VI – Expériences de Réalité Augmentée

86

86

VI.1 – Quelques rappels sur la RA

- Définition sur la Réalité Augmentée (RA) :
  - Information supplémentaire insérée **dans le point de vue de l'utilisateur** de la scène du monde réel, en temps réel.
  - Attention : « Réalité Enrichie » pour information non spatialement cohérente (non homogène).

Réalité Augmentée

Réalité Enrichie

87

87

VI.1 – RA idéale

- 3 propriétés (International Symposium of Augmented Reality, Azuma 1997) :
  - Combiner le réel et le virtuel
  - De manière interactive, en temps réel
  - En respectant l'homogénéité : point de vue caméras, 3D, couleurs

intégration arbitraire

correspondance caméras

cohérence spatiale

cohérence photométrique

Problème plus difficile que la réalité virtuelle !

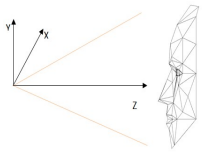
Source : Gilles Simon, La réalité augmentée, LORIA (France).

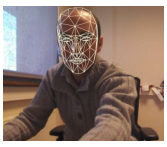
88

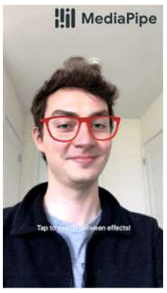
88

VI.2 – RA sans marqueur

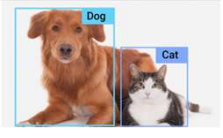
- Marker Less Tracking (MLT)
  - « Tracking » quelconque en 3D
    - Analyse d'image pour rechercher un modèle 3D pré-enregistré
    - Exemple : reconnaissance faciale (MediaPipe de Google : https://mediapipe.dev/)







  - « Natural Feature Tracking (NFT) »
    - Analyse d'image pour rechercher des « caractéristiques naturelles sur des surfaces planes texturées » (modèle 2D pré-enregistré)
    - Exemples : photos (visages, mains, …), logos colorés…



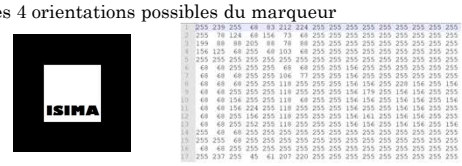
CLERMONT AUVERGNE INP Isima

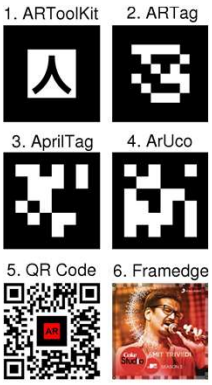
89

89

VI.3 – RA avec marqueur

- Avec marqueur : problème plus « simple »
  - Motif particulier en Noir & Blanc : « Fiducial Marker »
    - Carrés (par convention, de 8 cm de côté)
    - Bordurés de Noir (ARToolKit)
    - Fichier « .patt » (« Pattern ») : « simple » fichier texte donnant les 3 composantes R,G,B des 4 orientations possibles du marqueur





  - Reconnaissance « aisée » des marqueurs

Taille, donc, distance caméra

Orientation spatiale 3D

90

90

VI.4 – Réalité Augmentée avec ARToolKit

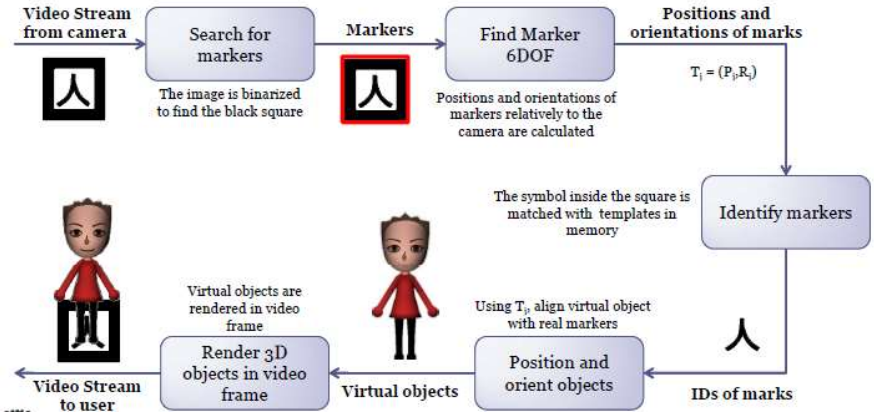
- Caractéristiques :
  - Bibliothèque gratuite de fonctions dédiées à la RA
    - développée à la base par Hirokazu Kato en 1999
    - de multiples variantes depuis…
  - Avec ou Sans Marqueurs : « Pattern », « NFT »
  - Détection de :
    - la position spatiale
    - l'orientation
  - Un ou de plusieurs « tracking » en simultané (mode « multi-marqueurs »)

