The Discrete Gould Transform And Its Applications

Hoang M. Le and Maurice Aburdene

Bucknell University

ABSTRACT

We present a new discrete transform, the Gould transform (DGT). The transform has many interesting mathematical properties. For example, the forward and inverse transform matrices are both lower triangular, with constant diagonals and sub-diagonals and both can be factored into the product of binary matrices. The forward transform can be used to detect edges in digital images. If G is the forward transform matrix and y is the image, then the two dimensional DGT, GyG^T can be used directly to detect edges. Ways to improve the edge detection technique is to use the "combination of forward and backward difference", $G^T(Gy)$ to better identify the edges. For images that tend to have vertical and horizontal edges, we can further improve the technique by shifting rows (or columns), and then use the technique to detect edges, essentially applying the transform in the diagonal directions.

1. INTRODUCTION TO SOME COMBINATORIAL NOTATIONS

We present a new transform for signal and image processing applications. The discrete Gould transform is based on combinatorial notation described in Riordan [1, p 49]. Therefore, before presenting the transform, we will introduce the mathematical convention to facilitate the presentation of the transform.

By convention, $\binom{n}{k} = C(n,k) = n!/k!(n-k)!$ is the usual notation for a binomial coefficient when n and k are positive integers. However, in the cases where either n or k is negative, we use the following:

$\binom{n}{0} = 1$ when $n = 0, \pm 1, \pm 2, ...$

$\binom{0}{k} = \delta_{0k}$, with δ_{nm} being the Kronecker delta $(\delta_{nn} = 1, \delta_{nm} = 0, n \neq m)$

$\binom{n}{-m} = 0, n = 0, \pm 1, \pm 2, ..., m = 1, 2, ...,$

and

$\binom{-n}{m} = (-1)^m \binom{n+m-1}{m}, n = 0,1,2,..., m = 0,1,2,...,$

2. GOULD'S CLASS OF INVERSE RELATIONS

A class of inverse relation was given by Gould (1961)[1] , in the form

$$F(n) = \sum_{k=0}^{n} (-1)^k \binom{n}{k} \binom{a+bk}{n} f(k),$$

$$\binom{a+bn}{n} f(n) = \sum_{k=0}^{n} (-1)^k \frac{a+bk-k}{a+bn-k} \binom{a+bn-k}{n-k} F(k)$$

By replacing a with p, b with q, $F(n)$ with a_n, $b_n = \binom{p+qn}{n} f(n)$, and using the binomial relation $\binom{n}{m}\binom{m}{p} = \binom{n}{p}\binom{n-p}{m-p}$, we have

$$a_n = \sum_{k=0}^{n} (-1)^k \binom{n}{k}\binom{p+qk}{n} f(k) = \sum_{k=0}^{n} (-1)^k \binom{p+qk-k}{n-k}\binom{p+qk}{k} f(k)$$

$$= \sum_{k=0}^{n} (-1)^k \binom{p+qk-k}{n-k} b_k$$

and

$$b_n = \sum_{k=0}^{n} (-1)^k \frac{p+qk-k}{p+qn-k}\binom{p+qn-k}{n-k} a_k$$

Let $q = 1$ then, we obtain

$$a_n = \sum_{k=0}^{n} (-1)^k \binom{p}{n-k} b_k, \quad b_n = \sum_{k=0}^{n} (-1)^k \frac{p}{p+n-k}\binom{p+n-k}{n-k} a_k = \sum_{k=0}^{n} (-1)^k \binom{p-1+n-k}{n-k} a_k = \sum_{k=0}^{m} (-1)^n \binom{-p}{n-k} a_k$$

Now if we replace b_k by $(-1)^n b_k$, we have our version of the discrete Gould transform

$$a_n = \sum_{k=0}^{n} (-1)^{k+n} \binom{p}{n-k} b_k, \quad b_n = \sum_{k=0}^{n} (-1)^{n-k} \binom{-p}{n-k} a_k \tag{1}$$

We note that p can be any positive integer**.**

3. THE DISCRETE GOULD TRANSFORM

3.1 Transform matrix

Based on the inverse relations in (1), the $N \times N$ transform matrix G is defined as $G_{jk} = (-1)^{j+k}\binom{p}{j-k}$ for $j,k = 0,1,...,N-1$, where p is our chosen positive integer.

