ARC Reference Modules for Real Character Representation

SIGGRAPH Web3D Meeting
Aug. 8 2012

Kwan-Hee Yoo
Chungbuk National University

- Set of principles
- Terms and their precise definitions
- ARC reference module for real character representation
- Use cases

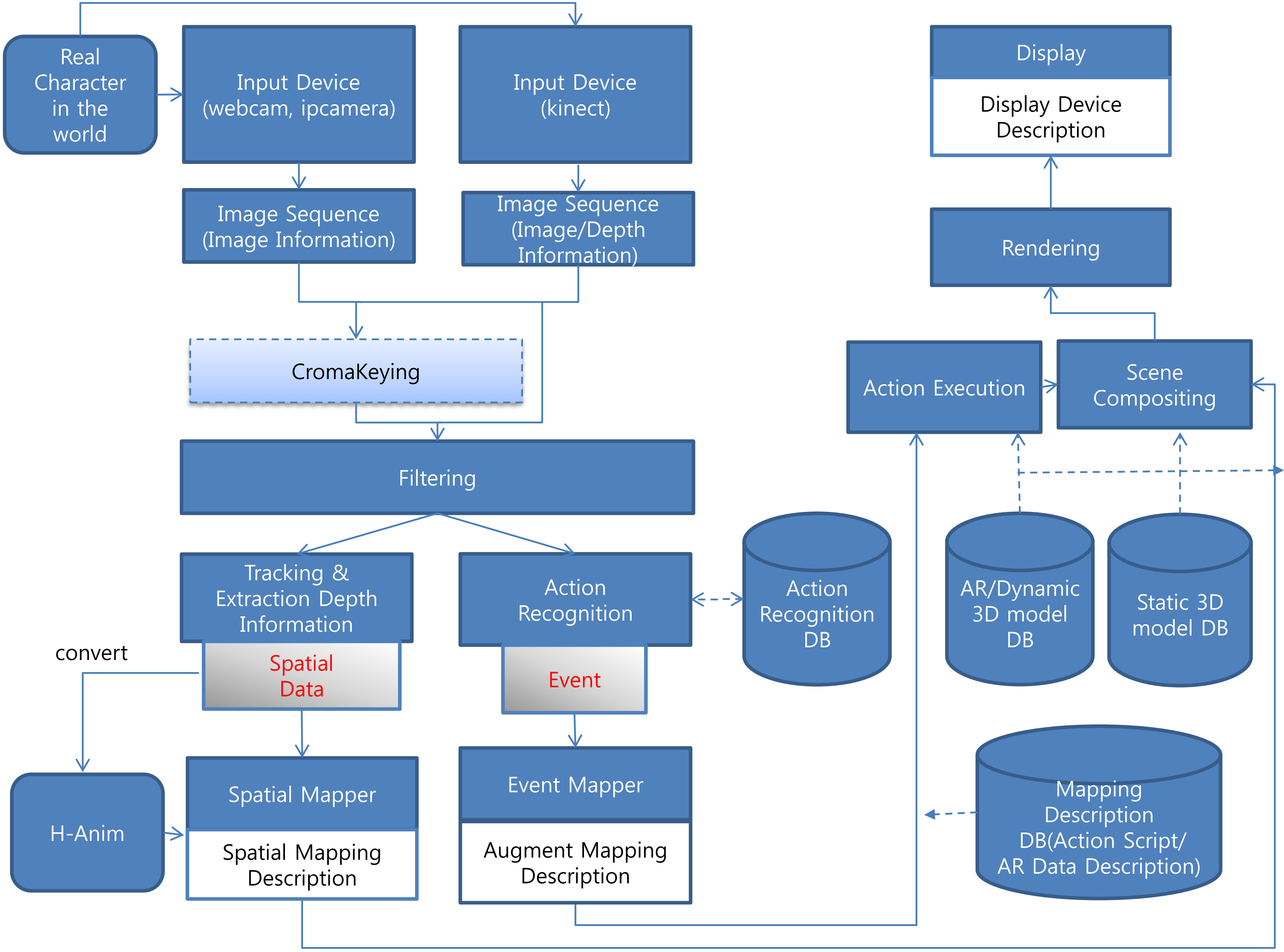
Set of principles

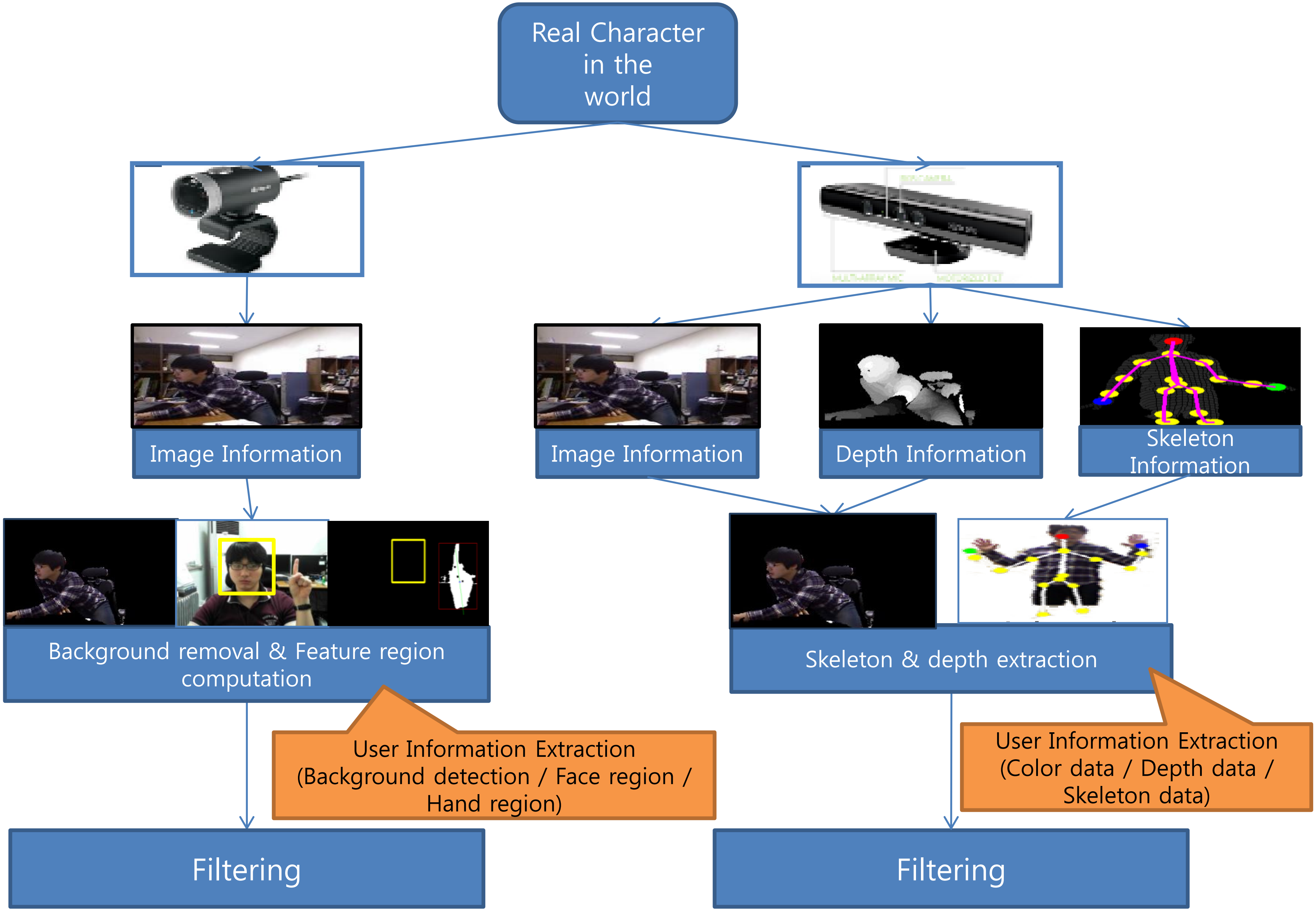
- This module is made to represent real character into ARC space based on ARC Reference Model
- Representation items of real characters
    - Spatial Mapper: Embedding real characters in real space into those in ARC space. Real characters obtained from (for an example) depth camera can be animated after converting their action data to H-Anim
    - Event Mapper: connect recognized events acted by real characters to reactions due to their events (interaction between real character and ARC environment)

Terms and their precise definitions

- Real characters
- Real space
- ARC space
- BBRS: Bounding box in real space
- BBARCS: Bounding box in ARC space
- Spatial mapper
- Spatial mapping description
- H-Anim
- (Virtual & Real) Character animation

- Action – gesture, voice, face
- (Action) recognition
- Event
- Event Mapper
- Augmented mapping description


Real Character in the world
Input Device (webcam, ipcamera)
Input Device (kinect)
Display
Display Device Description
Image Sequence (Image Information)
Image Sequence (Image/Depth Information)
Rendering
CromaKeying
Action Execution
Scene Compositing
Filtering
Tracking & Extraction Depth Information
Action Recognition
Action Recognition DB
AR/Dynamic 3D model DB
Static 3D model DB
convert
Spatial Data
Event
H-Anim
Spatial Mapper
Spatial Mapping Description
Event Mapper
Augment Mapping Description
Mapping Description DB(Action Script/ AR Data Description)



