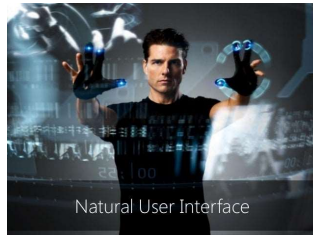


IV.4 – Interaction 2D basique : détecteur « simpliste »

- Rappel : Interaction 2D
 - Utilisation de la webcam pour détecter des « mouvements » de la part de l'utilisateur
- NUI (« Natural User Interface ») : « Air Gesture Interface »
 - capture indirecte (pas de capteur directement en contact avec l'utilisateur)
 - l'utilisateur peut agir/interagir naturellement

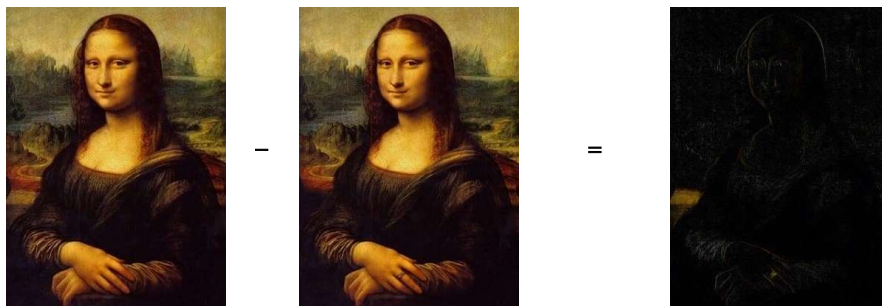


- Principe : comparaison des images du flux vidéo entre elles

62

IV.4 – Différences entre images

- « Jeu des erreurs », par comparaison d'images
 - Différences des couleurs de chacun des pixels des deux images



- Soustraction de chaque composante couleur des pixels, en mode RGB : Différences en couleur

```
diffPixelR = abs(red(pixelImg1) - red(pixelImg2));
diffPixelG = abs(green(pixelImg1) - green(pixelImg2));
diffPixelB = abs(blue(pixelImg1) - blue(pixelImg2));
```
- Création de l'image des différences, en couleur

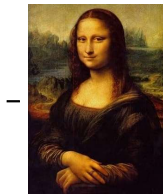

```
differenceImg.pixels[i] = color(diffPixelR, diffPixelG, diffPixelB);
```
- A noter : quantification du **taux de similitude** facilement calculable :
 - somme cumulée des diffPixel sur toute l'image

63

IV.4 – Simplification : niveaux de gris

- Simplification pour améliorer les traitements à suivre
 - Limiter la durée des calculs... pour augmenter le FrameRate (taux de rafraichissement)
- Passage en niveaux de gris
 - Différences exprimées en niveaux de gris : (R,G,B) => (G,G,G)
 - Création de l'image des différences, en couleur


```
diffPixel = diffPixelR + diffPixelG + diffPixelB; // ou la moyenne... => /3
differenceImg.pixels[i] = color(diffPixel, diffPixel, diffPixel);
```



Différences en gris

Différences en gris (moy.)

64

IV.4 – Exemple 8 : Jeu des différences

```
// Calcul de la difference entre deux images fixes, prechargees
...
PImage image1, image2, differenceImg; // Images
int nbPixels;
color pixelImg1, pixelImg2;
int diffPixel; // total
int diffPixelR, diffPixelG, diffPixelB; // par composante
...
differenceImg = createImage(image1.width, image1.height, RGB);
for (i = 0; i < nbPixels; i++) {
  pixelImg1 = image1.pixels[i];
  pixelImg2 = image2.pixels[i];

  // Calcul des differences sur les 3 composantes couleur : R, G et B
  diffPixelR = abs(((pixelImg1 >> 16) & 0xFF) - ((pixelImg2 >> 16) & 0xFF));
  // A noter : equivalent a : abs( red(pixelImg1) - red(pixelImg2) )
  diffPixelG = abs(((pixelImg1 >> 8) & 0xFF) - ((pixelImg2 >> 8) & 0xFF));
  diffPixelB = abs((pixelImg1 & 0xFF) - (pixelImg2 & 0xFF));

  diffPixel = diffPixelR + diffPixelG + diffPixelB;
  differenceImg.pixels[i] = color(diffPixelR, diffPixelG, diffPixelB); // Difference en couleur
}
differenceImg.updatePixels();
image(differenceImg, 0, 0); // Restitution de l'image des differences
}
```

65

IV.4 – Binarisation

□ Binarisation : passage en N&B (Noir & Blanc)

- A partir du niveaux de gris (G,G,G)
- Binarisation « centrée » : N ou B selon que G est inférieur ou supérieur à 128
 - **Seuil** (« Threshold »), généralement indiqué sous forme normalisée : seuil = 0,5 car 128/256
`differenceImg.filter(THRESHOLD)`
- Binarisation à seuil : N ou B selon la position par rapport au seuil, quelconque (de 0 à 1)
 - seuil fixe : difficile à régler
 - automatique : par exemple, valeur médiane du gris (séparant les pixels en deux populations égales) ou encore, analyse des histogrammes (nombre de pixels pour chaque niveau de gris)



Binarisation seuil 0,025



Binarisation seuil 0,1



Binarisation seuil 0,25



Binarisation seuil 0,3

66

IV.4 – Seuillage / postérisation

□ Autre variante des seuillages : postérisation

- Limitation du nombre de couleur possible, sur chaque canal (donc, 2 à 255)
- Exemples : Posterize de 2 à 4, appliqué sur l'image en niveaux de gris
 - `differenceImg.filter(THRESHOLD)`



Posterize 2



Posterize 3



Posterize 4

67

IV.4 – Filtrage

□ Retrait des « bruits »

- Bruit : pixels isolés, dus à des « parasites » :
 - Petite variation sur les capteurs (appareil photo, caméra)
 - Légers changement de luminosité, ...
- Filtrage par érosion : retrait des pixels aberrants
 - Chaque pixel de l'image est remplacé par le plus sombre de ses 8 voisins (matrice 3x3)
 - `differenceImg.filter(ERODE)`



Différences en gris

Filtrage d'érosion



Erode