



V – Expériences de Réalité Augmentée

75

75

V.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

76

76

V.1 – RA idéale

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



Problème plus difficile
que la réalité virtuelle !

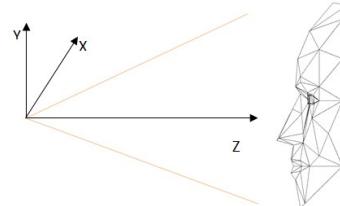
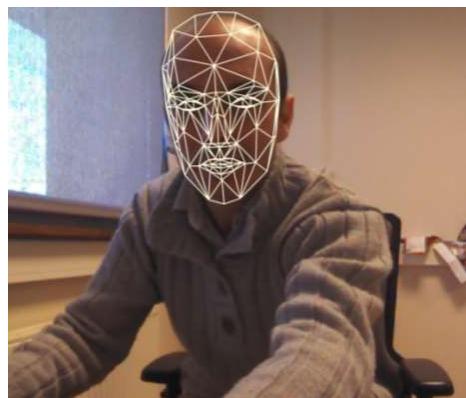
Source : Gilles Simon, *La réalité augmentée*, LORIA (France).

77

77

V.2 – RA sans marqueur

- Marker Less Tracking (MLT)
 - Analyse d'image sans marqueur : n'importe quel type de support peut être employé pour réaliser la RA.
 - Exemple : reconnaissance faciale



78

78

V.3 – RA avec marqueur

- Problème plus « simple » :
 - imprimer un motif particulier sur une feuille : le marqueur

Id Markers

Pattern Markers en pattern (texte) ou image

Natural Feature Tracking (NFT)

- Reconnaissance « aisée » des marqueurs

Taille, donc, distance caméra

Orientation spatiale 3D

79

79

V.3 – RA avec marqueur

- Placement de l'objet 3D, de manière cohérente
 - Respect de l'orientation
 - Respect de la taille (proportionnelle à la taille du marqueur décelé)
 - si possible, respect des conditions d'éclairage...
 - ... avec calcul des ombres projetées (plus difficile en Temps Réel)

80

80

V.4 – Réalité Augmentée avec ARToolKit

- Caractéristiques :

- Bibliothèque gratuite de fonctions dédiées à la RA (la plus utilisée !), développée par Hirokazu Kato en 1999



- Utilisation de marqueurs noirs et blancs ou d'images ou encore de NFT
 - Détection de la position spatiale
 - Détection de l'orientation
 - Détection d'un ou de plusieurs marqueurs simultanément (mode « multi-marqueurs »)

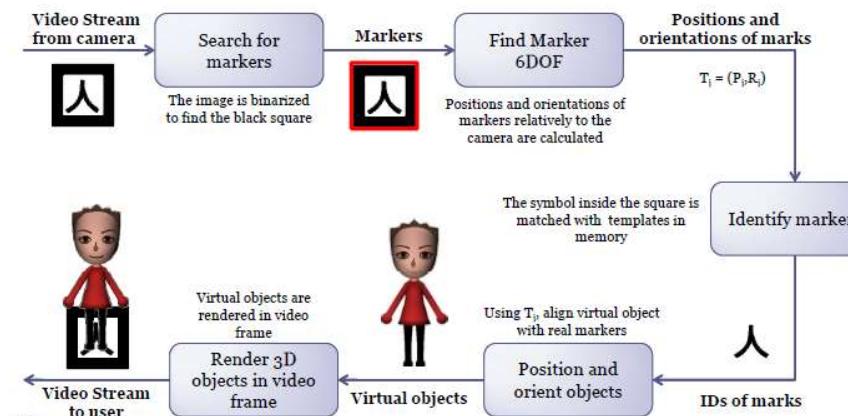
- Bibliothèque ARToolkit portée sur Processing (depuis 2015) :

- 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>

81

81

V.4 – Flux de traitement dans ARToolKit

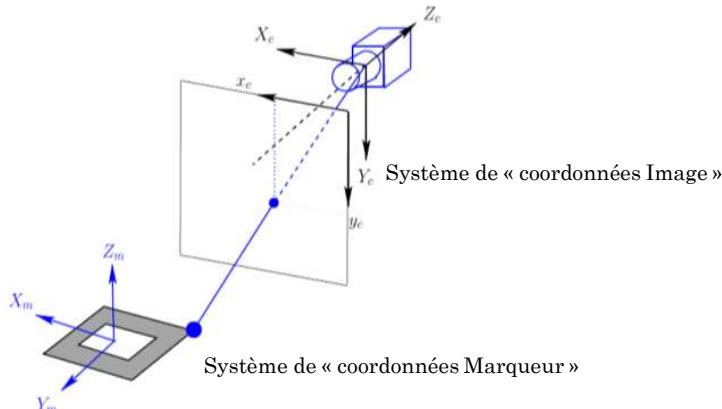


82

82

V.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`

83

83

V.6 – Mise en œuvre de la Réalité Augmentée (1/3)