Real Character in the world

Image Information

Image Information

Depth Information

Skeleton Information

Background removal & Feature region computation

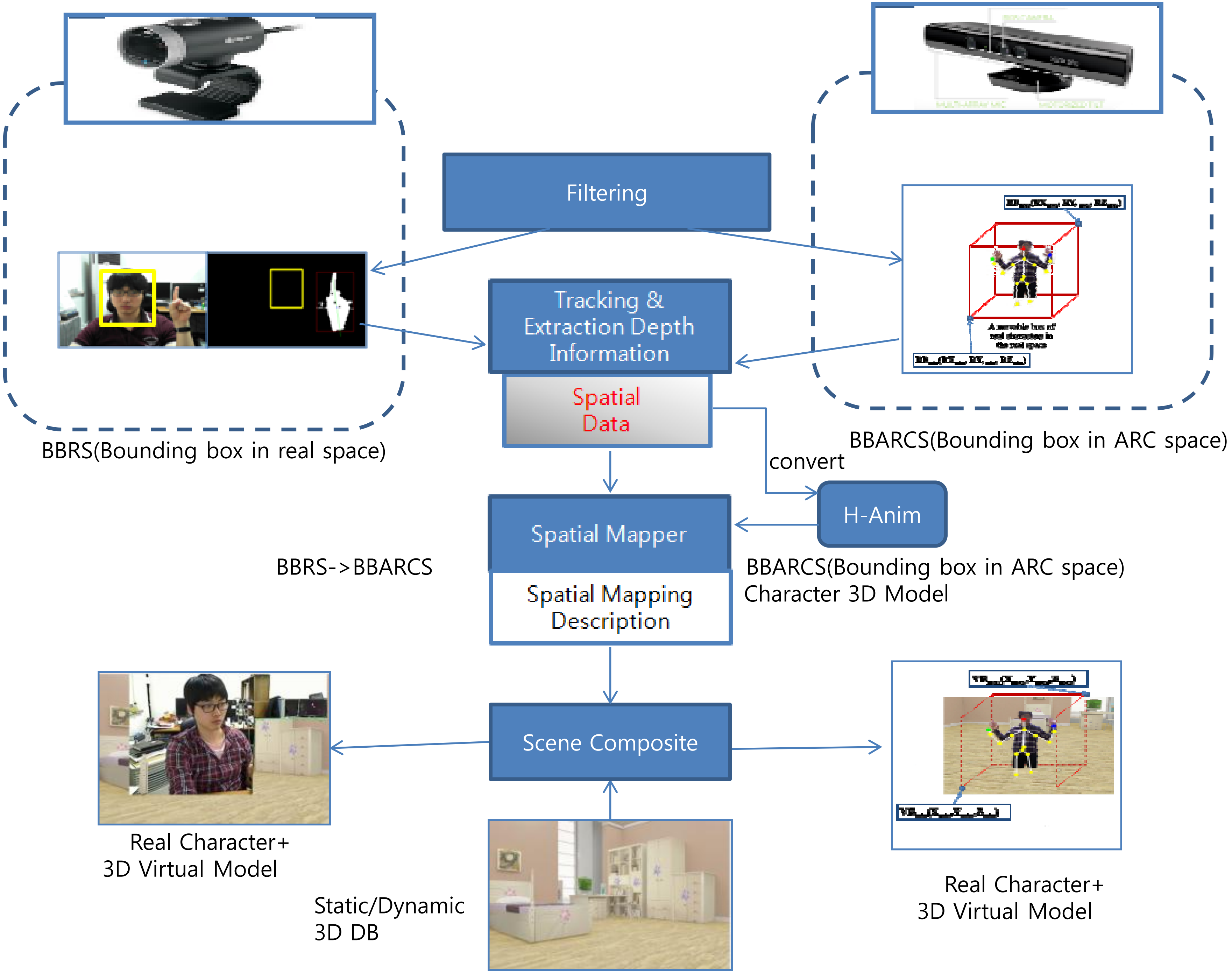
User Information Extraction (Background detection / Face region / Hand region)

Skeleton & depth extraction

User Information Extraction (Color data / Depth data / Skeleton data)

Filtering

Filtering


Filtering
Tracking & Extraction Depth Information
Spatial Data
convert
H-Anim
BBRS(Bounding box in real space)
BBARCS(Bounding box in ARC space)
Spatial Mapper
Spatial Mapping Description
BBRS->BBARCS
BBARCS(Bounding box in ARC space)
Character 3D Model
Scene Composite
Real Character+
3D Virtual Model
Static/Dynamic
3D DB
Real Character+
3D Virtual Model

