If we now combine like terms, taking note of how each line is formed from the previous, we have a pattern whose coefficients are those of the Pascal triangle:

$$
\begin{array}{lccccccccc}
 & & & & V_n & & & & \\
= & & & V_{n-1} & + & A_{n-1} & & & \\
= & & V_{n-2} & + & 2A_{n-2} & + & L_{n-2} & & \\
= & V_{n-3} & + & 3A_{n-3} & + & 3L_{n-3} & + & 1 \\
= & V_{n-4} & + & 4A_{n-4} & + & 6L_{n-4} & + & 4(1) \\
= & V_{n-5} & + & 5A_{n-5} & + & 10L_{n-5} & + & 10(1) \\
= \cdots = & V_0 & + & \binom{n}{1} A_0 & + & \binom{n}{2} L_0 & + & \binom{n}{3} (1).
\end{array}
$$

That is, we again have (2).

To generalize to (3), we note that just as a plane divides every volume it slices into two smaller volumes, a hyperplane divides every m-dimensional region it partitions into two smaller m-dimensional regions. If x_1 through x_m are the coordinates of our m-dimensional space, then a hyperplane would have equation $a_1x_1 + a_2x_2 + \cdots + a_mx_m = k_m$. Without loss of generality, we assume that the nth hyperplane has equation $x_m = 0$ and that all points of intersection of any m of the first $n - 1$ hyperplanes have $x_m > 0$. Then following the reasoning behind (4), each of the $S_{m-1,n-1}$ m-dimensional subspaces that intersect $x_m = 0$ are cut by it into two parts. Thus,

$$S_{m,n} = S_{m,n-1} + S_{m-1,n-1} \tag{6}$$

and corresponding Pascal triangles will again produce (3). (For the earliest reference, see [**6**]; for other presentations, see [**3, 4**]).

References

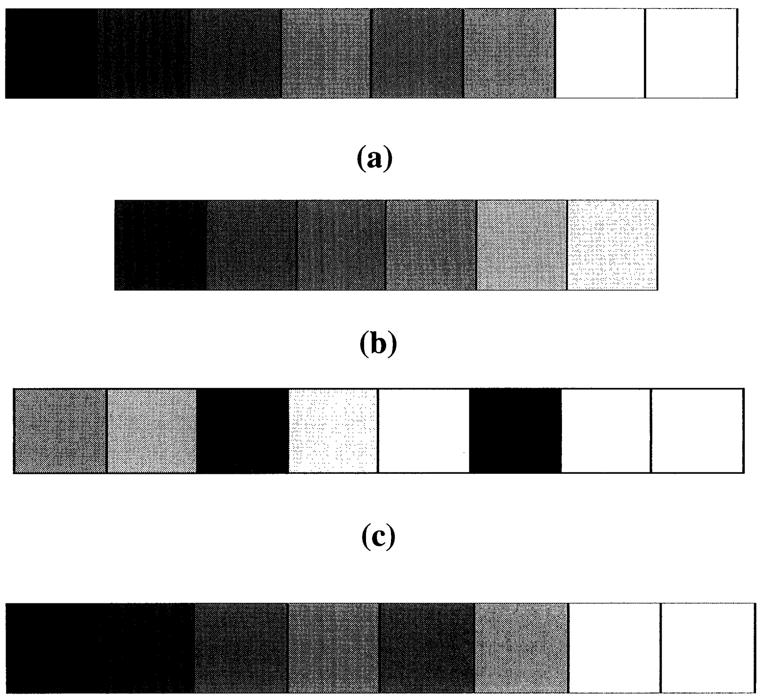
1. Gerald L. Alexanderson and John E. Wetzel, Simple divisions of space, *Mathematics Magazine*, 51:4 (1978) 220–225.
2. R. C. Buck, Partitioning of space, *American Mathematical Monthly* 50 (1943) 541–544.
3. Jerome W. Kerr and John E. Wetzel, Platonic divisions of space, *Mathematics Magazine*, 51:4 (1978) 229–234.
4. Derek J. Price, Some unusual series occurring in n-dimensional geometry, *Mathematical Gazette*, 30 (1946) 149–150.
5. G. Pólya, *Induction and Analogy in Mathematics*, Princeton University Press, 1954.
6. J. Steiner, Einige Gesetze über die Theilung der Ebene und des Raumes, *J. Reine Angew. Math* 1 (1826) 349–364.

