

# DISPLAY Lab

Εργ. Ψηφιακής Επεξεργασίας Σήματος & Εικόνας  
Πολυτεχνείο Κρήτης - HMMY

## ΤΗΛ312 – Ψηφιακή Επεξεργασία Εικόνας

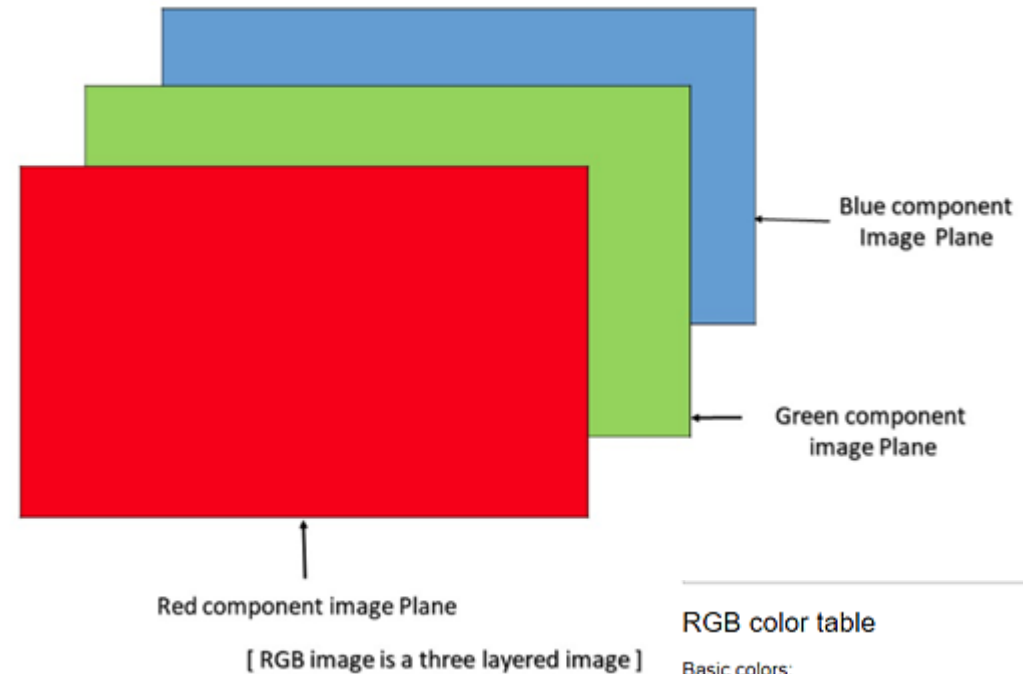
1<sup>η</sup> εργ. άσκηση:  
Χρωματικά Μοντέλα

Ακ. Έτος: 2018 – 2019

Επιμέλεια: Ντίνα Μοιρογιώργου







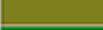



# Indexed RGB Εικόνες

- Μία RGB εικόνα μπορεί να θεωρηθεί ως μία 'στοίβα' 3 εικόνων, τη μία πάνω στην άλλη: Red, Green & Blue component.
- Στη Matlab, μία RGB εικόνα είναι ουσιαστικά ένας  $M * N * 3$  πίνακας, του οποίου η κάθε θέση (colour pixel) σχετίζεται με τρεις τιμές που αντιστοιχούν στο red, green & blue colour component.



