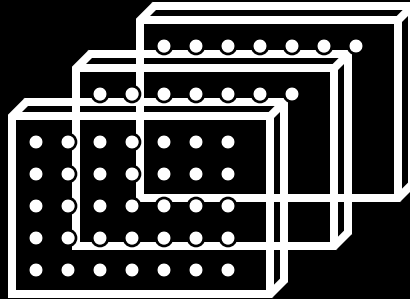


# MULTIDIMENSIONAL ARRAYS



Handling events in time  
and space

# Why multidimensional arrays?

- More complex and interesting data are frequently multi-D.
- Examples are
  - Medical images(space and time)
  - Digital video(space and time)
  - Color images(R,G,B)

# RGB space

- Here is a multidimensional array!

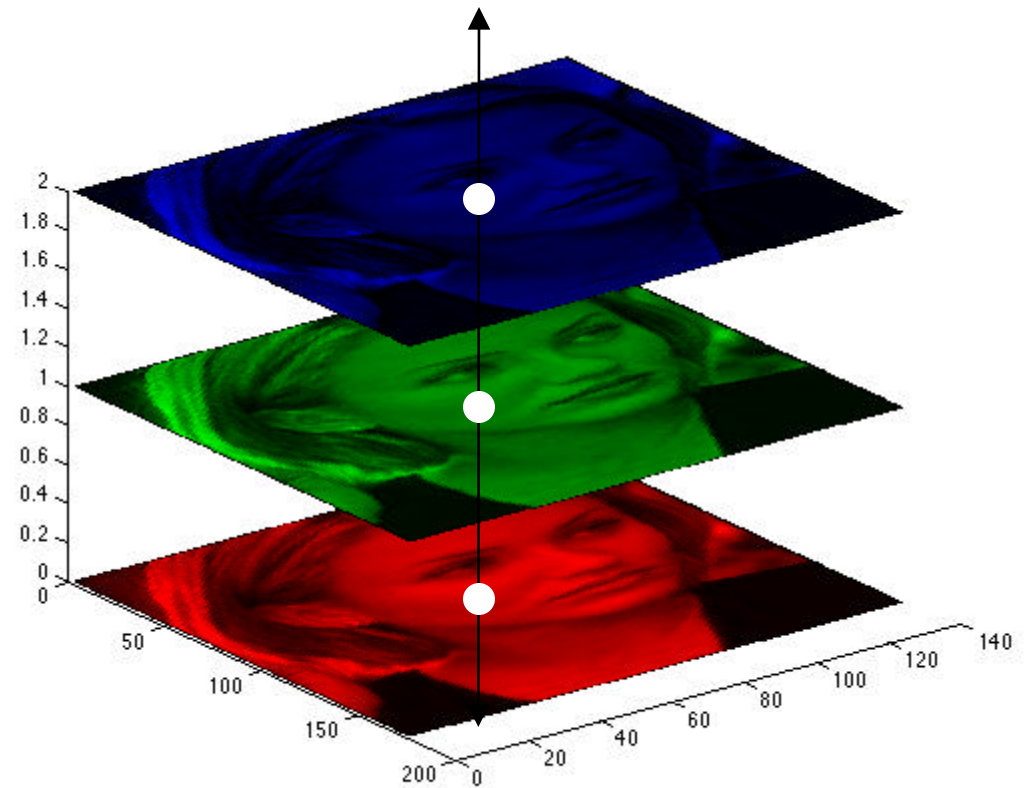


# Displaying Images

- Load the image into an array `X`
  - If the image is already in `.mat` format (native MATLAB data type) just type `load filename.mat`
  - We will cover other file formats, such as jpeg, later on.
- Check your workspace for the name of variable holding image data
- Type `image(X)` where `X` is the array name

