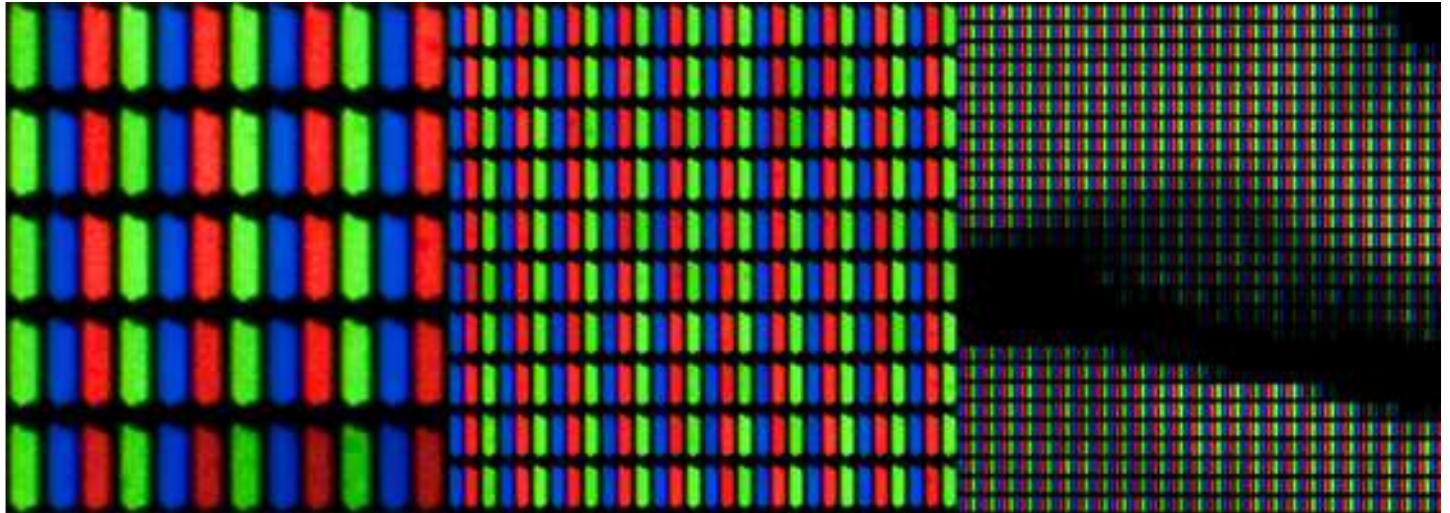


0_ la couleur numérique

Principe :

Chaque pixel est défini selon un principe de synthèse additive émise par 3 diodes élémentaires de couleur respective, Rouge, Verte et Bleue, définissant un pixel carré.

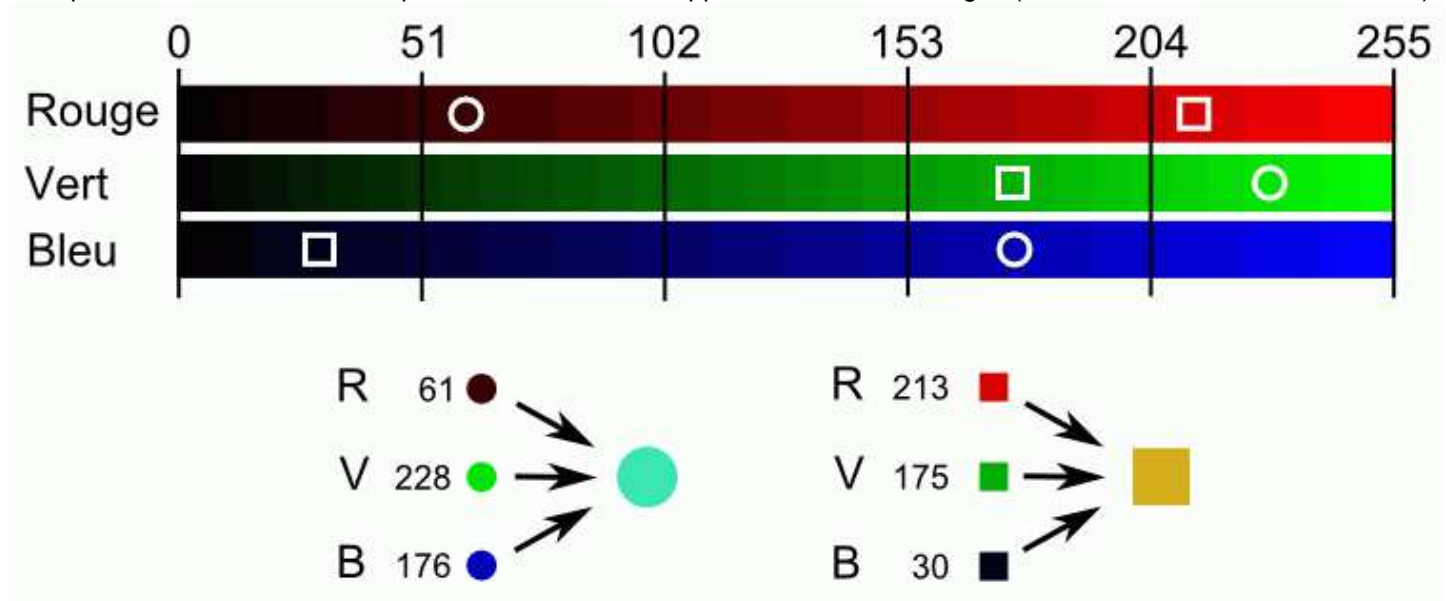


Des diodes élémentaires RVB à l'image numérique

Encodage des couleurs :

La quantité de lumière émise par chaque diode est définie par une valeur de 0 à 255. Cette valeur peut également être définie en binaire ou en hexadécimal.