- Déclaration et initialisation d'une scène de recherche de « multi-marqueurs »

```
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
// ou bien sceneMM.addARMarker("logo_ISIMA.png", 80);
sceneMM.addARMarker("Marqueur_Kanji.patt", 80); // Marqueur 1
```

La taille est fournie : 80 mm (taille classique)

- Le fichier « .patt » est un « simple » fichier texte donnant les 3 composantes R,G,B des 4 orientations possibles du marqueur



- 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
```

84

84

V.6 – Mise en œuvre de la Réalité Augmentée (2/3)

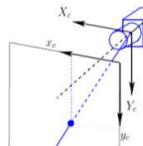
Solution 1 : modification de l'image en 2D, en coordonnées « image »
donc, « Réalité Enrichie »

- Récupération des coordonnées du marqueur repéré :

- Fonction `getMarkerVertex2D`
- Information renournée : Tableau des positions en coordonnées « images »

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

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



- Modifications graphiques de l'image :

- directement par les fonctions classiques de tracé 2D :

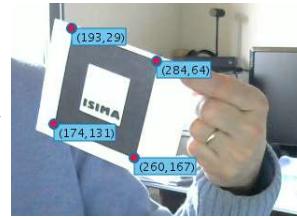
```
text, rect, point, ellipse, ...
```

- 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()
```

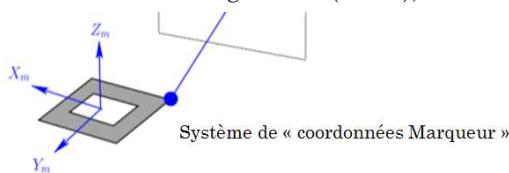


85

85

V.6 – Mise en œuvre de la Réalité Augmentée (2/2)

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



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

```
sceneMM.beginTransform(0); // Début de transformation
... // Changement de repère : centre du marqueur numero 0
... // puis divers tracés : line, ellipse, box, ..., translate, ...
sceneMM.endTransform(); // retour en "coordonnées image"
```



Exemple 8 : RA avec cube

86

86

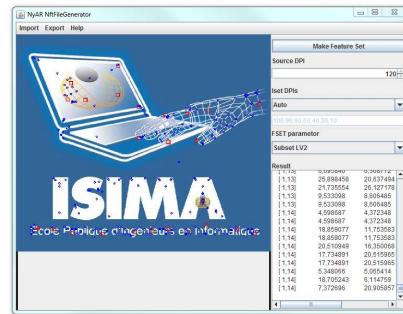
V.7 – Variante avec marqueur NFT

- Marqueur avec image « quelconque » : Natural Feature Tracking (NFT)
 - 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 marqueur NFT
- Utilisation du marqueur NFT :
 - Déclaration et initialisation d'une scène de recherche de « multi-marqueurs NFT »


```
MultiNft sceneNft;
sceneNft= new MultiNft(this
    widthCapture, heightCapture,
    "camera_para.dat",
    NyAR4PsgConfig.CONFIG_PSG);
```
 - Ajout d'un ou des fichiers de description des marqueurs, **séquentiellement** :


```
sceneNft.addNftTarget("Image_ISIMA",80); // Marqueur 0
```
 - Recherche du ou des marqueurs 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
```



87

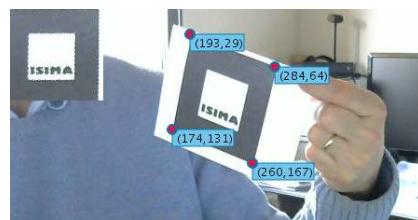
87

V.8 – Fonctionnalités étendues

- Récupération de l'image correspondant au marqueur : pickup


```
pickupImage, pickupRectImage, pickupMarkerImage
PIImage ImageVue; // Image au format Processing de 100x100
ImageVue = createImage(100,100,RGB);

ImageVue= sceneMM.pickupMarkerImage(0, -40, -40, // Coin 1 marqueur 0
40, -40, // Coin 2
40, 40, // Coin 3
-40, 40, // Coin 4
100,100); // Taille en pixels
ou :
(0, -40, -40, // Coin 1
80, // Largeur du marqueur
80, // Hauteur
100,100); // Taille en pixels
```
- Divers :
 - `getConfidence` : taux reconnaissance
 - `setThreshold`, `setLostDelay`, `setConfidenceThreshold` : paramétrages



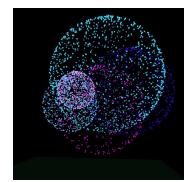
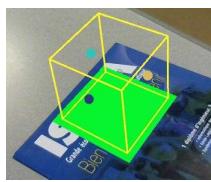
88

88

TD 6 Réalité Augmentée sur Marqueur Isima



- Travail demandé :
 - Créer un marqueur de type Image en NFT : logo Prep'Isima, plaquette de l'école...
 - Faire apparaître le repère si marqueur reconnu
 - Faire apparaître un objet « dynamique » :
 - une ou des balles rebondissantes,
 - ou encore un feu d'artifice !



89

89