

ORIGINAL ARTICLE

ROTHSTEIN CODE TRANSFORMATIONS AS A DIGITAL OPTICAL SCALING METHOD

A.P. Dagher

Department of Diagnostic Radiology and Nuclear Medicine, University of Maryland Medical System, United States

ABSTRACT – The Rothstein code data transformation method is a largely forgotten and underutilized method originally theorized from optics and retinal recruitment. The code is a binary representation of the slope of a line or scaling factor and can transform data and images similar to matrix multiplications except with built-in interpolations. The theoretical validation of the method presented in the original publications is complex. The purpose of this paper is to borrow from quantum theory and dual nature of light, particle and wave, to simplify the theoretical understanding. With this understanding, the method becomes more intuitive in the example applications that follow. These application examples include digital resampling of data for signal processing applications, examples for matrix scaling such as for the management of discordant data sets, and for image processing.

ARTICLE HISTORY

Revised: 11 May 2020

Accepted: 13 July 2020

KEYWORDS

Rothstein code, interpolation, data transformations, optics, photons.

INTRODUCTION

Affine transformations of data (Schneider, and Eberly, 2003), such as images, have been the mainstay of data scaling, an operation that requires the two steps of matrix multiplication followed by the interpolation of data, to smooth out distortions or fill in the gaps from the matrix operation. The Rothstein code is a scalar data transformation method with built-in interpolation. References to the method are limited in the literature with a large time gap since its introduction. The code was originally introduced as a paradigm for computer vision and implemented as a method for geometric image transformation without matrix multiplication (Weiman and Rothstein, 1972, 1976; Weiman 1976; Dagher 1985; Snyder and Mackie, 1990). Rothstein's code is the binary representation of the slope of a line within a grid, represented by a one-dimensional Boolean array or vector of 1's and 0's. This is shown in Figure 1 for a factor of $3/2$.

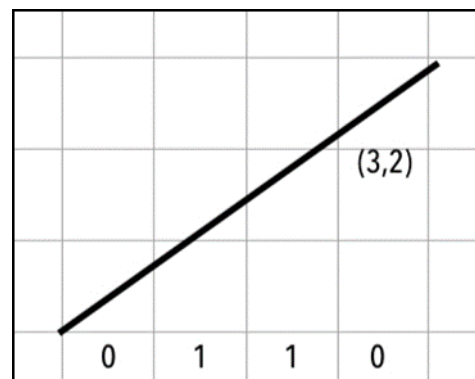


Figure 1. Rothstein Code of the Slope $3/2$ in a Grid.

As the line crosses a horizontal line in the coordinate system, a 1 is coded; otherwise a 0 is coded. The code repeats once the factor is represented, thereby becoming a digital representation of the slope of the line where the ratio is the number of 1's, represented by p , over the number of pixels needed to quantify the slope, or q . For this slope, there are three possible ways the code can be represented in a 3 point vector: **011**, **101**, and **110**. The binary code for slope or factor p/q is generated by a modulo operation $\%$ in the following loop of operations

```
for (i = 1; i <= q; i++) {
    if (i * p % q < p and (i * p) >= q)
```

```

    code[i - 1] = 1;
else code[i - 1] = 0;
}

```

Any decimal factor can be represented by a Rothstein code using its integer fraction. For example, a factor transformation of 0.125 is 125/1000 which can be reduced to the lowest term 1/8 ($p = 1, q = 8$). The factor 133/1000 would require the integer fraction 133/1000, already in the lowest term ($p = 133, q = 1000$).

The binary code can direct a scalar transformation with the permutations of the 1's and 0's implementing a matrix shrink or stretch by guiding a resampling of data. The 0's guide a 0-padding of data for a stretch with permutations of the code effecting an averaged interpolation of the data. In the case of a shrink, the 0's guide data eliminations where again the permutations of the code effect the interpolation. This will be demonstrated more clearly in a following sections. First however, we present the theoretical validation of the method using the particle theory of light passing through a lens.

OPTICAL THEORETICAL BASIS

The geometric proof of concept for the Rothstein code is detailed in the original dissertation (Weiman and Rothstein, 1972) and summarized by Weiman (1976). This proof is based on an optical geometric projection of an image in an analog sense that is computationally interpreted as a time-shared projection of a digital image onto a larger (or smaller) digital image. Since the transformed image has elements composed from several positions in the original image, the transformed image elements have values averaged from several original elements and these are directly related to the transformation factor.

A simpler proof however could be based on the dual nature of light that come from classical and quantum physics and is presented here. To quote Albert Einstein, "We have two contradictory pictures of reality; separately neither of them fully explains the phenomena of light, but together they do." (Einstein and Infeld, 1938).

