

Kinect 1

Caméra RGB (capteurs CMOS), VGA (640x480), 30 Hz

Emetteur de lumière IR

Caméra IR (capteur CMOS), QVGA (320x240),
16 bits

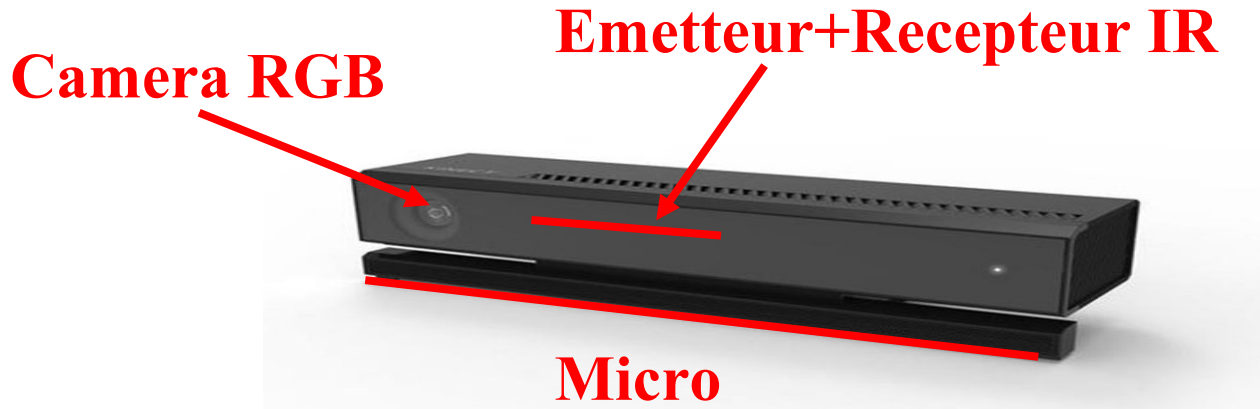


Bâti motorisé en Pan

Puissance de la lumière IR reçue dépend de la distance à l'objet le plus proche
(et d'autres facteurs)

Codage de la distance à l'objet le plus proche à partir d'une puce PS1080
Intégrée dans la Kinect

Kinect 2



Camera RGB: résolution $1920\text{pt} \times 1080\text{pt}$

Image profondeur: résolution $512\text{pt} \times 424\text{pt}$

Mesure par temps de vol des signaux IR

Fréquence: 30 Hz

Connectique: USB 3

Windows ≥ 8.1

Kinect 1

Capteurs

- profondeur (IR)
- image RGB VGA
- microphone

Effecteurs

- moteur (rotation 1D)

IR (bas niveau)

- grille 2D profondeurs
- image 2D RGB
- signal sonore

IR (haut niveau)

- suivi de mouvement (24 points et positions angulaires) sur le corps Humain dont 20 mesurés et 4 interpolés

Sensor	Joint	Sensor	Joint
0	Head	12	Right Elbow
1	Neck	13	Right Wrist
2	Torso	14	Right Hand
3	Waist	15	Right Fingertip
4	Left Collar	16	Left Hip
5	Left Shoulder	17	Left Knee
6	Left Elbow	18	Left Ankle
7	Left Wrist	19	Left Foot
8	Left Hand	20	Right Hip
9	Left Fingertip	21	Right Knee
10	Right Collar	22	Right Ankle
11	Right Shoulder	23	Right Foot

Kinect 2

Capteurs

- profondeur (IR)
- image RGB Full HD
- microphone

IR (bas niveau)

- grille 2D profondeurs
- image 2D RGB
- signal sonore

IR (haut niveau)

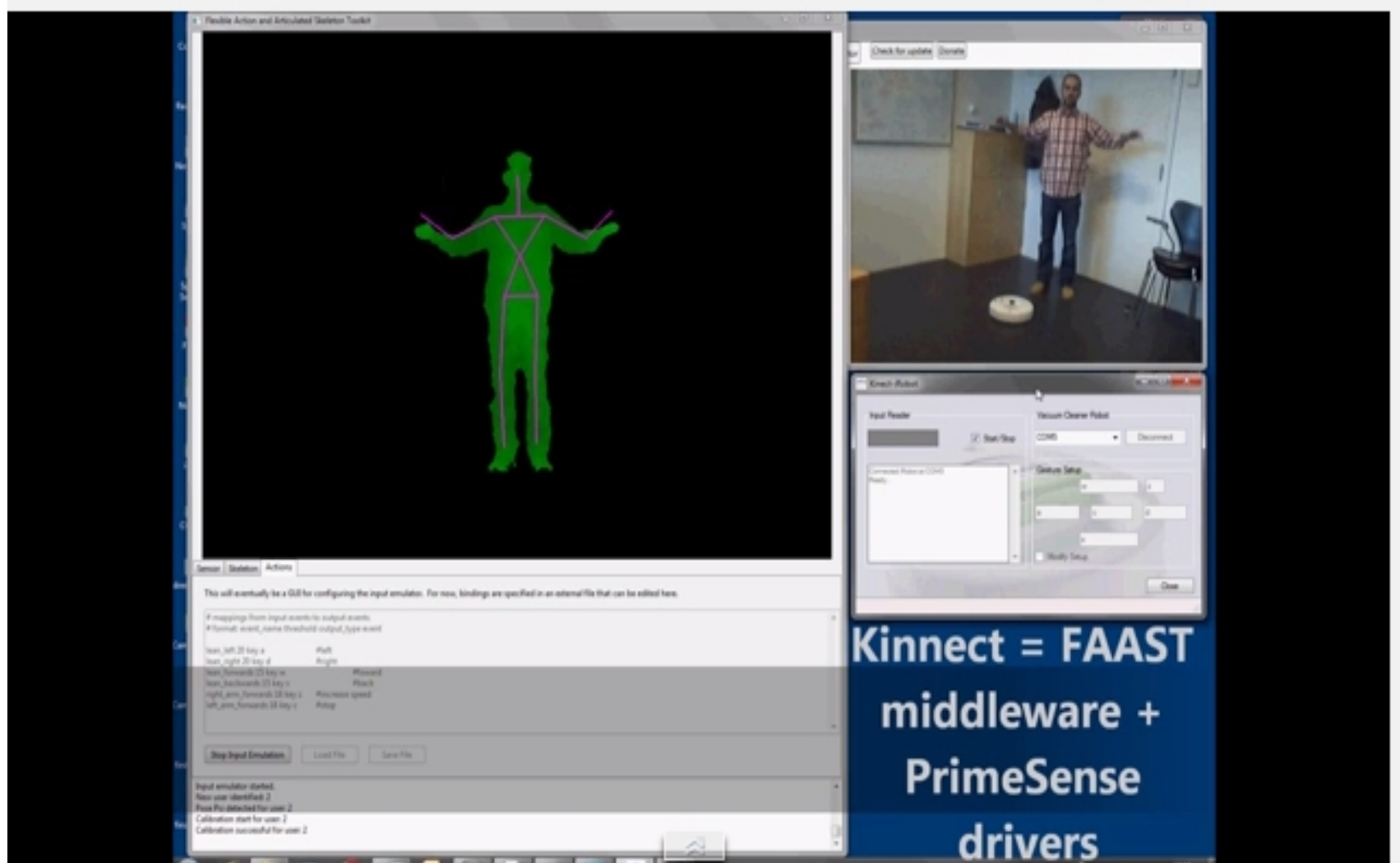
- suivi de mouvement (26 points et positions angulaires) sur le corps Humain dont 20 mesurés et 4 interpolés