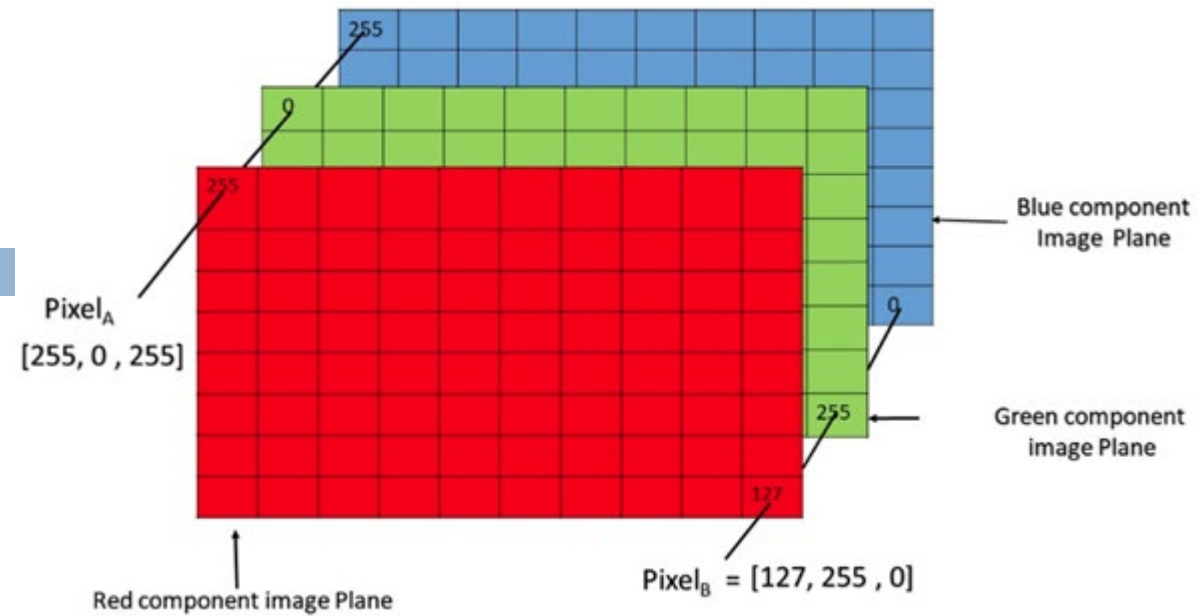
RGB color table

Basic colors:

Color	HTML / CSS Name	Hex Code #RRGGBB	Decimal Code (R,G,B)
	Black	#000000	(0,0,0)
	White	#FFFFFF	(255,255,255)
	Red	#FF0000	(255,0,0)
	Lime	#00FF00	(0,255,0)
	Blue	#0000FF	(0,0,255)
	Yellow	#FFFF00	(255,255,0)
	Cyan / Aqua	#00FFFF	(0,255,255)
	Magenta / Fuchsia	#FF00FF	(255,0,255)
	Silver	#C0C0C0	(192,192,192)
	Gray	#808080	(128,128,128)
	Maroon	#800000	(128,0,0)
	Olive	#808000	(128,128,0)
	Green	#008000	(0,128,0)
	Purple	#800080	(128,0,128)
	Teal	#008080	(0,128,128)
	Navy	#000080	(0,0,128)

# Indexed 8-bit RGB Εικόνας

- Όπως φαίνεται στην εικόνα, το  $\text{Pixel}_A$  έχει την τιμή (255,0,255) η οποία καθορίζεται από το συνδυασμό των εντάσεων χρώματος (Intensities) των Red, Green & Blue component.



Εάν θεωρήσουμε έναν πίνακα εικόνας RGB, έστω  $I$ , τότε:  
 $I(:, :, 1)$  – Red colour component (plane) της εικόνας RGB  
 $I(:, :, 2)$  – Green colour component (plane) της εικόνας RGB  
 $I(:, :, 3)$  – Blue colour component (plane) της εικόνας RGB

## Bit Depth:

Ο αριθμός των bits που χρησιμοποιούνται για την αποθήκευση ενός pixel καθορίζει το bit depth μίας RGB εικόνας. Π.χ. εάν κάθε χρωματικό κανάλι είναι μία 8-bit εικόνα, τότε η RGB εικόνα έχει βάθος 24 bits.

Εάν η RGB εικόνα είναι τύπου 'double', τότε κάθε χρωματικό κανάλι παίρνει τιμές στο διάστημα  $[0, 1]$ .