Since $\binom{n}{-m} = 0, n = 0,\pm1,\pm2,..., m = 1,2,...,$ the transform matrix is lower triangular matrix.

G is defined by two parameters, the number of rows and columns N, and the parameter p that can be selected for a specific application.

For example, the case $p = 2, N = 4$ yields:

$$G(4,2) = \begin{bmatrix} 1 & & & \\ -2 & 1 & & \\ 1 & -2 & 1 & \\ & 1 & -2 & 1 \end{bmatrix}$$

3.2 One-dimensional Gould transform

Given a set of data $y[n]$ with $n = 0,1,...N-1$, and a positive integer p, the discrete transform is given by

$$Y[n] = \sum_{k=0}^{n} (-1)^{k+n} \binom{p}{n-k} y[k] \qquad (2)$$

where $G = G(N, p)$ is the $N \times N$ transform matrix, and y is the $N \times 1$ vector whose elements are $y[n]$. Table 1 shows the Gould transform of various signals.

3.3 The inverse Gould transform

Based on the inverse relations in (1), we can see that the $N \times N$ inverse transform matrix G^{-1} is defined as

$G_{jk}^{-1} = (-1)^{j-k} \binom{-p}{j-k}$ for $j,k = 0,1,..., N-1$, where p is our chosen positive integer.

Notice that according to our notation, $G_{jk}^{-1} = (-1)^{j-k} \binom{-p}{j-k} = \binom{j-k+p-1}{j-k} = \binom{j-k+p-1}{p-1}$

So if we have an $N \times 1$ vector whose elements are $Y[n]$, then the inverse Gould transform can be represented as:

$$y[n] = \sum_{k=0}^{n} (-1)^{n-k} \binom{-p}{n-k} Y[k] \qquad (3)$$

Some characteristics of the transform matrix and the inverse transform matrix are worth mentioning. For example, they are both lower triangular matrices. Both matrices are completely determined by the first column. Specifically, one column can be obtained from the preceding one by shifting all the entries down one row. Furthermore, both matrices have constant diagonals. This suggests that both forward and inverse transforms can be computed very fast in parallel.

3.4 Mathematical properties of the transform matrices

3.4.1 Determinants

For any n and p, det $G(n, p) = 1$ and det $G^{-1}(n, p) = 1$. Since both G and G^{-1} are triangular, then their determinants are equal to 1.

3.4.2 Eigenvalues

The eigenvalues of G(n,p) are all 1 for any n and p.

Proof

The eigenvalues λ satisfy $|G(n, p) - \lambda I| = 0$. The values from the first row of $G(n, p)$ consist of the value 1 at location (1,1) and 0 elsewhere. Notice that if we remove the first row and first column of the matrix $G(n, p)$, the remaining matrix is $G(n-1, p)$. Thus

$$|G(n, p) - \lambda I| = (1-\lambda)|G(n-1, p) - \lambda I| = (1-\lambda)^2 |G(n-2, p) - \lambda I| = ... = (1-\lambda)^{n-1}|G(1, p) - \lambda I|$$
$$= (1-\lambda)^n$$

From here we can see that all n roots of the polynomial $\det(G(n, p) - \lambda I)$ are 1.

3.4.3 Factoring the transform matrix

The forward transform matrix can be factored based on the transform matrix for $p = 1$.
For any n and any p, we have the relationship:

$$G(n, p) = G(n,1)^p$$

Since $G(n,1)$ has the form $\begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \end{bmatrix}$, $G(n, p)$ then can be factored into the matrices consisting of only 1 and -1.

As a consequence of the above relationship, for any n and any p, we also have

$$G(n, p)^{-1} = (G(n,1)^{-1})^p$$

Since $G(n,1)^{-1}$ has the form $\begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & 1 & 1 & \\ 1 & 1 & 1 & 1 \end{bmatrix}$, this also provides a way to factor the inverse transform matrix into binary matrices.

This property can be proved by mathematical induction.