Nodes

Nodes for real character

```
DeviceNode
{
  SFString[in]      id
  SFString[in]      type
  SFFloat[in]       fov
  SFInt[in]         framerate
  SFImage[out]      image
  MFString[out]     jointType
  MFVector[out]     value
}
```

Nodes

Nodes for real character

```
BBRSNode
{
  SFString[in]      id
  SFString[in]      description
  SFVector[out]   startpoint
  SFVector[out]   endpoint
  DeviceNode[in]            device
}

BBARCSNode
{
  SFString          id
  SFString          description
  SFVector[out]   startpoint
  SFVector[out]   endpoint
   // 가상공간을 명시하는게 좋을 듯 함
}
```

Use Cases of Spatial Mapper

General Camera Sensor for real character

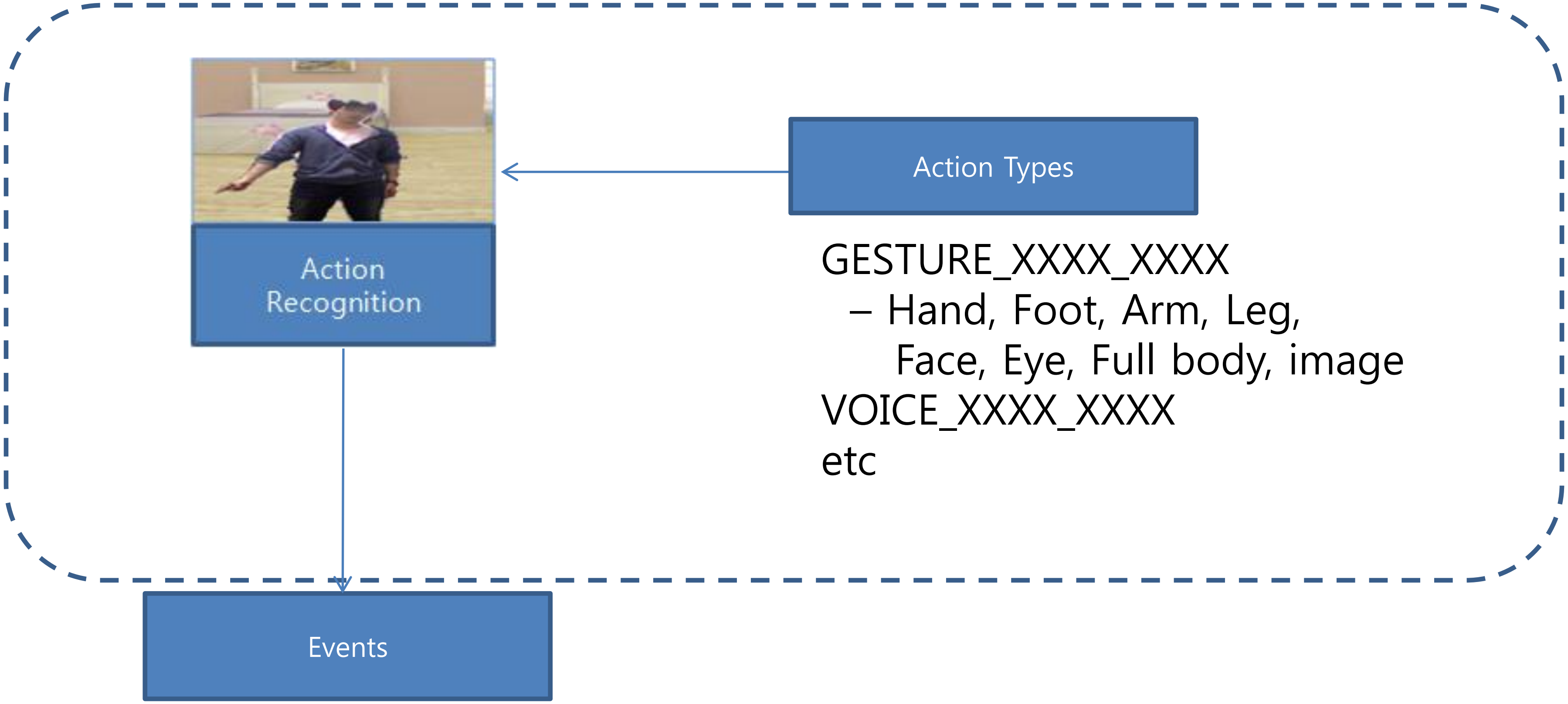
<device id = "cam0" type = "camera" fov="50" framerate="30"> </device>

