the appropriate point (Schroeder [26]).

Related uses of our algorithms might be in the search for hidden chambers in the pyramids using cosmic rays (Alvarez, et al. [1]), and the reconstruction of large, visually opaque three-dimensional objects from a few X-ray photographs. Because of the relatively few exposures necessary, our algorithms may be preferable to X-ray holography (Redman, et al. [18]) and tomography (Weinbren [27]) for human patients.

If the constraint of nonnegative gray levels is lifted, Algorithms 1 and 2 may also be used for objects containing embedded emittors. Such a case could arise in autoradiography using a collimating beam detector (Robinson and Jaffe [20]).

In situations where the transmitting of pictures is very expensive, for example, from Mars to Earth, our algorithms could be used for reducing this cost. This is because, if $m$ projections of $n$ rays each are adequate to describe the main features of an $n \times n$ picture (and as we have seen, usually $m \ll n$), we need to transmit only $mn$ numbers instead of $n^2$. If we assume that ray sums require the same average number of bits to transmit as the picture elements (this introduces a maximum of half-gray level average noise in the picture), then the transmission cost is reduced in the ratio of $n$ to $m$. This can easily be as high as 10, 20, or even more. This picture compression technique (Rosenfeld [21]) compares favorably with many others. (See, e.g. Roberts [19] for an interesting coding technique which reduces the number of bits to be transmitted by half.) It must be pointed out, however, that our decoding (reconstruction) is more expensive than in the other techniques. For this reason it would only be used when transmission costs are high enough to justify the computing cost in decoding.

Similarly, if storage costs are high, our algorithm could be applied to retrieve a picture compressed for storage purposes by taking its projections.

Chemical equilibrium problems for complex mixtures (White, Johnson, and Dantzig [28], Passy and Wilde [17]) involve maximizing a function very similar to our $S$, under linear constraints. Our algorithms may be generalizable to solve such problems. In any case, we have added to the repertoire of Monte Carlo methods for solving simultaneous linear equations (Hammersley and Handscomb [15], Gordon [12]).

Algorithm 3 maximizes eq. (11) in integers. This is apparently the first method for optimizing expressions of such form in integers over linear constraints (Balinski [2], cf. Saaty [23, 24]).

Chang [5] and Chang and Shelton [6] discuss the importance of reconstruction of binary patterns in the context of pattern recognition. If $l = 2$, our algorithms

can be considered to be alternative solutions to the problem of finding a binary pattern which satisfies certain projections. If we do not restrict $l$, our algorithms provide a solution to a generalized version of their problem.

To conclude on a less serious note, we wish to point out that the image reconstruction problem, as we have approached it, is a generalization of the magic square problem (see, e.g. Dudeney [10]). Algorithm 2 may be altered for solving magic squares by putting in the additional constraint that all gray levels must be distinct.

*Acknowledgments.* We thank Professor Cyrus Levinthal for suggesting the original problem, Karen Gordon for testing algorithms and helping digitize the picture of Judy by hand, Judith Carmichael for providing the photograph, and Dr. Alan K. Bruce for use of a densitometer. We have been helped by discussions with Dr. Patricia Eberlein, Dr. Peter W. Aitchison, Dr. Norman Abrahamson, Dr. Thomas Gaarder, and Dr. Dieter Girmes. Computing time was provided by the Columbia University Computer Center, the Woods Hole Oceanographic Institution Information Processing Center (National Science Foundation Grant GJ-133), and the SUNY/Buffalo Computing Center. We also thank the Xerox Computer Science Laboratory in Rochester, New York, for the use of the Xerox LDX output device and halftone routines. In particular we wish to thank Mr. Arthur R. Axelrod and Dr. Pierre A. Lavallee of Xerox Corporation for their help in producing the halftones.

