Abstract—The discrete tomography problem is to reconstruct a binary image from a limited set of its projections (sums of subsets of pixels). We formulate it as a one-bit phase retrieval problem and solve it using the (iterative) hybrid input-output algorithm of Fienup. Numerical examples illustrate the algorithm.

Keywords—Binary image, discrete tomography

I. INTRODUCTION

A. Problem Statement

The goal is to reconstruct a binary image \( x_{ij} \) from a limited set of projections (sums of subsets of \( x_{ij} \))
- \( y_n = \sum_{i,j} H_{n,i,j} x_{ij} \) is underdetermined;
- \( x_{ij} \) is zero or one everywhere (binary image);
- \( H_{n,i,j} \) is a sparse binary matrix (sums of \( x_{ij} \)).

We also generalize to arbitrary underdetermined linear combinations of \( x_{ij} \) (arbitrary matrices \( H_{n,i,j} \)).

B. Review of Discrete Tomography

The discrete tomography problem is to reconstruct a binary image from its row and column sums. This is equivalent to reconstructing the image from its discrete Fourier transform (DFT) values on the axes.

The solution to this problem is not unique. But all solutions are related to each other by “switches” of discrete Fourier transform (DFT) values on the axes.

C. Review of Hybrid Input-Output Algorithm

The hybrid input-output algorithm (HIO) was developed by Fienup to solve the phase retrieval problem of reconstructing an image from knowledge of the magnitude of its DFT and image information, e.g.,

- Support: \( x_{ij} = 0 \) outside a known region;
- Non-negativity: \( x_{ij} \geq 0 \) everywhere;
- Symmetry: \( x_{ij} \) has rotational symmetry.

II. REFORMULATION OF DISCRETE TOMOGRAPHY AS PHASE RETRIEVAL

A. Discrete Projection-Slice Theorem

The projection-slice theorem of tomography states that projections of an image are samples of its Fourier transform along radial lines. The discrete projection-slice theorem is derived from the 2-D DFT, which is:

\[
X_{k_1,k_2} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x_{i,j} e^{-j2\pi(i_1k_1+j_1k_2)}
\]

Setting \( k_2 = 0 \) gives the 1-D DFT of row sums:

\[
X_{k_1,0} = \sum_{i=0}^{N-1} e^{-j2\pi(i_1k_1)} \sum_{j=0}^{N-1} x_{i,j}
\]

Setting \( k_1 = 0 \) gives the 1-D DFT of column sums:

\[
X_{0,k_2} = \sum_{j=0}^{N-1} e^{-j2\pi(j_2k_2)} \sum_{i=0}^{N-1} x_{i,j}
\]

For more projections, setting \( k_1 = k_2 = k \) gives

\[
X_{k,k} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x_{i,j} e^{-j2\pi(i+j)}
\]
which is the 1-D DFT of the 135° projection
\[ p_n = \sum_{i_1=0}^{N-1} \sum_{i_2=0}^{N-1} x_{i_1,i_2} \delta[n - (n_1 + n_2) \mod(N)] \] (7)
Similarly, setting \( k_1 = -k_2 = k \) gives
\[ X_{k,-k} = \sum_{i_1=0}^{N-1} \sum_{i_2=0}^{N-1} x_{i_1,i_2} e^{-j \frac{2\pi k}{N}(i_1-i_2)} \] (8)
which is the 1-D DFT of the 45° projection
\[ p_n = \sum_{i_1=0}^{N-1} \sum_{i_2=0}^{N-1} x_{i_1,i_2} \delta[n - (n_1 - n_2) \mod(N)] \] (9)
Thus knowledge of various projections of \( x_{ij} \) is equivalent to knowledge of the 2-D DFT of \( x_{ij} \) along radial lines of \((k_1, k_2)\) at angles of 45°, 90°, 135° and 180°.

Note that the 45° and 135° projections are aliased; to avoid this we must zero-pad the 2-D DFT. This would make the iterative HIO algorithm much more difficult to implement, since the forward and inverse 2-D DFTs could no longer be used exclusively here.

B. Binary Image as One-Bit Phase

The binary image \( x_{ij} = \{0, 1\} \) can be mapped to a bilevel image \( x_{ij} = \pm 1 \) by simply converting the data:

- **General:** \( y_i \leftarrow 2y_i - \sum_{j=1}^{N} h_{ij} \)
- **Row sums:** \( y_i \leftarrow 2y_i - N \)
- **Col. sums:** \( y_i \leftarrow 2y_i - N \)
- **45° sums:** \( y_i \leftarrow 2y_i - (N - |N - i|) \)
- **135° sums:** \( y_i \leftarrow 2y_i - (N + |N - i|) \)

Double each projection and subtract off the number of elements being summed in that projection.

Then the projection of \( x_{ij} \) onto \pm 1 is performed by
\[ x_{ij} \leftarrow \text{sign}[x_{ij}] = 1e^{j \arg[x_{ij}]} \] (10)
which is the same as the projection \( Q \) in the HIO algorithm of setting the magnitude of \( x_{ij} \) to its known value (here, unity everywhere); \(|x_{ij}| = 1\) everywhere.