Depth Camera Sensor for real character

<device id = “depthcam1" type = “depthcamera" fov="50" framerate="30“ joint_type = “left_hand” value = “0 0 0”> </device>

Get two points for a bounding box in real space
<bbrs id = “bbrs1" description = “movable space of real characters in real space" startpoint=“0 0 0" endpoint=“640 480 100” device=“cam0”> </bbrc>

Get two points for a bounding box in ARC space
<bbarcs id = “bbarcs1" description = “movable space of real characters in ARC space" startpoint=“30 30 20" endpoint=“60 40 10”> </bbarcs>



Action Recognition

Action Types

GESTURE_XXXX_XXXX
- Hand, Foot, Arm, Leg, Face, Eye, Full body, image

VOICE_XXXX_XXXX

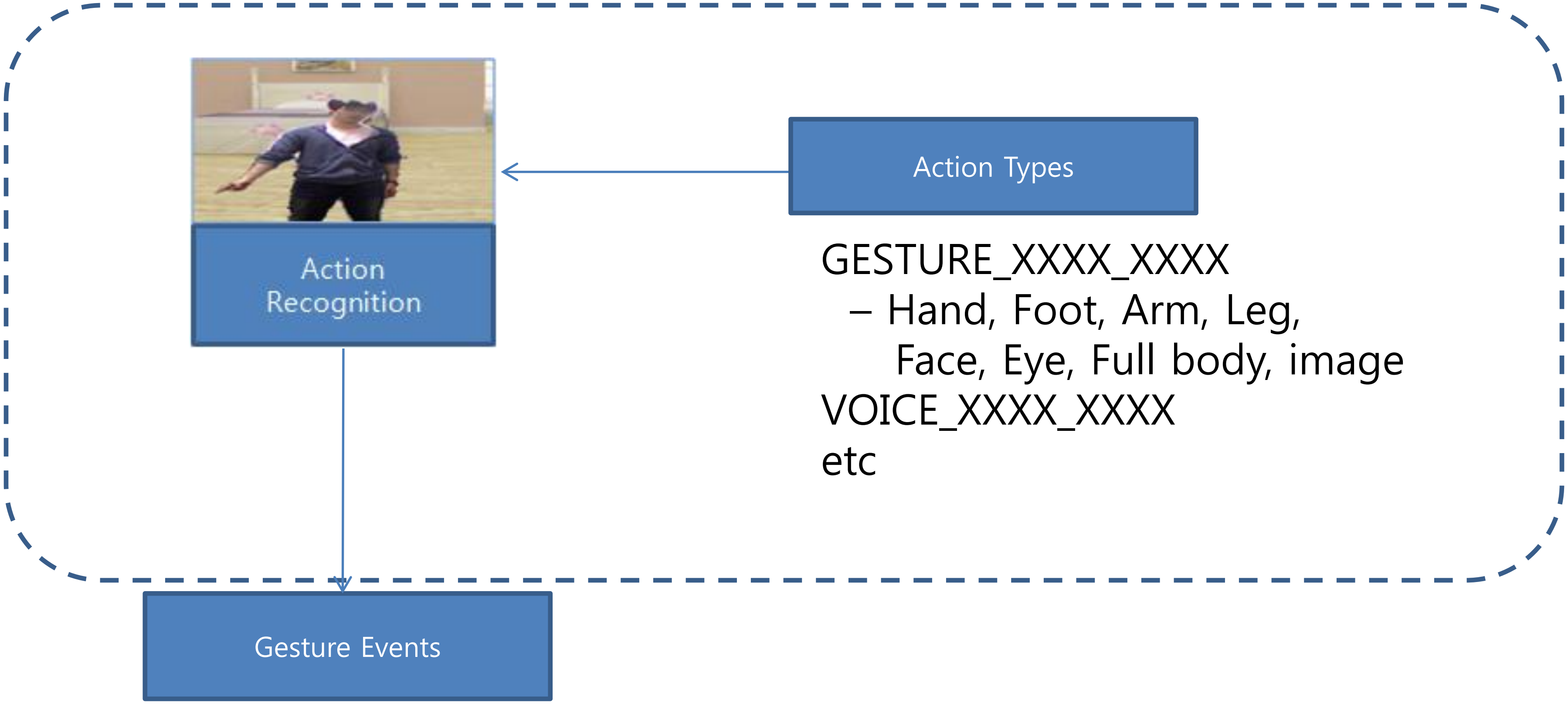
etc

Events

● point

image

↗ vector



Action Recognition

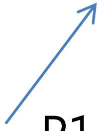
Action Types

GESTURE_XXXX_XXXX
– Hand, Foot, Arm, Leg,
Face, Eye, Full body, image
VOICE_XXXX_XXXX
etc