The laws of refraction through a thin lens are based on the wave theory of light and are well established (Bueche, 1975). A ray diagram for a converging lens summarizes the minimization of an object to an inverted real image based on the focal length of the lens (f). The ratio of the distance of the inverted real image to the lens (d') over the distance of the object to the lens (d) determines the minimization factor $M = I'/I = d'/d$ as shown in Figure 2.

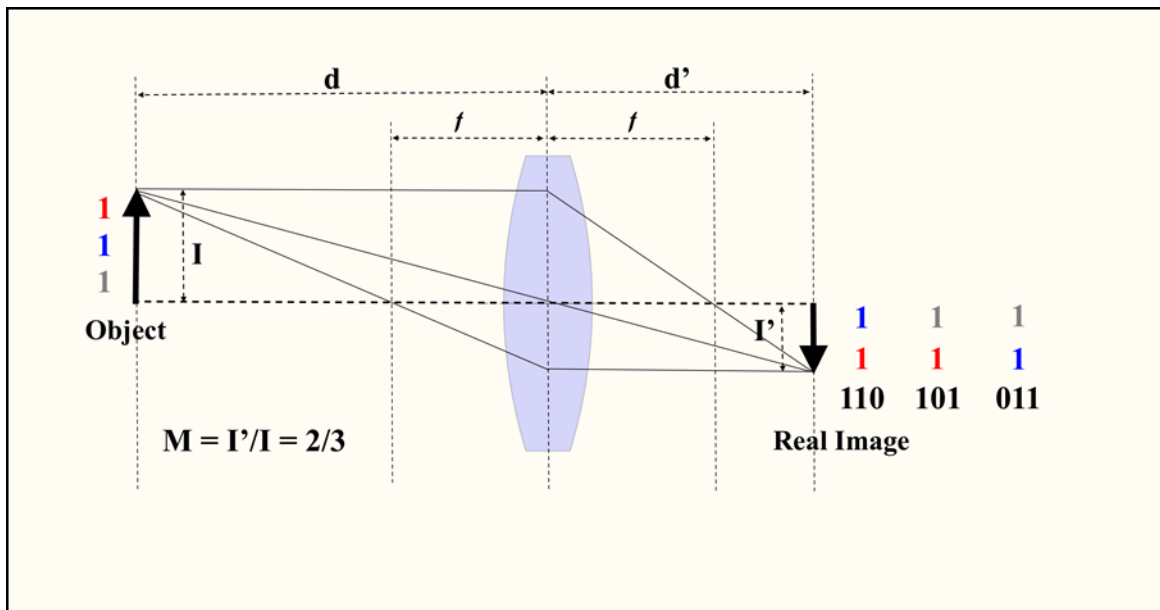


Figure 2. Ray diagram for the 2/3 shrink to a real image from a converging lens with Rothstein mapping of all potential photon positions

Using the particle nature of light, consider the hypothetical and simple one-dimensional case where only three photons are needed to represent the object being scaled by the factor 2/3. The smaller real image has two possible positions for the three photons from the object, represented by the vertical 1's in Figure 2. We color code the 1's representing the photon positions for clarity and to preserve sequence, noting that the real image is inverted. The process is of selecting two of the three photons onto the smaller real image with three possible results. This process represents the Rothstein code for 2/3 with its three possible permutations: **110**, **101** and **011**. Averaging the data values from these three possible digital guideposts would be equivalent to an analog shrink by 2/3. This is further demonstrated in the examples of the following section.

The ray diagram for creation of a magnified virtual image from a converging lens would have the same parameters: focal length of the lens (f) and magnification factor $M = I'/I = d'/d$. This is shown in Figure 3.

