Digital Photographs and Matrices

- Digital Photographs
- Color Model for 24-bit
- Visualization of Matrix Addition
- Visualization of Matrix Scalar Multiplication
- Color Separation Illustration
- Decoding with a Computer Algebra System
- The Checkerboard Visualization

Digital Photographs

A digital camera stores image sensor data as a matrix A of numbers corresponding to the color and intensity of tiny sensor sites called **pixels** or **dots**. The pixel position in the print is given by row and column location in the matrix A.

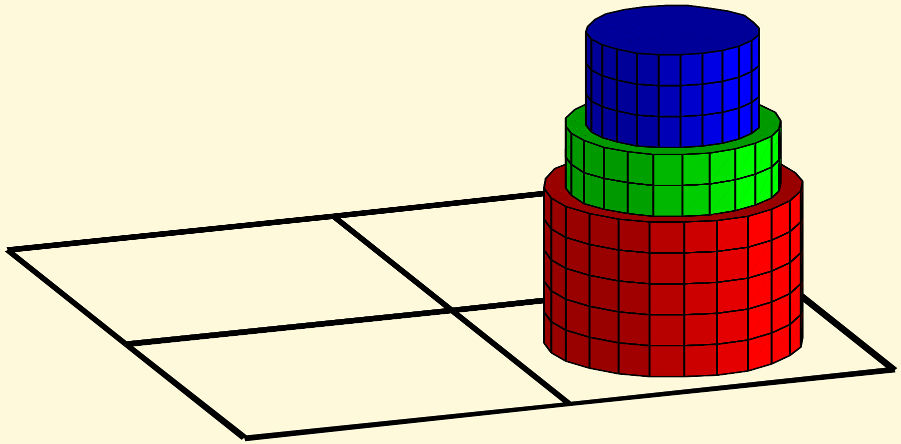

Figure 1. Checkerboard visualization.

Illustrated is a stack of checkers, representing one photodiode site on an image sensor inside a digital camera. There are 5 red, 2 green and 3 blue checkers stacked on one square. The checkers represent the number of electrons knocked loose by photons falling on each RGB-filtered site.

Color Model for 24-bit

In $\mathbf{24}$-bit color, a pixel could be represented in matrix $\boldsymbol{A}$ by a coded integer

$$\boldsymbol{a = r + (2^8)g + (2^{16})b.}$$

Symbols $\boldsymbol{r}$, $\boldsymbol{g}$, $\boldsymbol{b}$ are integers between $\mathbf{0}$ and $\mathbf{255}$ which represent the intensity of colors red, green and blue, respectively. For example, $\boldsymbol{r = g = b = 0}$ is the color **black** while $\boldsymbol{r = g = b = 255}$ is the color **white**. Grander schemes exist, e.g., $\mathbf{32}$-bit and $\mathbf{128}$-bit color.[a]

---

[a] A typical beginner's digital camera makes low resolution color photos using $\mathbf{24}$-bit color. The photo is constructed of $\mathbf{240}$ rows of dots with $\mathbf{320}$ dots per row. The associated storage matrix $\boldsymbol{A}$ is of size $\mathbf{240 \times 320}$. The identical small format is used for video clips at up to 30 frames per second in video-capable digital cameras.

The storage format **BMP** stores data as bytes, in groups of three $\boldsymbol{b}$, $\boldsymbol{g}$, $\boldsymbol{r}$, starting at the lower left corner of the photo. Therefore, $\mathbf{240 \times 320}$ photos have $\mathbf{230,400}$ data bytes. The storage format **JPEG** reduces file size by compression and quality loss.

Visualization of Matrix Addition

Matrix addition can be visualized through matrices representing color separations, a technique invented by James Clerk Maxwell.

When three monochrome transparencies of colors red, green and blue (RGB) are projected simultaneously by a projector, the colors add to make a full color screen projection.

The three transparencies can be associated with matrices $\boldsymbol{R}$, $\boldsymbol{G}$, $\boldsymbol{B}$ which contain pixel data for the monochrome images. Then the projected image is associated with the matrix sum $\boldsymbol{R} + \boldsymbol{G} + \boldsymbol{B}$.

Visualization of Matrix Scalar Multiplication

Scalar multiplication of matrices has a similar visualization.

The pixel information in a monochrome image (red, green or blue) is coded for intensity.

The associated matrix $\boldsymbol{A}$ of pixel data when multiplied by a scalar $\boldsymbol{k}$ gives a new matrix $\boldsymbol{kA}$ of pixel data with the intensity of each pixel adjusted by factor $\boldsymbol{k}$.

The photographic effect is to adjust the range of intensities. In the checkerboard visualization of an image sensor, factor $\boldsymbol{k}$ increases or decreases the checker stack height at each square.