Proof

For $p = 1,$ the statement becomes obvious.
Suppose that we have $G(n, k-1) = G(n,1)^{k-1}$, for some positive integer $k >= 2$.
We will show that $G(n, k) = G(n,1)^k$.
By definition, matrix $G = G(n, k)$ is defined by

$$G(i,\ j) = (-1)^{i+j}\binom{k}{i-j} = (-1)^{i+j}(\binom{k-1}{i-j} + \binom{k-1}{i-j-1})$$
$$= 1\times(-1)^{i+j}\binom{k-1}{i-j} + (-1)\times(-1)^{i+j-1}\binom{k-1}{i-j-1}$$

Therefore, $G(n, k) = G(n,1)\times G(n, k-1) = G(n,1)\times G(n,1)^{k-1} = G(n,1)^k$

By the principle of mathematical induction, the relationship is proved.

3.4.4 Basis functions

Our Gould transform matrix has a set of basis vectors $G_k = G(x, k, p) = (-1)^{(x+k)}\binom{p}{x-k}$ with p being the constant and x is the variable. Notice that x belongs to the set of integers. This set of basis vectors satisfies $X = \sum_{k=0}^{n} x_k G_k$. Varying p yields different basis functions for the different transform matrices $G(n, p)$. Here $G_k = G(x, k, p)$ with x varying from 0 to n-1 forms the k^{th} column of our transform matrix.

Notice that each basis vector from the transform matrix is the shifted version of the preceding basis vector.

For example, in the transform matrix G(8,3), the k^{th} column can be obtained by shifting the $(k-1)^{th}$ column.

$$\begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & -3 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -3 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 3 & -3 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 3 & -3 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 3 & -3 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 3 & -3 & 1 \end{matrix}$$

This shifting operation is equivalent to applying the *right shift matrix* to one of the basis vector.

$$\begin{bmatrix} 0 & & & & & & & \\ 1 & 0 & & & & & & \\ & 1 & 0 & & & & & \\ & & 1 & 0 & & & & \\ & & & 1 & 0 & & & \\ & & & & 1 & 0 & & \\ & & & & & 1 & 0 & \\ & & & & & & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \\ \\ \\ \\ \\ x_n \end{bmatrix} = \begin{bmatrix} 0 \\ x_1 \\ x_2 \\ \\ \\ \\ \\ x_{n-1} \end{bmatrix}$$

Some properties of basics vectors are as follows:

Property 1: $\sum_{k=0}^{\infty} G(x,k,p) = 0$ when $x >= p$

Corollary: When $i \geq p$, the sum of all the numbers in the i^{th} row is equal to 0.

Property 2: $\sum_{x=0}^{\infty} G(x,k,p) = 0$

Corollary: When $j \leq n - p$, the sum of all the numbers in the j^{th} column is equal to 0.

Property 3: $\sum_{x=0}^{\infty} (-1)^{x+k} G(x,k,p) = \sum_{k=0}^{\infty} (-1)^{x+k} G(x,k,p) = 2^p$

3.4.4 Relations to other discrete transforms

The basis function of our transform matrix can be related to some discrete polynomials, such as the binomial and Hermite polynomials.

The binomial polynomials can be expressed as:

$$p(x,i,N) = \binom{N}{x} \sum_{l=0}^{i} (-2)^l \binom{i}{l} \frac{x^{(l)}}{N^{(l)}}$$

We can rewrite the binomial polynomials as:

$$\begin{aligned} p(x,i,N) &= \sum_{l=0}^{i}(-2)^l\binom{i}{l}\frac{x^{(l)}}{N^{(l)}}\binom{N}{x} = \sum_{l=0}^{i}(-2)^l\binom{i}{l}\frac{x(x-1)...(x-l+1)}{N(N-1)...(N-l+1)}\frac{N!}{x!(N-x)!} \\ &= \sum_{l=0}^{i}(-2)^l\binom{i}{l}\frac{(N-l)!}{(x-l)!(N-x)!} \\ &= \sum_{l=0}^{i}2^l\binom{i}{l}(-1)^l\frac{(N-l)!}{(x-l)!((N-l)-(x-l))!} \\ &= \sum_{l=0}^{i}(-1)^x 2^l\binom{i}{l}G(x,l,N-l) \end{aligned}$$

The discrete Hermite polynomials differ from the discrete binomial polynomials only by the normalization factor $\binom{N}{x}$.

Thus we can derive a similar relationship between the Hermite polynomials and our basis functions.