Εάν είναι τύπου 'uint8', το αντίστοιχο διάστημα τιμών είναι το  $[0, 255]$ , ενώ για RGB εικόνες τύπου 'uint16' είναι  $[0, 65535]$

# Διάβασμα παλέτας 8-bit RGB Εικόνας (Matlab)

- ❑ `clc; clear all; close all;`
- ❑ `[X,map]=imread('windmill.bmp');`
- ❑ `X_new=zeros(256);`
- ❑ `for i=1:length(X_new)`
- ❑     `X_new(i,:)=i;`
- ❑ `end`
- ❑ `imwrite(X_new, map, 'paletta.bmp');`

# Διάβασμα παλέτας 8-bit RGB Εικόνας (Matlab)

The screenshot shows the MATLAB Editor with a file named 'PSEE1.m'. The main window displays a 344x510 uint8 image 'X'. The 'Variables' window on the right shows the following variables:

Name	Value
i	256
map	256x3 double
X	344x510 uint8
X_new	256x256 double

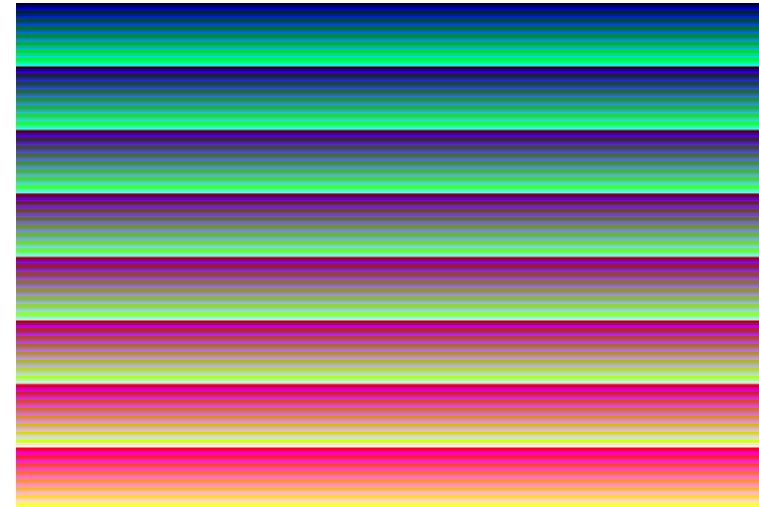
The screenshot shows the MATLAB Editor with a file named 'PSEE1.m'. The main window displays a 256x3 double image 'map'. The 'Variables' window on the right shows the following variables:

Name	Value
i	256
map	256x3 double
X	344x510 uint8
X_new	256x256 double

The screenshot shows the MATLAB Editor with a file named 'PSEE1.m'. The main window displays a 256x256 double image 'X\_new'. The 'Variables' window on the right shows the following variables:

Name	Value
i	256
map	256x3 double
X	344x510 uint8
X_new	256x256 double

# Διάβασμα παλέτας 8-bit RGB Εικόνας (Matlab)

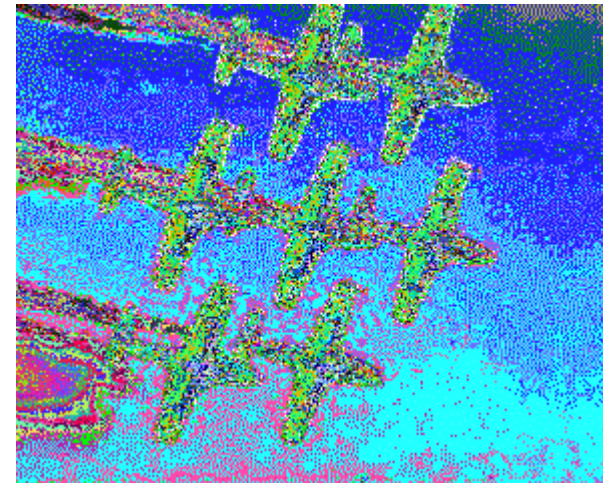
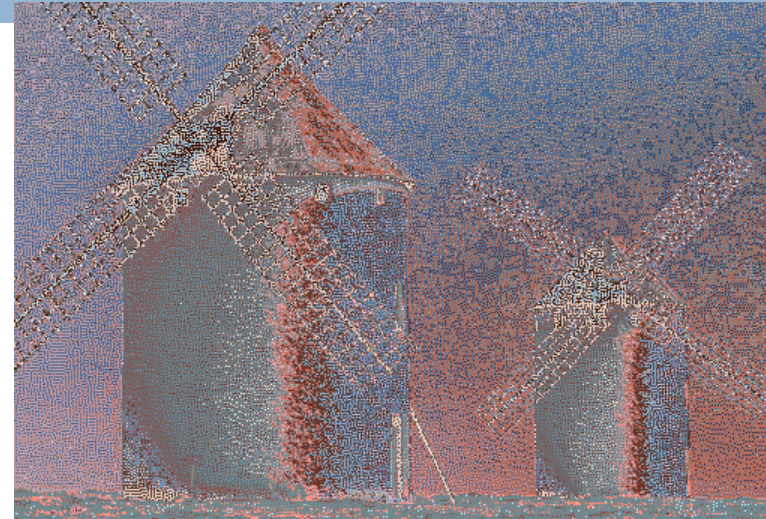




# Ανταλλαγή Παλέτας - swap palettes (Matlab)

- % Read the images
- [A,mapA] = imread('windmill.bmp');
- [B,mapB] = imread('formation.bmp');
  
- % Swap the colormaps of the images
- imwrite(A,mapB,'image1\_swapped.bmp')
- imwrite(B,mapA,'image2\_swapped.bmp')

# Ανταλλαγή Παλέτας - swap palettes (Matlab)





## HSI **2** RGB Color Model

1)  $A \vee I \leq 0.5 \quad M = I(1 + S)$

$A \vee I > 0.5 \quad M = I + S - IS$

2)  $m = 2I - M$

### Υπολογισμός του R:

□  $H < 60 \quad R = m + (M - m)(H/60)$

□  $H < 180 \quad R = M$

□  $H < 240 \quad R = m + (M - m)$

□  $H < 360 \quad R = m$

# HSI **2** RGB Color Model

## Υπολογισμός του G:

- $H < 120$        $G = m$
- $H < 180$        $G = m + (M - m)$
- $H < 300$        $G = M$
- $H < 360$        $G = m + (M - m)$

## Υπολογισμός του B:

- $H < 60$        $B = M$
- $H < 120$        $B = m + (M - m)$
- $H < 240$        $B = m$
- $H < 300$        $B = m + (M - m)$
- $H < 360$        $B = M$

# HSI 2 RGB Color Model

- Αντίθετα με το χρωματικό μοντέλο RGB, το HSI μοντέλο διαχωρίζει την ένταση χρώματος (image intensity) από την απόχρωσή του ή αλλιώς από τη χρωματική πληροφορία.
- Στη Μηχανική Όραση, συχνά απαιτείται διαχωρισμός των χρωματικών συνιστωσών από την πληροφορία της έντασής τους. Παραδείγματα όπου αυτή η ενέργεια απαιτείται είναι:
  - ▣ robustness to lighting changes
  - ▣ removing shadows