Color Separation Illustration

Consider the coded matrix

$$\mathbf{X} = \begin{pmatrix} \mathbf{514} & \mathbf{3} \\ \mathbf{131843} & \mathbf{197125} \end{pmatrix}.$$

We will determine the monochromatic pixel data $\mathbf{R}$, $\mathbf{G}$, $\mathbf{B}$ in the equation $\mathbf{X} = \mathbf{R} + \mathbf{2^8 G} + \mathbf{2^{16} B}$.

First we decode the scalar equation $\boldsymbol{x = r + 2^8 g + 2^{16} b}$ by these algebraic steps, which use the modulus function $\mathbf{mod}(\boldsymbol{x, m})$, defined to be the remainder after division of $\boldsymbol{x}$ by $\boldsymbol{m}$. We assume $\boldsymbol{r}$, $\boldsymbol{g}$, $\boldsymbol{b}$ are integers in the range $\mathbf{0}$ to $\mathbf{255}$.

$\boldsymbol{y} = \mathbf{mod}(\boldsymbol{x}, \mathbf{2^{16}})$	The remainder should be $\boldsymbol{y = r + 2^8 g}$.
$\boldsymbol{r} = \mathbf{mod}(\boldsymbol{y}, \mathbf{2^{8}})$	Because $\boldsymbol{y = r + 2^8 g}$, the remainder equals $\boldsymbol{r}$.
$\boldsymbol{g = (y - r)/2^8}$	Divide $\boldsymbol{y - r = 2^8 g}$ by $\mathbf{2^8}$ to obtain $\boldsymbol{g}$.
$\boldsymbol{b = (x - y)/2^{16}}$	Because $\boldsymbol{x - y = x - r - 2^8 g}$ has remainder $\boldsymbol{b}$.
$\boldsymbol{r + 2^8 g + 2^{16} b}$	Answer check. This should equal $\boldsymbol{x}$.

Decoding with a Computer Algebra System

Computer algebra systems can provide an answer for matrices $\boldsymbol{R}$, $\boldsymbol{G}$, $\boldsymbol{B}$ by duplicating the scalar steps. Below is a `maple` implementation that gives the answers

$$\boldsymbol{R} = \begin{pmatrix} \mathbf{2} & \mathbf{3} \\ \mathbf{3} & \mathbf{5} \end{pmatrix}, \boldsymbol{G} = \begin{pmatrix} \mathbf{2} & \mathbf{0} \\ \mathbf{3} & \mathbf{2} \end{pmatrix}, \boldsymbol{B} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{2} & \mathbf{3} \end{pmatrix}.$$

```
with(LinearAlgebra:-Modular):
X:=Matrix([[514,3],[131843,197125]]);
Y:=Mod(2^16,X,integer); # y=mod(x,65536)
R:=Mod(2^8,Y,integer);  # r=mod(y,256)
G:=(Y-R)/2^8;           # g=(y-r)/256
B:=(X-Y)/2^16;          # b=(x-y)/65536
X-(R+G*2^8+B*2^16);     # answer check
```

The Checkerboard Visualization

The result can be visualized through a checkerboard of 4 squares. The second square has 5 red, 2 green and 3 blue checkers stacked, representing the color $x = (5) + 2^8(2) + 2^{16}(3)$ - see Figure 1. A matrix of size $m \times n$ is visualized as a checkerboard with mn squares, each square stacked with red, green and blue checkers.

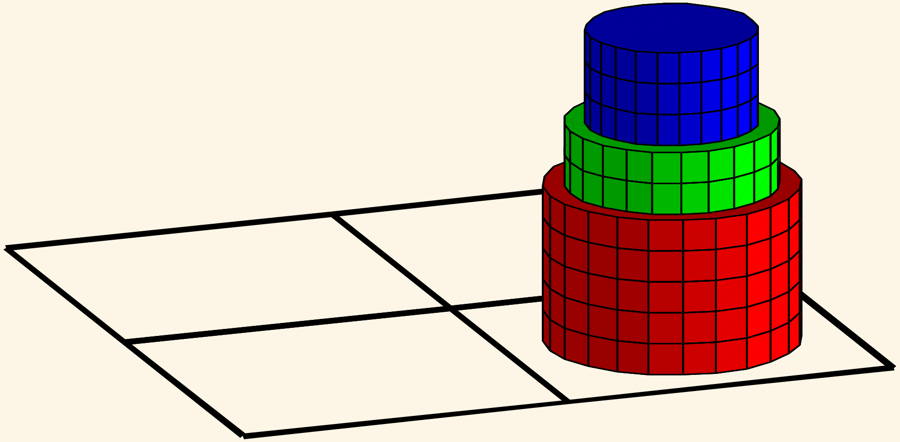

Figure 2. Checkerboard visualization.

Illustrated is a stack of checkers, representing one photodiode site on an image sensor inside a digital camera. There are 5 red, 2 green and 3 blue checkers stacked on one square. The checkers represent the number of electrons knocked loose by photons falling on each RGB-filtered site.