3.4.5 Data storage

The data size (word length) needed to store the transformed image or data can be easily estimated. For example, if we know that an NxN block of input data has maximum and minimum values that differ my m (largest value of a forward difference) , then we provide a bound for the transformed data. Specifically, if we apply the transform matrix G(N, 1) to this NxN block, then we can assure that the output data will not exceed m regardless of the values of the input data. If we apply the transform matrix G(N, 2) to this block, we can assure that the output data will not exceed 2m.

We compared the mean and standard deviation of the transform coefficients of discrete cosine transform (DCT) and the DGT using random image data ranging from 0to 255. The DGT coefficients have a mean of about 1/3 of the mean of the DCT coefficients and a standard deviation of about 1/6 of the standard deviation of the DCT coefficients.

4. APPLICATIONS IN EDGE DETECTION

4.1 Using the two-dimensional Gould transform

The original idea was to apply the 2-D Gould transform for p =1 to an image to detect the edges. This is equivalent to GyG^T, in which G is the transform matrix, and y is the image. We will demonstrate the technique using the case $N = 3$.

Suppose that the image $y = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & k \end{bmatrix}$,

The transform matrix $G = \begin{bmatrix} 1 & & \\ -1 & 1 & \\ & -1 & 1 \end{bmatrix}$ and its transpose $G^T = \begin{bmatrix} 1 & -1 & \\ & 1 & -1 \\ & & 1 \end{bmatrix}$.

If we apply the 2-D transform to the image, intermediate result Gy is

$Gy = \begin{bmatrix} a & b & c \\ d-a & e-b & f-c \\ g-d & h-e & k-f \end{bmatrix}$, and the 2-D transform is $GyG^T = \begin{bmatrix} a & b-a & c-b \\ d-a & (e-b)-(d-a) & (f-c)-(e-b) \\ g-d & (h-e)-(g-d) & (k-f)-(h-e) \end{bmatrix}$.

The essence of the 2-D Gould transform is that we perform the 1-D transform for every column, and then applying the

1-D transform for every row of the resulting matrix. This technique can certainly be used for edge detection. It can detect the sharp changes in the input data. In the above demonstration, if e is large relative to b and d , then the transition from b to e in the vertical direction and the transition from d to e in the horizontal direction are the sharp changes in the image. This change will be detected as $(e-b)-(d-a)$ will have a large value, at the same time $b-a$ and $d-a$ have small values. Figure 1 shows example of an image and its DGT transform.

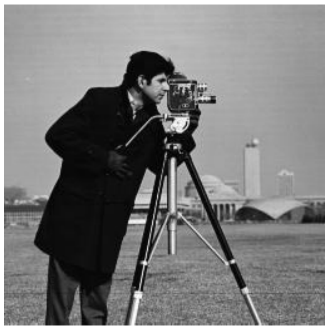
Figure 1(a). Cameraman

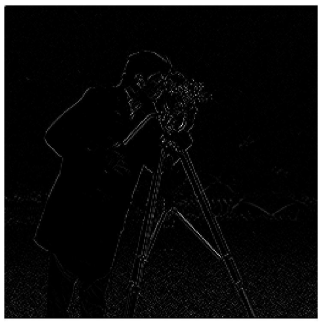
Figure 1(b). Edge Detection

In this case, typically four pixels will be involved in deciding whether a particular pixel is part of an edge. And this could be its weak spot. For example, consider the case where k and h are approximately equal, and both values are much greater than e and f . In the vertical direction, k should belong to an edge because there is a sharp difference between f and k . However, the value $(k-f)-(h-e)=(k-h)-(f-e)$ will approximately be 0.

One way to avoid this is to perform the 2-D Gould transform on the image y as describe above to get Y1, and then perform another transform on the transpose of image y to get Y2, and then add Y1 and Y2 to get image Y as shown in Figure 2.

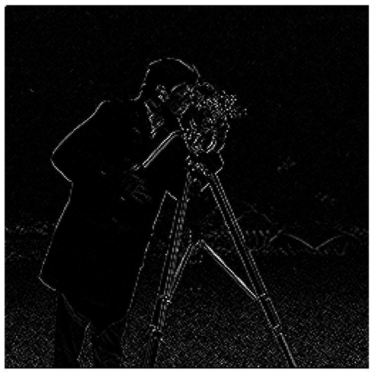
Figure 2. Enhanced edge detection method

4.2 Using the combination of forward and backward difference

Another technique results in much better edge detection than the 2-D transform technique. Using the notation from section 4.1, we know that performing the forward transform Gy is the same as applying the 1-D transform for every column. Now, instead of applying the 1-D transform for every row of the resulting matrix, we apply another column transform to it, but the transform matrix will be the transposed version of the original one. Then,
$Y=G^T(Gy)$.