# 3-D array

- Red, green  
And blue  
Pixels each  
Define a  
dimension



# Images in Time

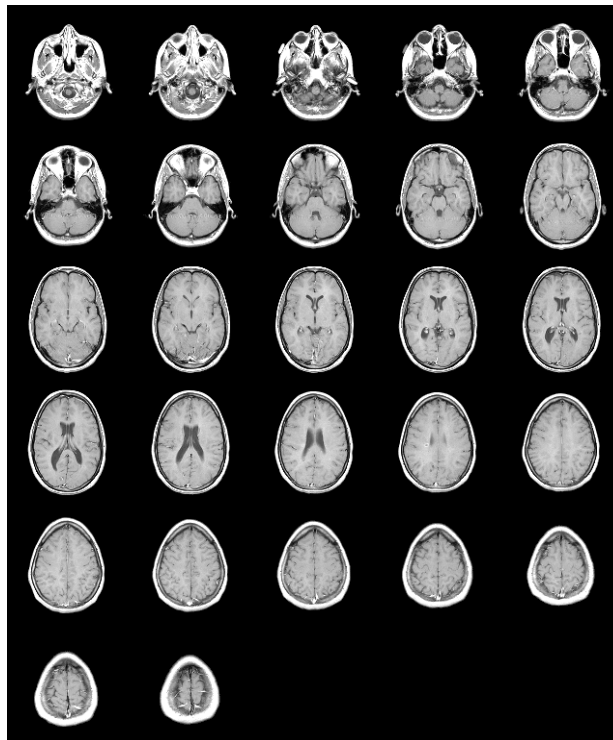
Space



Time

# Medical Imaging

- Each frame is 128x128 and there are 27 of them

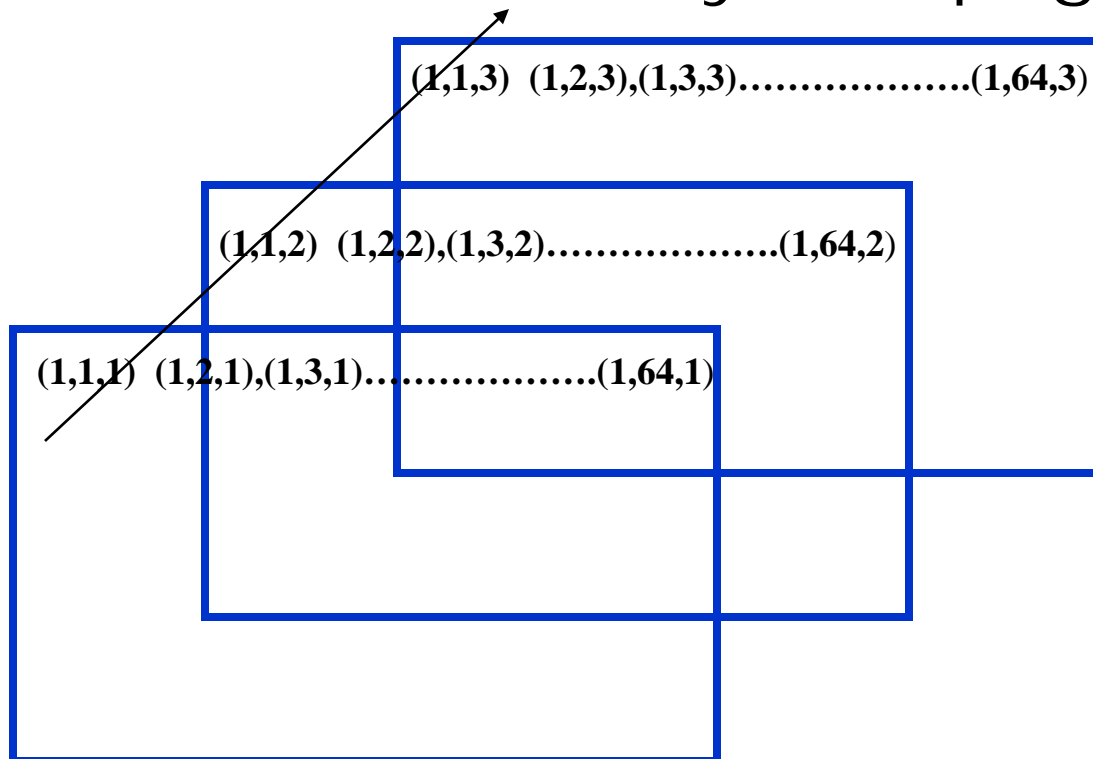


# Forming MD Arrays

- The syntax for the MD array is a logical extension of 1D and 2D arrays
- Here in addition to rows and columns we have **pages**

# 3-D array Structure: row, column and now page

- An  $n \times m \times 3$  array of 3 pages



# Creating MD arrays

- Start with a 3x3 matrix
  - $A = [5 \ 7 \ 8; 0 \ 1 \ 9; 4 \ 3 \ 6]$
- Add a second page
  - $A(:,:,2) = [1 \ 0 \ 4; 3 \ 5 \ 6; 9 \ 8 \ 7]$
- A now is 3x3x2