Gesture Events

● point

P= (x,y,z)



P2

vector

P1

P1=(x1,y1,z1)

P2=(x2,y2,z2)

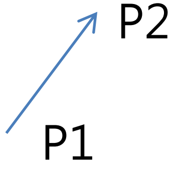
image

skeleton

Gesture Events

Posture ● point

P= (x,y,z)



vector

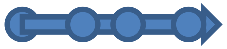
P1=(x1,y1,z1)

P2=(x2,y2,z2)

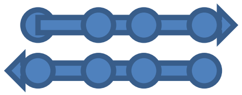
image

skeleton

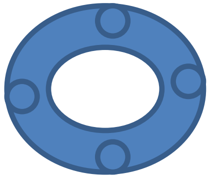
Gesture



X_POS, X_NEG
Y_POS, Y_NEG
Z_POS, Z_NEG



X_WAVE
Y_WAVE
Z_WAVE



CW(CLOCKWISE)
CCW(COUNTER CLOCKWISE)

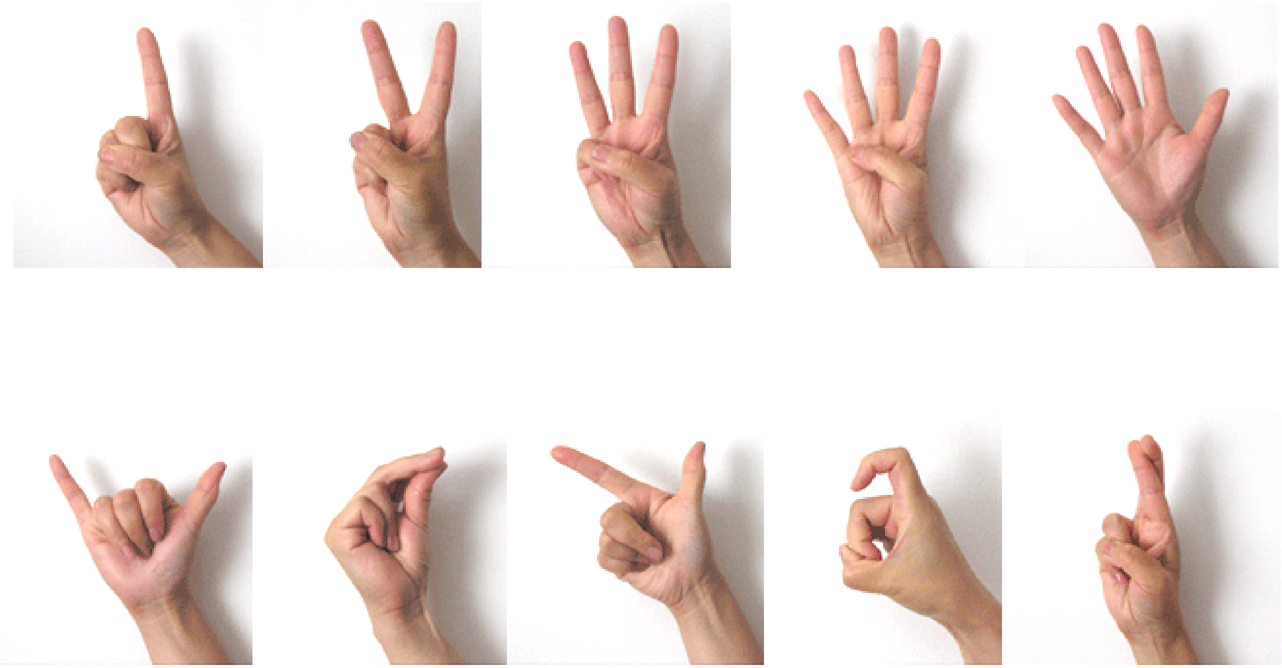
Gesture Events Types

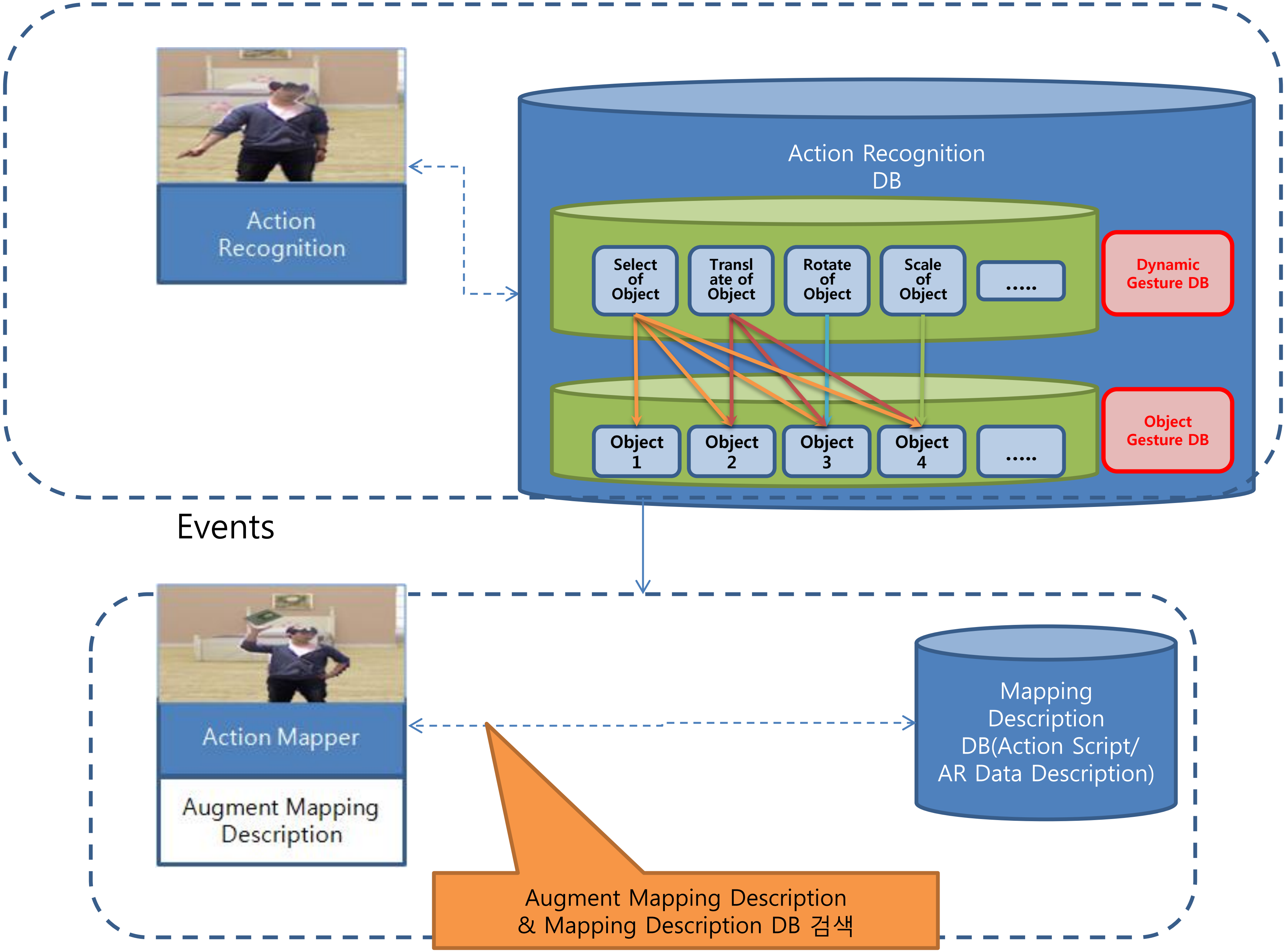
Action	Objects	Move	Definition of action types
Gesture	Left hand(LH)	The x positive direction	GESTURE_LH_X_POS
	Right hand(RH)	The x negative direction	GESTURE_RH_X_NEG
	Left foot(LF)	The y positive direction	GESTURE_LF_Y_POS
	Right foot(RF)	The y positive direction	GESTURE_RF_Y_POS
	Left Lower Arm(LLA)		
	Left Arm(LA)		
	Right Lower Arm(RLA)		
	Face(F)		
	Eye(E)		
	Full body(FB)		

Use case for controlling a camera movement by using Gesture Events

	LEFT_HAND	RIGHT_HAND	Functions
Camera control using hand gestures	← GESTURE_LH_X_NEG	-	**Left (Rotation)** Movement of a camera in x-positive direction
	-	→ GESTURE_RH_X_POS	**Right (Rotation)** Movement of a camera in x-negative direction
	↑ GESTURE_LH_Y_POS	GESTURE_RH_Y_POS ↑	**Up (Rotation)** Movement of a camera in y-positive direction
	↓ GESTURE_LH_Y_NEG	- ↓ GESTURE_RH_Y_NEG	**Down (Rotation)** Movement of a camera in y-negative direction
	← GESTURE_LH_X_NEG	→ GESTURE_RH_X_POS	**Zoom In (Scaling)** Decreasing distance between a model center and a camera center
	→ GESTURE_LH_X_POS	← GESTURE_RH_X_NEG	**Zoom Out (Scaling)** Increasing distance between a model center and a camera center

Image Events





Action Recognition

Action Recognition DB

Select of Object

Transl ate of Object

Rotate of Object

Scale of Object

.....