This technique will detect the changes in the vertical direction . Let us take an example with N=3 to illustrate this idea.

Again, suppose that the image $y = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & k \end{bmatrix}$. With p =1, then $G = \begin{bmatrix} 1 & & \\ -1 & 1 & \\ & -1 & 1 \end{bmatrix}$, $G^T = \begin{bmatrix} 1 & -1 & \\ & 1 & -1 \\ & & 1 \end{bmatrix}$, and

$$Gy = \begin{bmatrix} a & b & c \\ d-a & e-b & f-c \\ g-d & h-e & k-f \end{bmatrix}$$

We note that this forward operation (Gy) in the horizontal direction can be an edge detector all by itself. The edges are show in Figure 3(a). However, it can be improved. Now we apply the transform

$$G^T(Gy) = \begin{bmatrix} a-(d-a) & b-(e-b) & c-(f-c) \\ (d-a)-(g-d) & (e-b)-(h-e) & (f-c)-(k-f) \\ g-d & h-e & k-f \end{bmatrix}.$$

We can see that if d is much greater than a and g then $2a-d, g-d$ will be discarded, $2d-a-g$ will be large and this technique can detect the edges in the horizontal directions easily. The result of applying this combination of forward and backward transform in the horizontal direction is shown in Figure 3(b). Notice that this result is much better than performing only the forward transform Gy.

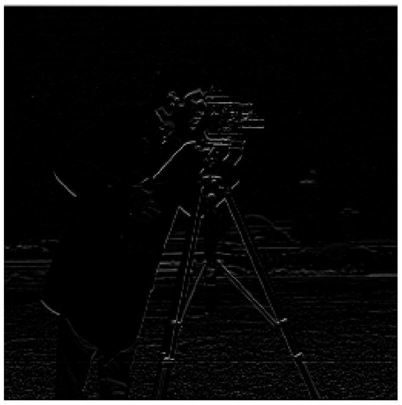
Figure 3(a). Edge detection using the forward difference only

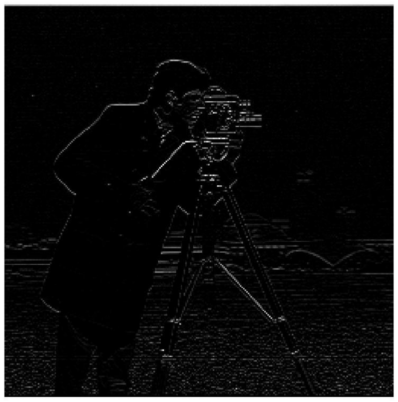
Figure 3(b). Edge detection using horizontal forward-backward difference combination

In addition, we can detect the edges in the vertical directions by repeating the same operations for the transpose of image y as shown in Figure 4(a). The addition of the two images 3(b) and 4(a) will give the edges of y in both directions. The result of this technique is shown in Figure 4(b)

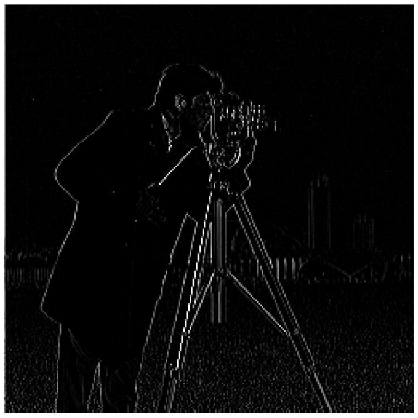

Figure 4(a). Edge detection using vertical Forward-backward difference combination