- Bibliothèque ARToolkit portée sur Processing (depuis 2015) :
  - Nyar4psg par Ryo Iizuka / « nyatla » (« Nyatla Augmented Reality for ProceSsinG »)
    Version courante : NyAR4psg : 3.0.7 et NyARToolkit : 5.0.9
    http://nyatla.jp/nyartoolkit/wp/?page_id=198/
    https://github.com/nyatla/NyARToolkit-for-Processing

91

91

VI.4 – Flux de traitement dans ARToolKit

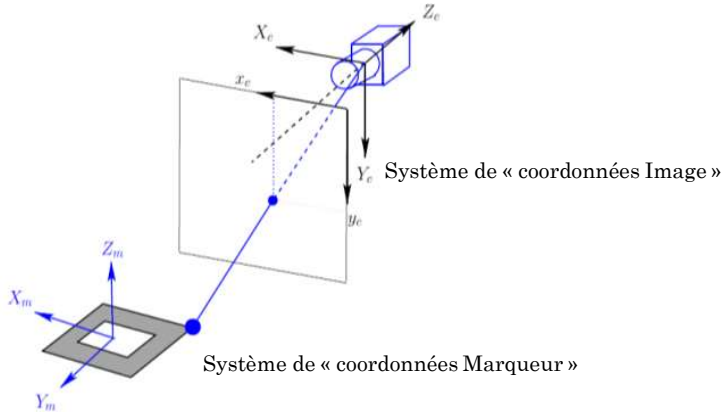


92

92

VI.5 – ARToolKit – Double système de coordonnées

- Principe des systèmes de coordonnées : Deux repères !



- Modification de l'image en utilisant le repère « marqueur » :
  - en 2D (X,Y) : `size(640,480); // (0,0) est le centre du repere du marqueur`
  - en 3D (X,Y,Z) : `size(640,480,P3D); // (0,0,0) est le centre du repere`

93

93

VI.6 – Principe en mode « multi-marqueurs »

- Déclaration et initialisation d'une scène de recherche

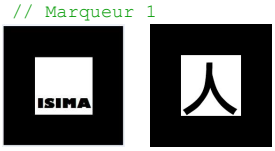
```
MultiMarker sceneMM;
sceneMM = new MultiMarker(this,widthCapture, heightCapture,
                          "camera_para.dat",
                          NyAR4PsgConfig.CONFIG_PSG);
```

  - avec dimensionnement de la zone de recherche
  - et avec déclaration des paramètres de la caméra

- Ajout d'un ou des fichiers de description des marqueurs, **séquentiellement** :

```
sceneMM.addARMarker("Marqueur_ISIMA.patt",80); // Marqueur 0
sceneMM.addARMarker("Marqueur_Kanji.patt",80); // Marqueur 1
```

  - La taille est fournie : 80 mm (taille classique)



- Recherche du ou des marqueurs dans la scène :

```
sceneMM.detect(webCam);  // Ou bien sur une image au lieu webCam !
```

- Analyse de présence « Test de réussite » :

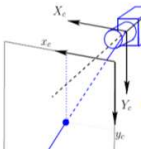
```
if (sceneMM.isExist(0)) { // Marqueur 0 trouve
```

94

94

VI.6 – Mise en œuvre d'une Réalité Enrichie

Solution 1 : modification de l'image en 2D, en coordonnées « image »
donc, **« Réalité Enrichie »**

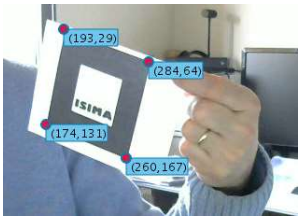


- Récupération des coordonnées du marqueur repéré :
  - Fonction `getMarkerVertex2D()`
  - Information retournée : Tableau des positions en coordonnées « images »

```
PVector[] pos2d; // Positions des coins en 2D

pos2d = sceneMM.getMarkerVertex2D(0);
// pos2d[0].x et pos2d[0].y : coordonnees images du coin 1
// pos2d[1].x et pos2d[1].y : coordonnees images du coin 2 ...
```

- Modifications graphiques de l'image :
  - directement par les fonctions primitives classiques de tracé 2D :
    `text`, `rect`, `point`, `ellipse`, ...
  - utilisation des déplacements avant tracé :
    `translate()`
  - A noter : sauvegarde et récupération du mode de tracé de processing (matrices de position et d'orientation du repère)
    `pushMatrix()` et `popMatrix()`

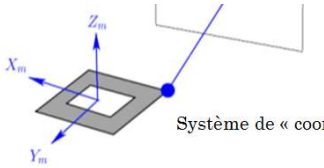


95

95

VI.6 – Mise en œuvre de la Réalité Augmentée

Solution 2 : modification de l'image en 3D (ou 2D), en coordonnées « Marqueur »