Image Reconstruction in Linear Algebra

Andrzej Kędzierawski (kedziera@geneseo.edu) and Olympia Nicodemi (nicodemi@geneseo.edu), SUNY Geneseo, Geneseo, NY 14454

Recently, inspired by [**1**], we have been using one and two dimensional image reconstruction problems in our introductory course in linear algebra to motivate and illustrate various topics. We suppose that real world scenes are in "black and white" and that they come to us pre-digitalized as an array of pixels. Our hypothetical camera photographs a scene (an array) by matrix multiplication using a known matrix C. It transforms the original scene somewhat, perhaps blurring some of the details. Our main task is an inverse problem: to try to reconstruct the original scene from the photograph. The key to the classroom success of this application is a set of Maple programs we developed to produce visual images from digital data stored in arrays. These programs are available from the authors by e-mail.

We begin with a one-dimensional scene that is a row of pixels represented by a vector. Entries less than or equal to -1 appear black, entries greater than or equal to $+1$ appear white, with shades of gray in between. For instance, the vector $x = [-1, -.7, -.5, -.3, -.1, 0, 1, 1]$ has eight pixels that change gradually from black to gray and abruptly to white. (See Figure 1a.) A simple scanning camera moving across the row might receive light from three adjacent pixels and average those values, thus blurring and smoothing the image. Under this assumption, the resulting camera image y is easy to calculate by $y_i = \frac{1}{3}(x_{i-1} + x_i + x_{i+1})$. Note that y has two pixels fewer than the original scene.



(a)

(b)

(c)

(d)

Figure 1.

Multiplying the vector $x \in R^m$ that records our original scene by the $(m-2) \times m$ matrix

$$C = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 1 & \cdots & 0 \\ 0 & 0 & 1 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

models this blurring effect. The camera image $y = Cx$ that results from "photographing" the scene in Figure 1a with the camera modeled by the matrix C is printed in Figure 1b.

If z is a second scene, computing $C(x + z)$ results in a "double exposure." The results are particularly surprising visually when z is in the null space of C. Basis vectors for the null space of C are $n_1^T = [0, 1, -1, 0, 1, -1, \ldots]$ and $n_2^T = [1, 0, -1, 1, 0, -1, \ldots]$. The scenes they represent are printed in Figure 2. They have very high local contrast (pixel to pixel oscillation) but any three consecutive pixels average to zero. The camera photographs them as uniform gray. Adding linear combinations of the null vectors to the original scene can change it dramatically but the camera image is unchanged. The scene in Figure 1c is obtained from the vector $v = x + n_1 + n_2$. But its photograph, $C(v) = C(x) = y$, is again the image in Figure 1b.

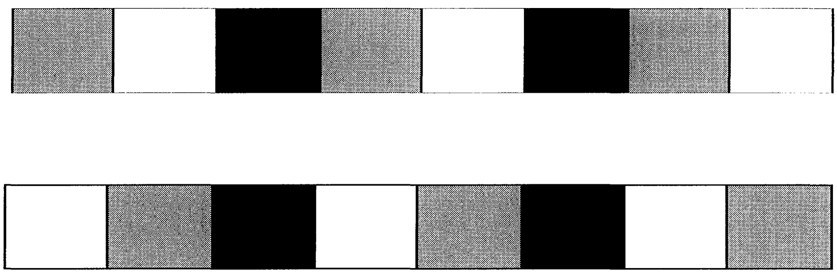


Figure 2.

Now, starting with the camera image y taken by our blurring camera C, we want to determine a reconstruction x_r of the original scene x. To solve the reconstruction problem (an inverse problem) we need to choose one solution from the infinitely many solutions to $Cx = y$. To do this, we must make some a priori assumptions about the properties of the reconstructed solution x_r. There are many reasonable strategies. (See [**1**].) To stay within the purview of elementary linear algebra, we choose the reconstructed solution x_r to be the (unique) vector that both solves the equation $y = Cx_r$ and is orthogonal to the null space of C. The choice seems reasonable for our simple blurring camera C: the reconstructed scene has no contribution from the visually noisy null vectors. The following is a simple procedure for solving the inverse problem. The solution x_r is orthogonal to the null space of C and therefore perpendicular to the basis vectors n_1 and n_2. Hence, x_r is the unique solution of the system:

$$\begin{bmatrix} C \\ n_1^T \\ n_2^T \end{bmatrix} \begin{bmatrix} \\ x_r \\ \\ \end{bmatrix} = \begin{bmatrix} y \\ 0 \\ 0 \end{bmatrix}.$$

The scene printed in Figure 1d is the reconstruction obtained by letting y be the vector illustrated in Figure 1b and solving the above system for x_r.