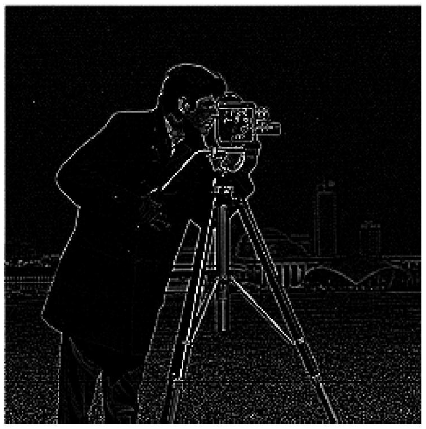

Figure 4(b). Edge detection using the combination of forward and backward differences in both directions

4.3 The improved forward-backward operation edge detection method

Earlier we described a method to detect edges using the combination of the forward and backward operations. We used the forward and backward operation to detect the vertical edges, and then used the same technique to detect the horizontal edges. However, it is important to note that many images are “upright” in nature, which means the images contain objects that have horizontal and vertical edges only. This could be a problem for our edge detector, because the above column edge detector will tend to cancel out the vertical edges (although the row edge detector will make up for it), and the row edge detector tends to cancel out the horizontal edges. Although the column edge detector and the row edge detector will complement each other, a diagonal edge detector method might be a better choice for “upright” images.

In this technique, we also detect the edges in two directions, which are the main diagonal and the subdiagonal direction. We demonstrate this technique for the case N = 3. Let the image be

$y = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & k \end{bmatrix}$. To perform the transform in the subdiagonal direction, we move the i^{th} row to the right $(i-1)^{th}$ step.

The first number of the i^{th} row will align with the i^{th} number of the first row.

After this step, we have the image $y = \begin{bmatrix} a & b & c & & \\ & d & e & f & \\ & & g & h & k \end{bmatrix}$. To obtain an $(2n-1)\times(2n-1)$ image, we pad our image with the same numbers as the top and the bottom number in each column. Doing this step to y yields:

$y = \begin{bmatrix} a & b & c & f & k \\ a & b & c & f & k \\ a & d & e & f & k \\ a & d & g & h & k \\ a & d & g & h & k \end{bmatrix}$. Now, applying the column detection technique for this image, using the Gould matrix for $N = 2n-1, p = 1$, and then extract the $N \times N$ matrix from the result, we will have the subdiagonal edge image. This is the subdiagonal result for the camera man image as shown in Figure 5 (a).

To perform the transform in the diagonal direction, we also move the rows, but to the left. Re-align the rows so that the i^{th} number of the i^{th} row aligns with the first number of the first row. The image y will have the form

$y = \begin{bmatrix} & & a & b & c \\ & d & e & f & \\ g & h & k & & \end{bmatrix}$. Now if we pad this image and apply the transform for the $(2n-1)\times(2n-1)$ image, we will have an edge image in the diagonal direction. This diagonal result for the camera man image as shown in Figure 5(b).

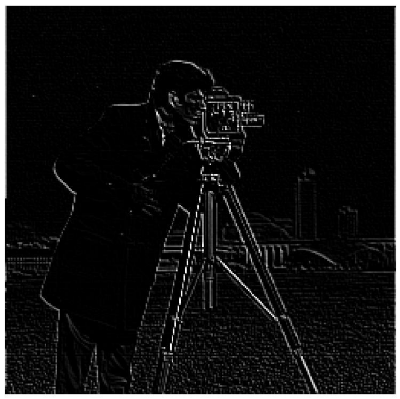
Figure 5(a). Subdiagonal edge detection

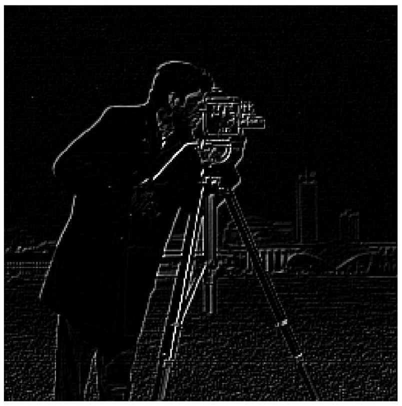
Figure 5(b). Main diagonal edge detection

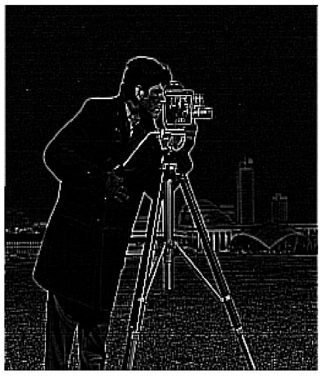
Figure 5(c). Result of improved method

When we add the two images (subdiagonal and diagonal technique), we obtain the result (better than the forward-backward for rows and columns technique) as shown in Figure 5 (c).