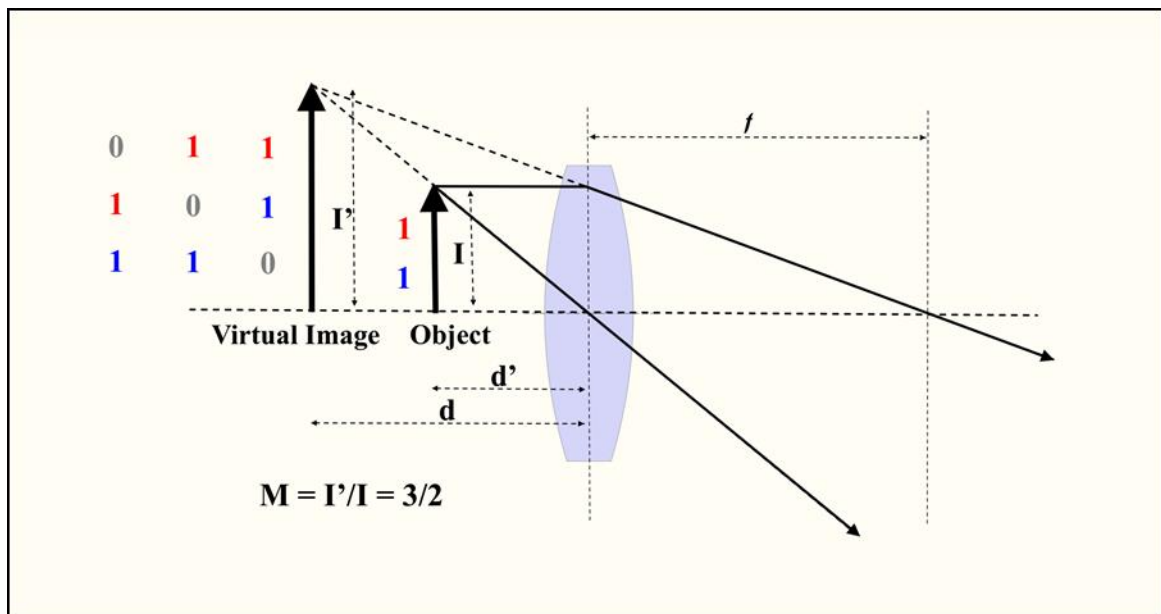


Figure 3. Ray diagram for the 3/2 magnification to a virtual image from a converging lens with Rothstein mapping of all potential photon positions.

Continuing within the particle nature of light paradigm, consider again a hypothetical and simple one-dimensional case where only two photons are needed to represent the object being magnified by the factor 3/2. The larger virtual image has three possible positions for the two photons from the object, represented by the vertical 1's in Figure 3. Again we color code the 1's representing the photons for clarity and to preserve order. This time the process is of 0-padding photon positions onto the virtual image with three possible results for the two photons to contribute to the larger virtual image. This process again parallels the Rothstein code for 3/2 with its three possible permutations: 110, 101 and 011. Averaging the data values from these three possible digital guideposts would be equivalent to an analog stretch by 3/2. This is also further demonstrated in the examples of the following section.

EXAMPLE APPLICATIONS

Resampling data

Rothstein code transformations are an effective tool to digitally resample data to arbitrary size with potential wide applicability in data processing (Dagher and Ptak, 2019). We can demonstrate using a simple example of a small integer vector of size $N=12$ to stretch with built-in interpolation. (Integer rounding is used although the method is equally valid with decimal arithmetic.)

1 3 7 3 1 3 7 3 1 3 7 3

First determine a stretch factor for a desired N ; for example 16 or a 1.33% stretch. This factor is converted to a fraction; in this case 16/12 reduced to the lowest term 4/3. Determine the four point Rothstein code using the modulus operation discussed above:

0 1 1 1

Repeat the code for the desired size of the new array (16).

0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1

The 0's will introduce the padded points by spacing the original series.

0 1 3 7 0 3 1 3 0 7 3 1 0 3 7 3

Now shift the binary code array by one location.

1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0

Repeat the step above with the new shifted code:

1 3 7 0 3 1 3 0 7 3 1 0 3 7 3 0

This step is repeated until the fraction $4/3$ is fully represented, in this case 4x total or 2 more shifts.

1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1
1 3 0 7 3 1 0 3 7 3 0 1 3 7 0 3

and

1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1
1 0 3 7 3 0 1 3 7 0 3 1 3 0 7 3

The vectors from all possible shifts are summed

3 7 13 21 9 5 5 9 21 13 7 3 9 17 17 9

Now divide the final array by the number of 1's in the code or p (3 in this case). The result is a new resampled and interpolated data series of length 16.

1 2 4 7 3 2 2 3 7 4 2 1 3 6 6 3

For completeness consider a shrink operation on the same integer array size $N=12$:

1 3 7 3 1 3 7 3 1 3 7 3

First determine a shrink factor for a desired N; for example , 8 or a 0.67% shrink. This factor is converted to a fraction; in this case $8/12$ or $2/3$ for a Rothstein code sequence of **110**.

This time repeat the code to the size of the original array (12).

1 1 0 1 1 0 1 1 0 1 1 0

The 1's will this time sample the original array to produce the new array of size 8.

1 3 3 1 7 3 3 7

Now shift the binary code array by one location and resample.