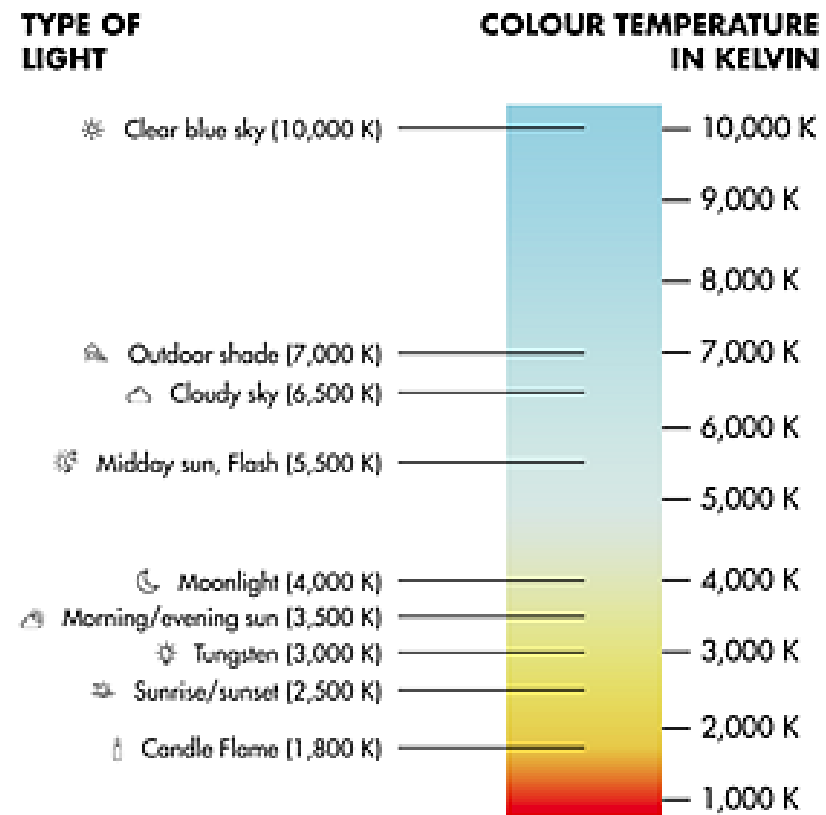
# White balance

- Το χρώμα ενός αντικειμένου επηρεάζεται από τις συνθήκες φωτισμού υπό τις οποίες το παρατηρούμε.
- Το ανθρώπινο μάτι και ο εγκέφαλος 'αντισταθμίζουν' το χρώμα για διαφορετικές συνθήκες φωτισμού. Αυτός είναι και ο λόγος που το λευκό είναι αντιληπτό ως το ίδιο είτε οι συνθήκες είναι εξωτερικές και ηλιόλουστες, είτε υπό σκιά ή συννεφιά, είτε ακόμη σε κλειστούς χώρους με ελλιπή φωτισμό.
- Η τεχνική του white balance method βοηθά στην προσομοίωση αυτής της διαδικασίας αντιστάθμισης των διαφορετικών τύπων φωτισμού ώστε ένα λευκό αντικείμενο να καθίσταται όντως λευκό.



# White balance

- Κάθε 'πηγή φωτός' έχει ένα χρώμα το οποίο αντιστοιχεί σε μία θερμοκρασία. Η κλίμακα για το color temperature scale μετράτε σε Kelvin.



# White balance



# White balance

Ορίζουμε μία εικόνα  $RGB\_image(x,y)$ , μεγέθους  $n \times n$ , όπου τα  $x$  και  $y$  υποδηλώνουν τη θέση του pixel.

Αρχικά, υπολογίζουμε τα:

- $R_{avg} = \frac{1}{n^2} \sum_{x=1}^n \sum_{y=1}^n R\_channel(x,y)$
- $G_{avg} = \frac{1}{n^2} \sum_{x=1}^n \sum_{y=1}^n G\_channel(x,y)$
- $B_{avg} = \frac{1}{n^2} \sum_{x=1}^n \sum_{y=1}^n B\_channel(x,y)$

Εάν οι τρεις αυτές τιμές των μέσων όρων του κάθε χρωματικού καναλιού είναι ίσες, τότε ικανοποιείται το προαναφερθέν κριτήριο (gray world assumption) και δεν απαιτείται περαιτέρω διόρθωση χρώματος.

Εάν δεν ικανοποιείται, που είναι και η πιο πιθανή περίπτωση, προχωρούμε με τους ακόλουθους υπολογισμούς:

- $R\_channel\_new(x,y) = \frac{G_{avg}}{R_{avg}} * R\_channel(x,y)$
- $G\_channel\_new(x,y) = G\_channel(x,y)$
- $B\_channel\_new(x,y) = \frac{G_{avg}}{B_{avg}} * B\_channel(x,y)$