# Extending MD Arrays

- We can logically extend the array in any dimension, row, column or page
- For example, to add a 3rd page
  - $A(:,:,3)=5$
  - Results in a 3rd page of 3x3, all 5's

# Generating MD Arrays using functions

- Many MATLAB functions can produce MD arrays
- `a=rand(4,3,2)` gives

`a(:,:,1) =`

0.1389	0.2722	0.4451
0.2028	0.1988	0.9318
0.1987	0.0153	0.4660
0.6038	0.7468	0.4186

`a(:,:,2) =`

0.8462	0.8381	0.8318
0.5252	0.0196	0.5028
0.2026	0.6813	0.7095
0.6721	0.3795	0.4289

# Concatenate

- You can make larger matrices using cat function.
- $B = \text{cat}(\text{dim}, A1, A2, A3)$  concatenates A's into a larger matrix along dimension dim

# Example cat Usage

- $B = \text{cat}(3, [2 \ 8; 0 \ 5], [1 \ 3; 7 \ 9])$ , then

$$B(:, :, 1) =$$

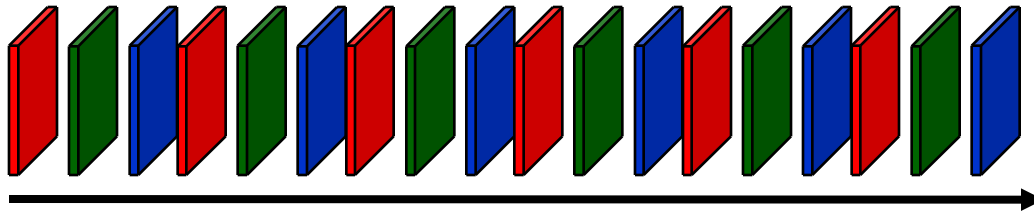
$$\begin{array}{cc} 2 & 8 \\ 0 & 5 \end{array}$$

$$B(:, :, 2) =$$

$$\begin{array}{cc} 1 & 3 \\ 7 & 9 \end{array}$$

# 4th Dimension

- What is the interpretation of 4D data?
- Think of color digital video.



- Dim 1 :rows of a single frame
- Dim 2: columns of a single frame
- Dim 3: R, G, and B, planes(pages)
- Dim 4: time

# Some numbers

- One minute video at 30 frames/sec.
  - Frame: 120 rowsx160 columns
  - 3 color planes
  - 1800 frames or pages
- The blue component of pixel at (100,50) 10 seconds into the video is  
$$A(100,50,3,300)$$

# Working with MD Arrays

- Basic indexing rules apply
  - $A(3,2,2)$  is element (3,2) on page 2
  - $A(2, [1\ 2], 3)$  reads as row 2, columns 1,2 of page 3

$a(:,:,1) =$

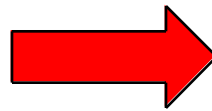
3	5	8
2	7	5
3	3	3

$a(:,:,2) =$

7	6	9
5	6	5
4	7	8

$a(:,:,3) =$

1	2	1
9	8	0
2	7	8



# Use of semi-colon

- $A(:,3,2)$  retrieves every row of column 3 on page 2

$a(:,:,1) =$

3 5 8  
2 7 5  
3 3 3

$a(:,:,2) =$

7 6 9  
5 6 5  
4 7 8

$a(:,:,3) =$

1 2 1  
9 8 0  
2 7 8

# Reshaping

- Reshape command works as before
  - $B = \text{reshape}(A, [s1, s2, s3, \dots])$
  - A is cast into another matrix of dimensions  $s1, s2, s3$  and so on
- B must have the same number of elements as A