0 1 1 0 1 1 0 1 1 0 1 1
3 7 1 3 3 1 7 3

This step is repeated until the fraction $2/3$ is fully represented, in this case 3x or one more shift.

1 0 1 1 0 1 1 0 1 1 0 1
1 7 3 3 7 1 3 3

The summed vector is then

5 17 7 7 17 5 13 13

Now divide the final summed array by the number of shifts performed or q (3). The result is a new resampled and interpolated data series of length 8.

2 6 2 2 6 2 4 4

An example for a longer simulated data series of three harmonics is shown in Figure 4 (Dagher and Ptak, 2019).

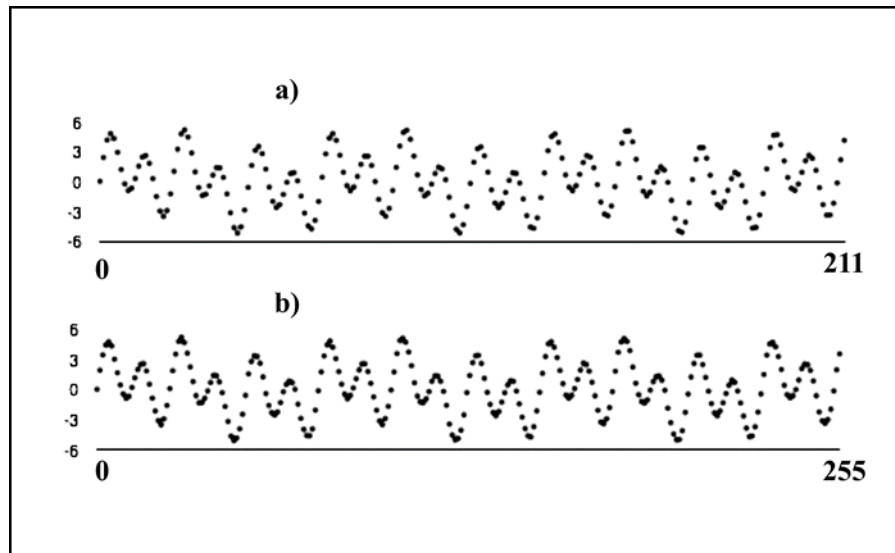


Figure 4. Simulated data of three sine waves $N = 211$ (a) resampled to $N = 255$ (b) using Rothstein code 256/212 reduced to lowest terms 64/53.

Resizing Discordant Data for Matrix Operations

Dagher (2020) shows utility of Rothstein code transformations for matrix operations where data sizes are discordant. For example, matrix subtraction requires matrices of equal size. Suppose a company A has a product line of 2 items priced differently across 2 regions. The product with regional price values can be represented by a 2×2 matrix

$$\mathbf{A} = \begin{vmatrix} 1 & 3 \\ 3 & 5 \end{vmatrix} \quad (\text{Eq. 1})$$

Suppose also a competitor company B with similar 2-product line has 3 regional distribution centers across the same area with the 2×3 matrix

$$\mathbf{B} = \begin{vmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \end{vmatrix} \quad (\text{Eq. 2})$$

In comparing the regional valuations of the two companies, such as for a merger, one may be interested in a map of the differences. Subtraction of these two matrices is untenable however we can scale one to the other's dimension using a Rothstein transformation. Matrix $\mathbf{B}$ requires a scaling in the horizontal axis to reduce it to 2 columns. This is a shrink by a factor of $2/3$ in the x direction. The Rothstein binary code representation of this fraction is again **011**, where 1's determine the columns to extract from $\mathbf{B}$.

$$\begin{vmatrix} 1 & 3 \\ 3 & 5 \end{vmatrix}$$

The code is then shifted to **101**, which determines new columns extracted

$$\begin{vmatrix} 1 & 3 \\ 3 & 5 \end{vmatrix}$$

The last shift of the code, **110**, results in a third matrix

$$\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}$$

The sum of these matrices

$$\begin{vmatrix} 4 & 8 \\ 10 & 14 \end{vmatrix}$$

is divided by the number of shifts used ($q = 3$). Rounding to the nearest integer we get the scaled matrix with built-in interpolation

$$\begin{vmatrix} 1 & 3 \\ 3 & 5 \end{vmatrix}$$

This is the same as **A** for a difference matrix of 0, indicating the companies are equivalent.

Alternatively a stretched **B** (Eq. 2) in the y direction into a 3 x 3 matrix subtracted from a stretched **A** (Eq. 1) in both directions into a 3 x 3 matrix could allow an alternative means of valuation.