We would obtain the same reconstruction x_r if we solved $Cx = y$ for the solution of minimal Euclidean norm. In each case, x_r is the projection of any particular solution x_p into the row space of C. In the context of least-square problems, linear algebra texts commonly introduce a matrix algebra approach to the column space projection problem: To find the solution x_0 that minimizes $\|Ax - b\|$ for an overdetermined system $Ax = b$ where A has full rank, multiply b by the pseudo-inverse of A so that $x_0 = (A^T A)^{-1} A^T b$. The projection of b into the column space of A is Ax_0. (See [**3**].) Students seldom see the counterpart for an underdetermined system $Cx = y$: To project a particular solution x_p of $Cx = y$ into the row space of C where C has full rank, let $A = C^T$ and project x_p into the column space of A so that $x_r = A(A^T A)^{-1} A^T x_p$. Since $A^T = C$ and $C x_p = y$, we have $x_r = C^T (CC^T)^{-1} y$. The matrix $C^T (CC^T)^{-1}$ is the generalized inverse of C. We denote it by C^+ (cf. [**2**]). Our reconstruction procedure reduces to $x_r = C^+ y$.

Next we step up to two-dimensional scenes with the pixel values now recorded in $m \times n$ arrays. Imagine that we use a simple camera that receives and averages the pixel values of a square of four adjacent pixels. The photograph of an $m \times n$ scene stored in matrix A is stored in the $(m-1) \times (n-1)$ matrix B where $b_{i,j} = \frac{1}{4}(a_{i,j} + a_{i+1,j} + a_{i,j+1} + a_{i+1,j+1})$. (See Figure 3.) A student might discover the following line of reasoning to produce a matrix C so that $B = CA$. Regard the original $m \times n$ scene as an $mn \times 1$ column vector x whose entries are the transpose of the first row of A, followed by the transpose of the second row of A, etc.. Construct the $(n-1) \times n$ matrix M and the $(m-1) \times m$ matrix S as follows:

$$M = \frac{1}{4}\begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \qquad S = \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

The camera matrix is the $(n-1)(m-1) \times nm$ partitioned matrix

$$C = \begin{bmatrix} M & M & 0 & \cdots & 0 \\ 0 & M & M & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & M \end{bmatrix}$$

obtained by replacing the 1's in S by the matrix M and the zeros in S by the $(n-1) \times n$ zero matrix $\mathbf{0}$. The camera image, $y = C x$, is an $(m-1)(n-1) \times 1$ vector that can be rearranged into an $(m-1) \times (n-1)$ array B.

The task of finding C^+ can be reduced to finding the generalized inverses of smaller matrices M and S. In general, suppose that $S = (s_{ij})$ and M are matrices of full rank with more columns than rows. Let P be the partitioned matrix

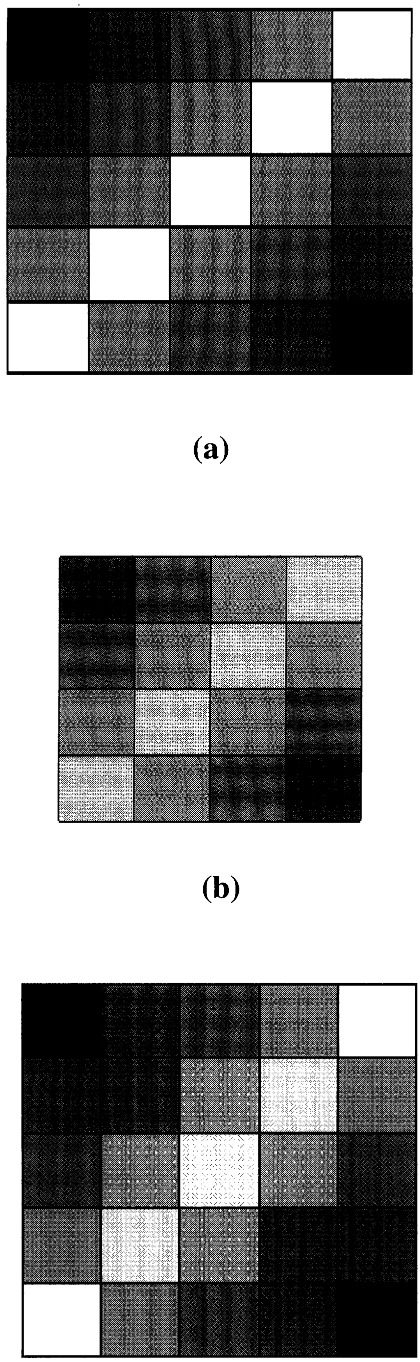
$$P = \begin{bmatrix} s_{11}M & s_{12}M & \cdots & s_{1m}M \\ s_{21}M & s_{21}M & \cdots & s_{2m}M \\ \cdots & \cdots & s_{ij}M & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}.$$