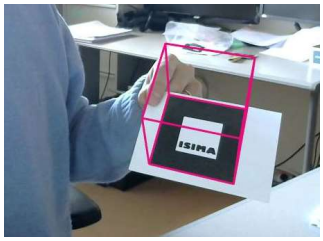
Système de « coordonnées Marqueur »

- Modifications graphiques de l'image (changement de repère) :

```
sceneMM.beginTransform(0); // Debut de transformation
// Ce beginTransform provoque le changement de repere dans Processing
// Toutes les commandes de traces se feront par
// rapport au nouveau système de coordonnees

  strokeWeight(2); // Trait epais
  stroke(#E802B7); // Couleur violacee
  noFill(); // Sans remplissage

  // Dessin en mode 3D : ici, un cube de 80mm de cote
  translate(0, 0, 40); // Placement du cube au dessus
                       // du marqueur, donc Z=40mm
  box(80, 80, 80);     // Cube de 8 cm de cote
sceneMM.endTransform();// retour en "coordonnees image"
```

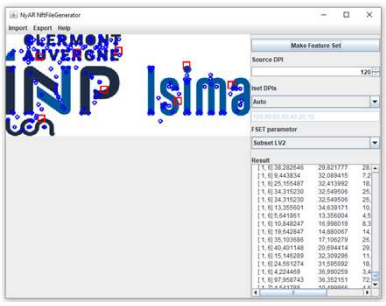


96

96

VI.7 – Variante sans marqueur, de type « NFT »

- Natural Feature Tracking : Image « quelconque », car mode 2D
  - Génération de l'ensemble des fichiers NFT (iset, fset et fset3) par « nftFilesGen »
    - menu fichier>Exemples>nyar4psg>nftFilesGen
    - « Import » importation du fichier image
    - « Make Feature Set » : génération des points d'intérêt
    - « Export > Save Feature Set Files » : création effective du marque NFT



- Utilisation du NFT :
  - Déclaration et initialisation d'une scène de recherche au format « multi-NFT»

```
MultiNft sceneNft;
sceneNft= new MultiNft(this
          widthCapture, heightCapture,
          "camera_para.dat",
          NyAR4PsgConfig.CONFIG_PSG);
```

  - Ajout d'un ou des fichiers de description des NFT, **séquentiellement** :

```
sceneNft.addNftTarget("Logo_Isima_INP",80); // Marqueur 0
```

  - Recherche du ou des NFT et analyse de présence identique aux marqueurs classiques :

```
sceneNft.detect(webCam);   // Detection sur le flux video de la webCam !
if (sceneNft.isExist(0)) { // Marqueur 0 trouve
```

97

97

VI.8 – Exemple 12 : Réalité Augmentée (avec marqueur)

```
/*********************************************************************************/
/*  MESNARD Emmanuel                                                  ISIMA    */
/*                                                                               */
/*          Exemple 12 : Realite Augmentee par ARtoolkit et Webcam               */
/*                                                                               */
/* Exemple_12_Realite_Augmentee.pde                    Processing 3.5.4      */
/*********************************************************************************/
import jp.nyatla.nyar4psg.*;   // ARToolKit (version 3.0.7)
...
// Declaration des variables globales
MultiMarker sceneMM; // Scene de recherche de "multi-marqueur"
boolean RE2D_RA3D;   // Choix du mode : Realite Enrichie 2D, ou Realite Augmentee 3D
...
void setup() {
...
  // Declaration de la scene de recherche avec parametres par defaut :
  // calibration de camera et systeme de coordonnees
  sceneMM = new MultiMarker(this, widthCapture, heightCapture,
    "camera_para.dat",
    NyAR4PsgConfig.CONFIG_PSG);
  // Declaration du marqueur a rechercher, avec sa dimension en mm
  sceneMM.addARMarker("Marqueur_Kanji.patt", 80); // Marqueur numero 0

  RE2D_RA3D = true; // Mode RE par defaut...
}
```

98

98

VI.8 – Exemple 12 : Réalité Augmentée

```
void draw() {
   if (webCam.available() == true) { // Verification de presence d'une nouvelle frame

    webCam.read(); // Lecture du flux sur la camera... lecture d'une frame

    sceneMM.detect(webCam);   // Recherche du marqueur dans la scene
    webCam.updatePixels();    // Mise a jour des pixels

    image(webCam, 0, 0);  // Affiche l'image prise par la webCam

    // Incrustation de l'image virtuelle si marqueur trouve
    if (sceneMM.isExist(0)) {
      // Le marqueur 0 est effectivement detecte dans le flux video
      if (RE2D_RA3D) {
        // Version 1 : trace d'informations basiques a l'ecran
        V1_RE_en_Coordonnees_Image();
      } else {
        // Version 2 : trace d'un cube 3D
        V2_RA_en_Coordonnees_Marqueur();
      }
    }
  }
}
```

99

99