Sensor	Joint	Sensor	Joint
0	Hip_Center	14	Ankle_Left
1	Spine	15	Foot_Left
2	Neck	16	Hip_Right
3	Head	17	Knee_Right
4	Shoulder_Left	18	Ankle_Right
5	Elbow_Left	19	Foot_Right
6	Wrist_Left	20	Spine_Shoulder
7	Hand_Left	21	Hand_Tip_Left
8	Shoulder_Right	22	Thumb_Left
9	Elbow_Right	23	Hand_Tip_Right
10	Wrist_Right	24	Thumb_Right
11	Hand_Right		
12	Hip_Left		
13	Knee_Left		

Accès aux données de la Kinect dans Unity

- Utilisation du logiciel FAAST comme producteur d'événements clavier/souris captables par Unity
 - Création d'une bibliothèque de gestes dans FAAST
 - Un geste <-> événement clavier/souris particulier
 - Récupération de l'événement dans Unity
 - `Input.GetKeyDown()`
 - `Input.GetMouseDown()`

FAAST: Squelettisation d'une à quatre personnes



Accès aux données de la Kinect 1 dans Unity

- Interface Microsoft Kinect SDK 1.7 avec Unity
 - Package Kinect1.7UnityPackage
 - Récupération du flux RGB
 - Récupération du flux DEPTH
 - Récupération de la squelettisation
 - En position
 - » Classe C# KinectPointController
 - » 20 GameObjects en entrée = 20 parties du corps
 - » Relation avec un SkeletonWrapper (interface Kinect1)
 - En rotation
 - » Classe C# KinectModeControllerV2
 - » 25 GameObjects en entrée = 25 parties du corps
 - » Relation avec une SkeletonWrapper (interface Kinect1)

Accès aux données de la Kinect 2 dans Unity

- Interface Microsoft Kinect SDK 2 avec Unity
 - Package KinectV2withMsSDK
 - Récupération du flux RGB
 - Récupération du flux DEPTH
 - Récupération de la squelettisation
 - En position
 - » Classe C# CubeManController
 - » 25 GameObjects en entrée = 25 parties du corps
 - » Relation avec un SkeletonWrapper (interface Kinect)
 - En rotation
 - » Classe C# KinectModeControllerV2
 - » 25 GameObjects en entrée = 25 parties du corps
 - » Relation avec une SkeletonWrapper (interface Kinect)

ANNEXE

Selection

- **3 états**

- Libre
- Sélectionnable
- Sélectionné

- **Mise en évidence graphique du changement d'état:**

- *GameObject.renderer.material.color = new Color(R, G, B, A).*

Sélection par Raycast

```
Ray ray = new Ray(transform.position,  
transform.forward);  
RaycastHit hit;
```

- Test de collision du rayon avec un *GameObject* muni d'un collider
if (Physics.Raycast(ray, out hit, range))
{
 - On peut ici voir la nature de l'objet rencontré
cible = hit.transform.gameObject;
... et traiter son état en conséquence.

Traçage des données

- Sur la Console Unity
 - `Debug.Log(data)`
- Dans un fichier
 - *using System.IO, using System.Text*
 - *System.IO.File.WriteAllText(nom_fich, texte) :*
Ecriture/Ecrasement d'un fichier existant.
 - *System.IO.File.AppendAllText(nom_fich, texte) :*
Ecriture en mode Append.
- *Sur la fenêtre graphique*
 - *Dans la fonction OnGUI() du script C#*
- `GUI.Label (Rect (coord), variable à afficher)`

Traçage des données

- Compter le temps ...
 - *using System.Diagnostics* à inclure
 - *public Stopwatch stopWatch : timer*
 - *stopWatch = new Stopwatch();*
 - *stopWatch.Start(); Démarrer le timer*
 - *TimeSpan ts = stopWatch.Elapsed;*

Temps écoulé depuis de démarrage

- *ts.TotalSeconds temps écoulé en secondes*