C. Application of HIO to Discrete Tomography

The HIO algorithm can thus be applied to discrete tomography by reformulation as phase retrieval:

1. Initialize with random \( x_{ij} \);
2. \( X_{k_1,k_2} = \text{DFT}[x_{ij}] \) (see (3));
3. Set \( X_{k,0}, X_{0,k}, X_{k,k}, X_{-k,-k} \) to known values;
4. \( \mathcal{P}(x_{ij}) = \text{DFT}^{-1}[X_{k_1,k_2}] \);
5. \( x \leftarrow x + \text{sign}[2\mathcal{P}(x_{ij}) - x] - \mathcal{P}(x_{ij}) \)
6. Continue until convergence (\( x_{ij} \) stops changing).

D. Application of HIO to General Matrix \( H \)

The HIO algorithm can also be applied to (1) with a more general matrix \( H \). Given data \( y_n \) as defined in (1), we now redefine the projection \( \mathcal{P}(x) \) to be
\[ \mathcal{P}(x) = x + H^T(HH^T)^{-1}(y - Hx) \] (11)
which is the projection of \( x \) onto the affine hyperplane \( y = Hx \). If the matrix \( H \) does not have full row rank, \( H^T(HH^T)^{-1} \) must be replaced by the appropriate pseudo-inverse, as in the second example below.

The problem with this is that computation of the 2-D DFT is relatively fast, requiring \( N^2 \log_2 N \) computations. However, the matrix-vector multiplications each require \( MN^2 \) computations, where \( M \) is the number of observations, summed over all of the projections. And of course the pseudo-inverse must be computed and stored, or a linear system of equations solved at each recursion. Hence the HIO algorithm seems much more suitable for discrete tomography.

III. NUMERICAL EXAMPLES

A. From Radial Fourier Samples

The 2-D DFT values \( \{X_{k,0}, X_{0,k}, X_{k,k}, X_{-k,-k}\} \) are equivalent to the aliased projections at 45°, 90°, 135° and 180° (see above). However, his is not enough data to ensure a unique solution; the two images

\[
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}
\]

have the same (unaliased) projections (note the four switches). Hence the problem is solved when a binary image whose projections agree with the data is found.

...
B. From Unaliased Projections

Here we formulate the problem as an underdetermined linear algebra problem \( y = Hx \), where

- \( H \) is a binary matrix that sums subsets of \( x_{ij} \);
- \( HH^T \) is diagonal, with some zeros on it, hence:
- \( H \) is always row-rank-deficient by seven, since:
- The projections all have \( X_{0,0} \) in common, and:
- \( X_{0,0} \) is real-valued, for a deficiency of \( 2(4) - 1 = 7 \);
- Hence the pseudo-inverse must be used in \( P \);
- \( H \) can be replaced with an arbitrary matrix below.

\[
\begin{align*}
clear;N=10;N1=2*N-1;N2=N*N; \\
X=round(rand(N,N));C=sum(X,1);R=sum(X,2); \\
for I=1:N-1;NE(N+I)=sum(diag(X,I));end \\
NW(N+I)=sum(diag(flipud(X),I));end \\
\%Convert from [0,1] to [-1,1]: \\
C1=2*C-N;NE1=2*NE-[1:N-1 N:-1:1]; \\
R1=2*R-N;NW1=2*NW-[1:N-1 N:-1:1]; \\
Y=[NW1 NE1 R1' C1]';Z=randn(N2,1); \\
\%Create projection matrix \( H \): \\
\%Note \( H \) is rank-deficient by 7. \\
for I=0:N-1;H(I+1:I+N,I*N+1:I*N+N)=eye(N); \\
H(N1+I+1:N1+1+I*N+1:N1+1+I*N+N)=flipud(eye(N));end \\
H=[H;kron(ones(1,N),eye(N));kron(eye(N),ones(1,N))]; \\
PH=pinv(H);Z0=PH*Y;A=eye(N2)-PH*H; \\
\%Hybrid Input-Output Recursion: \\
for J=1:100;PZ=A*Z+Z0; \\
W=sign(2*PZ-Z);Z=Z+W-PZ; \\
E(J)=sum(abs(Y-H*W)); \\
if(E(J)<0.001);break;end \\
\%Convert from [-1,1] to [0,1]: \\
W=(W+1)/2;W=reshape(W,N,N); \\
subplot(221),imagesc(X),axis off,colormap(gray) \\
subplot(222),imagesc(W),axis off,colormap(gray) \\
subplot(212),plot(E) \\
\end{align*}
\]

Typical runs of both programs are shown:

- The error tends to vary up and down for awhile, and then suddenly plunge down to zero (solution);
- The number of iterations required for convergence increases rapidly with problem size, but is finite;
- Original and reconstructed images are shown;
- The reconstructed image satisfies all projections;
- The reconstructed image is related to the original image by multiple switches of the kind shown above.