# Reshaping Example

$a(:,:,1) =$

```
3 5 8
2 7 5
3 3 3
```

$a(:,:,2) =$

```
7 6 9
5 6 5
4 7 8
```

$a(:,:,3) =$

```
1 2 1
9 8 0
2 7 8
```

●  $B = \text{reshape}(a, [9, 3])$

$b =$

```
3 7 1
2 5 9
3 4 2
5 6 2
7 6 8
3 7 7
8 9 1
5 5 0
3 8 8
```

# Operations on MD Arrays

- Element-by-element operations such as `.*`, `./` etc. work as before

`a(:,:,1) =`

1 1  
1 1

`a.*a=`

`a(:,:,2) =`

2 2  
2 2

`a(:,:,3) =`

3 3  
3 3

`ans(:,:,1) =`

1 1  
1 1

`ans(:,:,2) =`

4 4  
4 4

`ans(:,:,3) =`

9 9  
9 9

`ans(:,:,1) =`

1 1  
1 1

`ans(:,:,2) =`

4 4  
4 4

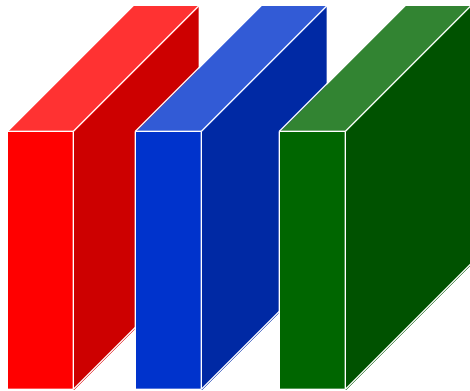
`ans(:,:,3) =`

27 27  
27 27

`a.^a=`

# Using MD Arrays

- Switch blue and green channels of an RGB image. What is the effect on colors?



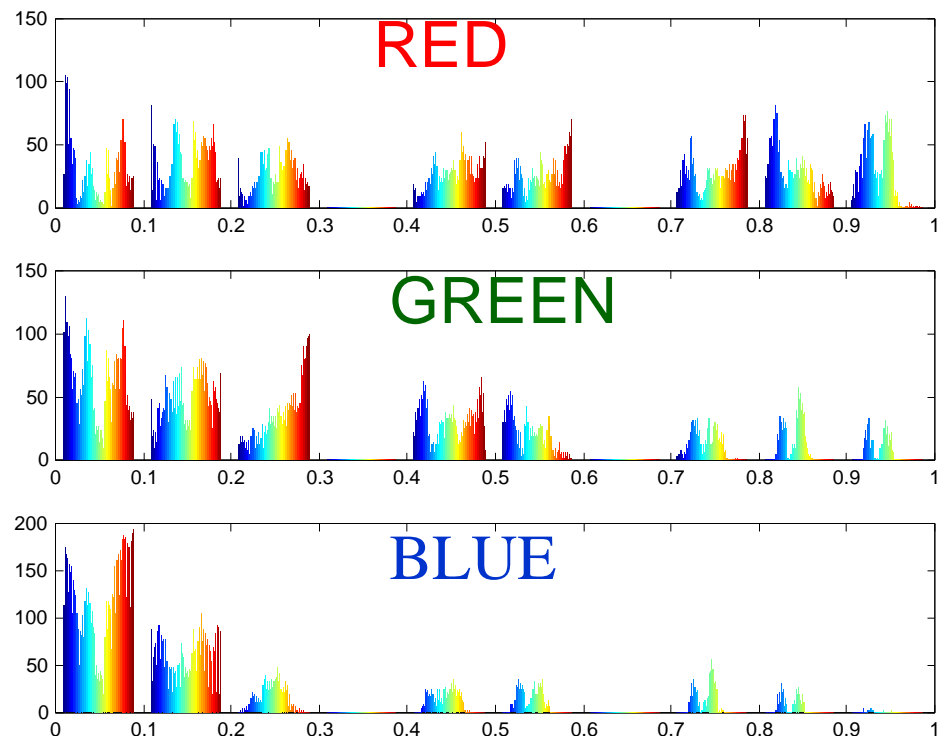
Swapped b&g channels



Original rgb order

# Color Histogram

- Histogram intensities of R,G and B channels
- Plot shows lots of red, dark and medium greens and lots of dark blues



# Hands-on Exercises

- Load and display clown.mat
  - Check your workspace and verify the 3-D nature of data
  - Print a 10x10 chunk of data from page 3 and look at the numbers
  - Find the mean intensity value of each channel and compare (use *mean*)
  - Do what is asked on slide 22
- Load and display diaz.mat
  - Produce color histogram of diaz and display them as shown on slide 23

# HOMework

- Load clown
- Find the mean of red, green and blue channels
- Find pixel positions with the lightest red, blue or green.
- What are those minimum intensities
- For some reason, pixels with blue intensities  $>0.5$  are of interest. Find them and replace them with 100% white. Display the new image.
- Pixels with red  $>90\%$  **and** blue  $>75\%$  are of interest. Find and replace them with 100% white and watch the result.