Lorsque les 3 valeurs sont identiques, la couleur définie appartient à l'échelle de gris (du noir 000 au blanc 255-255-255)



exemple de définition de 2 couleurs par leur composantes RVB

Transparence :

Une valeur de 0 à 255 définit en plus un niveau de transparence.

Mode couleur :

Il existe plusieurs modes d'encodage de la couleur numérique par l'instruction `colorMode`.

SYNTAXE	EXEMPLE	EXPLICATION
<code>colorMode(RVB, R, V, B);</code>	<code>colorMode(RVB, 255, 255, 255);</code>	Définit l'espace couleur RVB
<code>colorMode(HSB, H, S, B);</code>	<code>colorMode(HSB, 360, 100, 100);</code>	Espace couleur TSL (Teinte Saturation Lumière)

Exemples :

→ [image_1_pixels_setting.pde](#) = définition de chacun des pixels comme une valeur de gris aléatoire

→ [image_1_pixels_image.pde](#) = redéfinition d'une image pixel par pixel en mettant à 0 les valeurs de rouge et vert.

1_ l'image comme tableau de pixels

Principe :

L'image est décomposée en un tableau de pixels. Chaque pixel peut alors être précisément défini par ses valeurs R,V,B.

Affichage / Stockage :

Si l'image est affichée comme le tableau à 2 dimensions des pixels de l'écran, les données de chaque pixel sont stockées dans un tableau à 1 dimension, manipulable par programme.

How the pixels look:

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

How the pixels are stored:

0	1	2	3	4	5	6	7	8	9	.	.	.		
---	---	---	---	---	---	---	---	---	---	---	---	---	--	--

Fonctions :

- loadPixels() : fonction qui crée un tableau de pixels de la taille de la fenêtre de résultat.
- updatePixels() : fonction appelée pour l'affichage quand tous les pixels sont définis.

Exemples :

→ [image_1_pixels_setting.pde](#) = définition de chacun des pixels comme une valeur de gris aléatoire

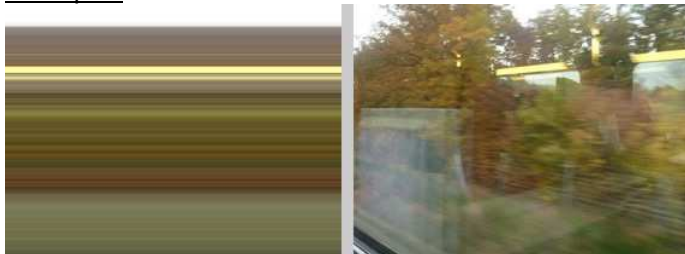
→ [image_1_pixels_image.pde](#) = redéfinition d'une image pixel par pixel en mettant à 0 les valeurs de rouge et vert.

3_ traitement d'image(s)

Il est dès lors possible de faire des calculs sur les pixels d'une image (photo) mais également d'un flux d'images (vidéo ou capture temps réel par une webcam, une caméra kinect, ...)

à partir d'une image fixe

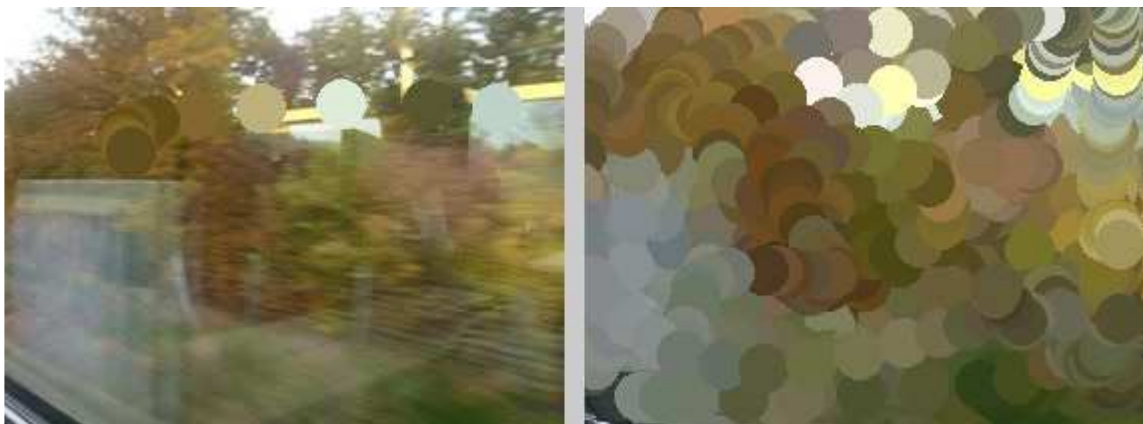
Exemples :



→ [photo_2_ligne_H.pde](#)



→ [photo_2_ligne_V.pde](#)



→ [photo_2_point_souris.pde](#) = redéfinition d'une image pixel par pixel en mettant à 0 les valeurs de rouge et vert.

à partir d'un flux d'images

Voir cours suivants : vidéo, webcam, caméra kinect