Received September 1970; revised June 1971

**References**

1. Alvarez, L.W., Anderson, J.A., El Bedwei, F., Burkhard, J., Fakhry, A., Girgis, A., Goneid, A., Hassan, F., Iverson, D., Lynch, G., Miligy, Z., Moussa, A.H., Mohammed-Sharkawi, and Yazolino, L. Search for hidden chambers in the pyramids. *Science 167* (1970), 832–839.
2. Balinski, M.L. Personal communication, 1969.
3. Bender, R., Bellman, S.H., and Gordon, R. ART and the ribosome. A preliminary report on the three-dimensional structure of individual ribosomes determined by an algebraic reconstruction technique. *J. Theoret. Biol. 29* (1970), 483–487.
4. Brillouin, L. *Science and Information Theory* (2nd ed.), Academic Press, New York, 1962.
5. Chang, S.-K. The reconstruction of binary patterns from their projections. *Comm. ACM 14*, 1 (Jan. 1971), 21–25.
6. Chang, S.-K., and Shelton, G.L. Two algorithms for multiple view binary pattern reconstruction. *IEEE Trans. Syst. Man. Cyber., 1* (1971), 90–94.
7. Crowther, R.A., Amos, L.A., Finch, J.T., DeRosier, D.J., and Klug, A. Three dimensional reconstruction of spherical viruses by Fourier synthesis from electron micrographs. *Nature 226* (1970), 421–425.
8. Crowther, R.A., DeRosier, D.J., and Klug, A. The reconstruction of a three-dimensional structure from projections and its application to electron microscopy, *Proc. Roy. Soc. London Ser. A 317* (1970), 319–340.
9. DeRosier, D.J., and Klug, A. Reconstruction of three dimensional structures from electron micrographs. *Nature 217* (1968), 130–134.
10. Dudeney, H.E. *Amusements in Mathematics*. Dover, New York, 1958.
11. Gaarder, T., and Herman, G.T. Algorithms for reproducing objects from their X-rays. *Computer Graphics and Image Processing*, in press.
12. Gordon, R. On Monte Carlo algebra. *J. Appl. Prob. 7* (1970), 373–387.
13. Gordon, R., Bender, R., and Herman, G.T. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *J. Theoret. Biol. 29* (1970), 471–481.
14. Gordon, R., and Herman, G.T. Reconstruction of pictures from their projections. *Quarterly Bulletin of the Center for Theoretical Biology of the State University of New York at Buffalo 4*(1) (1971), 71–151.
15. Hammersley, J.M., and Handscomb, D.C. *Monte Carlo Methods*. Methuen, London, 1965.
16. Hookes, D.E., Randall, J., and Hopkins, J.M. Problems of morphopoiesis and macromolecular structure in cilia. *Formulation and Fate of Cell Organelles*, Ed. K.B. Warren. Symp. Int. Soc. Cell Bio. 6, Academic Press, New York, 1967, pp. 115–173.
17. Passy, U., and Wilde, D.J. A geometric programming algorithm for solving chemical equilibrium problems. *SIAM J. Appl. Math. 16* (1968), 363–373.
18. Redman, J.D., Wolton, W.P., and Shuttleworth, E. Use of holography to make truly three-dimensional X-ray images. *Nature 220* (1968), 58–60.
19. Roberts, L.G. Picture coding using pseudo-random noise. *IRE Trans. IT-8* (1962), 145–154.
20. Robinson, K.R., and Jaffe, L.F. A proposed new technique for autoradiography. *Third International Biophysics Congress*, 1969, p. 305.
21. Rosenfeld, A. Picture processing by computer. *Comput. Surv. 1* (1969), 146–176.
22. Rosenfeld, A. *Picture Processing by Computer*. Academic Press, New York, 1969.
23. Saaty, T.L. On nonlinear optimization in integers. *Naval Res. Logist. Quart. 15* (1968), 1–22.
24. Saaty, T.L. *Optimization in Integers and Related Extremal Problems*. McGraw-Hill, New York, 1970.
25. Schmitt, E. Adaptive computer algorithms for optimization and root-finding. NTZ-Rep. No. 6, VDE Verlag, Berlin, 1969.
26. Schroeder, M.R. Images from computers and microfilm plotters. *Comm. ACM 12*, 2 (Feb. 1969), 95–101.
27. Weinbren, M. *A Manual of Tomography*. H.K. Lewis & Co. Ltd., London, 1946.
28. White, W.B., Johnson, S.M., and Dantzig, G.B. Chemical equilibrium in complex mixtures. *J. Chem. Phys. 28* (1958), 751–755.