4.4 Thresholding the image

We can see that although the above method can detect the edges, it also includes noise in the resulting image. In fact, the noise is not "real noise", because it comes from actual edges (though tiny) of the original image. We are interested in a way to exclude the noise out of the transformed image. To do this, we need to find a threshold T so that every pixel that has the value greater than T belongs to an edge, and any other pixel that is less than T will be ignored, or considered to be the background.

This in essence will give a binary image. The value of T will be image dependent. The goal of this section is to find ways to estimate the reasonable values for T. We used the basic global thresholding method presented in [2, pg. 599].

The initial value for T is the average gray level of the original image. For the cameraman image, Figure 6 shows the edges using this algorithm after performing the combined forward-backward difference operations.

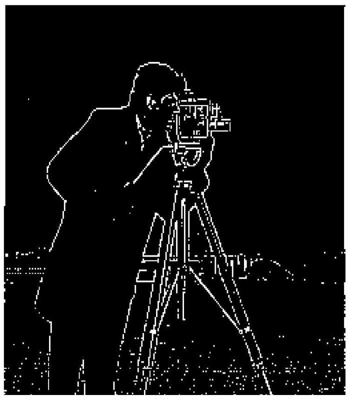

Figure 6: Edge detection using the transform and global thresholding method

As we can see, the presence of white dots in the lower half of the picture suggests that the noise-like edges are stronger than the building's edges in the original picture.

5. SUMMARY

The discrete Gould transform has applications in digital image processing such as edge detection since it allows a very quick way to detect edges in an image. The coefficients of the transform matrix perform a derivative operation of order p and so the inverse transform is "repeated integration" of order p. For example, p = 1 yields the coefficients 1 and -1 in the transform matrix, which can be used to approximate the first derivative of the input data. Case p = 3 yields the coefficients 1, -3, 3, -1, which can be used to approximate the third derivative. These derivative patterns suggest that the discrete Gould transform will detect the sharp changes in the input data, and will transform uniform input data to 0.

6. REFERENCES

[1] John Riordan, *Combinatorial Identities*, John Wiley and Sons, Inc., New York, 1979.
[2] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Addison- Wesley, 1992.

Table 1: Gould transforms

$x(n), n \ge 0$	$X(k), k \ge 0$	
$x(n)$	$\sum_{i=0}^{k} (-1)^{i+k} \binom{p}{k-i} x(i)$	(1)
$G(x(n))$	$\sum_{i=0}^{k} (-1)^{i+k} \binom{2p}{k-i} x(i)$	(2)
$ax(k) + by(k)$	$aX(k) + bY(k)$	(3)
$x(n-k_0), k_0 \ge 0$	$X(k-k_0)$	(4)
$x(n) * y(n)$	$G^{-1}(n,p)\{X(k) * Y(k)\}$	(5)
$\sum_{i=0}^{n} x(i)$	$\sum_{i=0}^{k} X(i)$	(6)
$\delta(k)$	$(-1)^k \binom{p}{k}$	(7)
$u(n)$	$(-1)^k \binom{p-1}{k}$	(8)
$(-1)^n \binom{a}{n}, a \in Z^+$	$(-1)^k \binom{a+p}{k}$	(9)
$e^{-\alpha n}$	$e^{-\alpha k}(1+e^{\alpha})^p$ (except the first p terms)	(10)
		(11)
$\sin(\omega n)$, p is even	$(-1)^{p/2} (2\sin(\frac{\omega}{2}))^p \sin(\omega(n-\frac{p}{2}))$ (except the first p terms)	(12)
$\sin(\omega n)$, p is odd	$(-1)^{p-1/2} (2\sin(\frac{\omega}{2}))^p \cos(\omega(n-\frac{p}{2}))$ (except the first p terms)	(13)
α^n	$\alpha^n (\frac{\alpha-1}{\alpha})^p$ (except for the first p terms)	(14)
$\alpha(1-\alpha)^n$	$\alpha(1-\alpha)^n (\frac{-\alpha}{1-\alpha})^p$ (except for the first p terms)	(15)