Object 1

Object 2

Object 3

Object 4

.....

Dynamic Gesture DB

Object Gesture DB

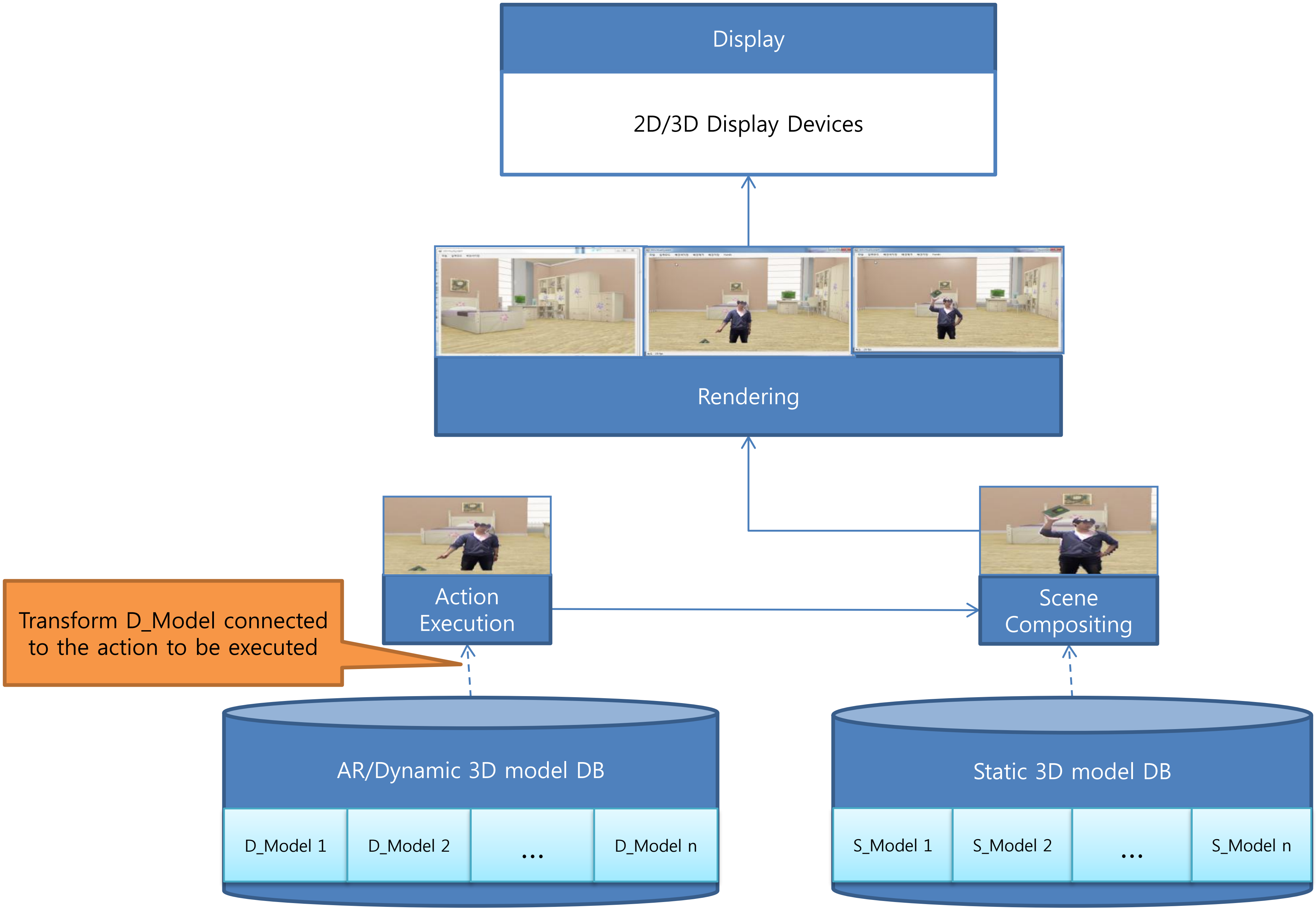
Events

Action Mapper

Augment Mapping Description

Mapping Description DB(Action Script/ AR Data Description)

Augment Mapping Description & Mapping Description DB 검색



Display

2D/3D Display Devices

Rendering

Action Execution

Scene Compositing

Transform D_Model connected to the action to be executed

AR/Dynamic 3D model DB

D_Model 1 | D_Model 2 | … | D_Model n

Static 3D model DB

S_Model 1 | S_Model 2 | … | S_Model n

Thank you

Kwan-Hee Yoo
khyoo@chungbuk.ac.kr