Image Processing

Image transformations were the primary emphasis in the original works (Weiman and Rothstein, 1972, 1976: Weiman 1976) and examples of these can be visually dramatic particularly since the method does not require the added interpolation steps needed in the equivalent transformations using matrix multiplication. The method has excellent preservation of detail. Figure 5 shows a scalar stretch and shrink using the Rothstein code.



Figure 5. Rothstein code stretch of 3/2 in the x-direction and 2/3 shrink in the y-direction of the original image on the left.

Weiman (1976) also demonstrates the code for image shears in addition to scaling. Shears in proper sequence with scaling operations can produce geometric rotations of arbitrary angle, also without matrix multiplication and with built-in interpolation. See Figure 6.

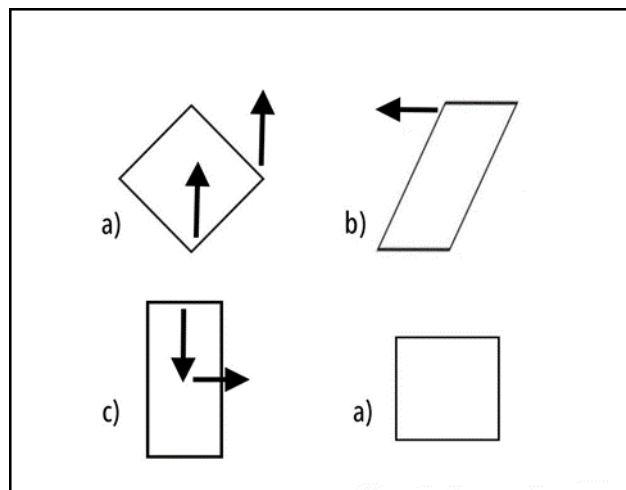


Figure 6. Rotation scalar operation sequence. (a) Shear +y to result in (b). (b) Shear -x to produce (c). (c) Shrink y, then stretch x. (d) rotation result.

CONCLUSION

The Rothstein code transformation is a valid method for data and matrix resampling and scalar operations. Why the method is under-utilized based on the literature is unclear. There is the possibility that the method is used under a different name than that used by the original publications. Even in this remote possibility, the original publications deserve recognition. The code can transform data and images similar to matrix multiplications and with built-in interpolations. The optical theoretical validation of the method presented in the original publications is supplemented in this work,

borrowing from quantum theory of the dual nature of light, particle and wave, to simplify the understanding. This theoretical basis in optics may signify a superior faithfulness to data detail for transformations.

REFERENCES

- [1] Bueche, F.J. (1975). Reflection and Refraction. Introduction to physics for scientists and engineers. New York: McGraw-Hill, 584-607.
- [2] Einstein, A., Infeld, L. (1938). The evolution of physics: the growth of ideas from early concepts to relativity and quanta. Cambridge University Press.
- [3] Dagher, A.P. (1985). Applying a Computationally Simple 2-D image transformation algorithm for digitized pictures. COMPINT Computer Aided Technologies, Montreal Canada.
- [4] Dagher, A.P. (2020). Rothstein's code transformations for discordant datasets: matrix operations. International Conference in Mathematics, Computer Engineering & Computer Science (ICMCECS2020).
- [5] Dagher, A.P., Ptak T. (2019). Resampling of data with Rothstein's code for the slope of a line. Fourth International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), Beirut, Lebanon, pp. 1-4.
- [6] Dagher, A.P., Ptak T. (2019). Resampling of data with Rothstein's code for the slope of a line. Fourth International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), Beirut, Lebanon, pp. 1-4.
- [7] Greiner, W. (2001). Wave aspects of matter. Quantum mechanics: an introduction. Springer. 32-35.
- [8] Schneider, P. K., Eberly, D. H. (2003). Geometric tools for computer graphics. Morgan Kaufmann. 98-104.
Snyder, A., Mackie, T.R. (1990). Continuous antialiased 2-D and 3-D transformations in discrete space, Computational Physics, 4,78-82.
- [9] Weiman C.F.R. Rothstein, J. (1972) Pattern recognition of retina-like devices. Computer and Information Science Dept., Ohio State University, OSU-CISRC-TR-72-8, 154.
- [10] University, OSU-CISRC-TR-72-8, 154. Weiman, C.F.R. (1976) Highly parallel digitized geometric transformations without matrix multiplication. Proceedings of the 1976 International Conference on Parallel Processing IEEE, 1-10.
- [11] Weiman, C.F.R., Rothstein J. (1976). Parallel and sequential specification of a context sensitive language for straight lines on grids. Computational Graphics and Image Processing, 5,106-124.