If $S^+ = (g_{ij})$ is the generalized inverse of S and M^+ is the generalized inverse of M, then:

$$P^+ = \begin{bmatrix} g_{11}M^+ & g_{12}M^+ & \cdots & g_{1m}M^+ \\ g_{21}M^+ & g_{22}M^+ & \cdots & g_{2m}M^+ \\ \cdots & \cdots & g_{ij}M^+ & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}.$$

In the case that M, S and C are

$$M = \frac{1}{4}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \qquad S = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix},$$



(a)

(b)

(c)

Figure 3.

$$C = \begin{bmatrix} M & M & 0 & 0 & 0 \\ 0 & M & M & 0 & 0 \\ 0 & 0 & M & M & 0 \\ 0 & 0 & 0 & M & M \end{bmatrix},$$

we have $S = 4M$ and $M^+ = 4S^+$ reducing the problem still further. Thus

$$S^+ = \frac{1}{5}\begin{bmatrix} 4 & -3 & 2 & -1 \\ 1 & 3 & -2 & 1 \\ -1 & 2 & 2 & -1 \\ 1 & -2 & 3 & 1 \\ -1 & 2 & -3 & 4 \end{bmatrix} \quad \text{and} \quad C^+ = \frac{1}{5}\begin{bmatrix} 4M^+ & -3M^+ & 2M^+ & -M^+ \\ M^+ & 3M^+ & -2M^+ & M^+ \\ -M^+ & 2M^+ & 2M^+ & -M^+ \\ M^+ & -2M^+ & 3M^+ & M^+ \\ -M^+ & 2M^+ & -3M^+ & 4M^+ \end{bmatrix}.$$

We used C^+ to find the reconstructed image in Figure 3c.

We graduate to a 31×31 image in Figure 4. The camera matrix C is 900×961 but to compute C^+, we needed only to calculate the generalized inverse of a 30×31 matrix.

How do we store and process data for much larger scenes? How can we reasonably overlap scenes? The topics introduced here provoke many questions that can spark and sustain productive student research.

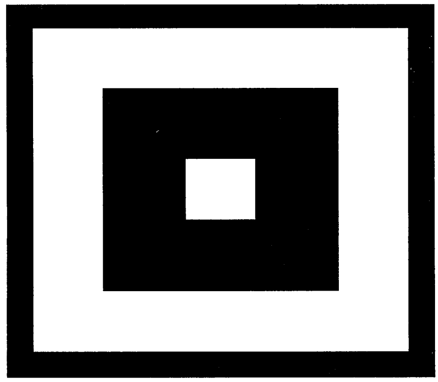
References

1. W. Menke, *Geophysical data analysis*, Academic Press, 1984; p 183.
2. G. Strang, *Linear Algebra and its Applications*, (3rd ed), Harcourt Brace Jovanovich, 1988.
3. A. Tucker, *Linear Algebra*, MacMillan, 1993.

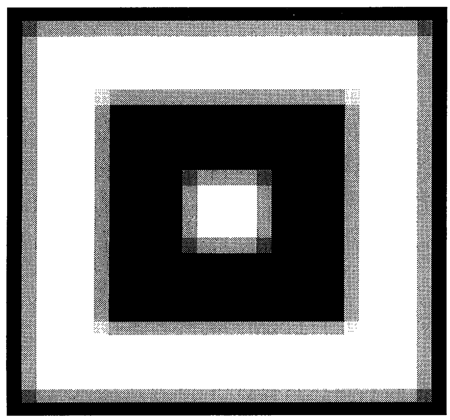
It's About Time!

From the Greencastle (Indiana) *Banner-Graphic*, November 17, 2000, page 16A:

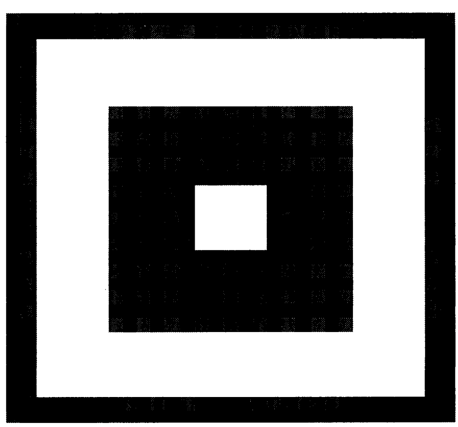
Indiana State Police cracking down on math lab operations



(a)



(b)



(c)

Figure 4.