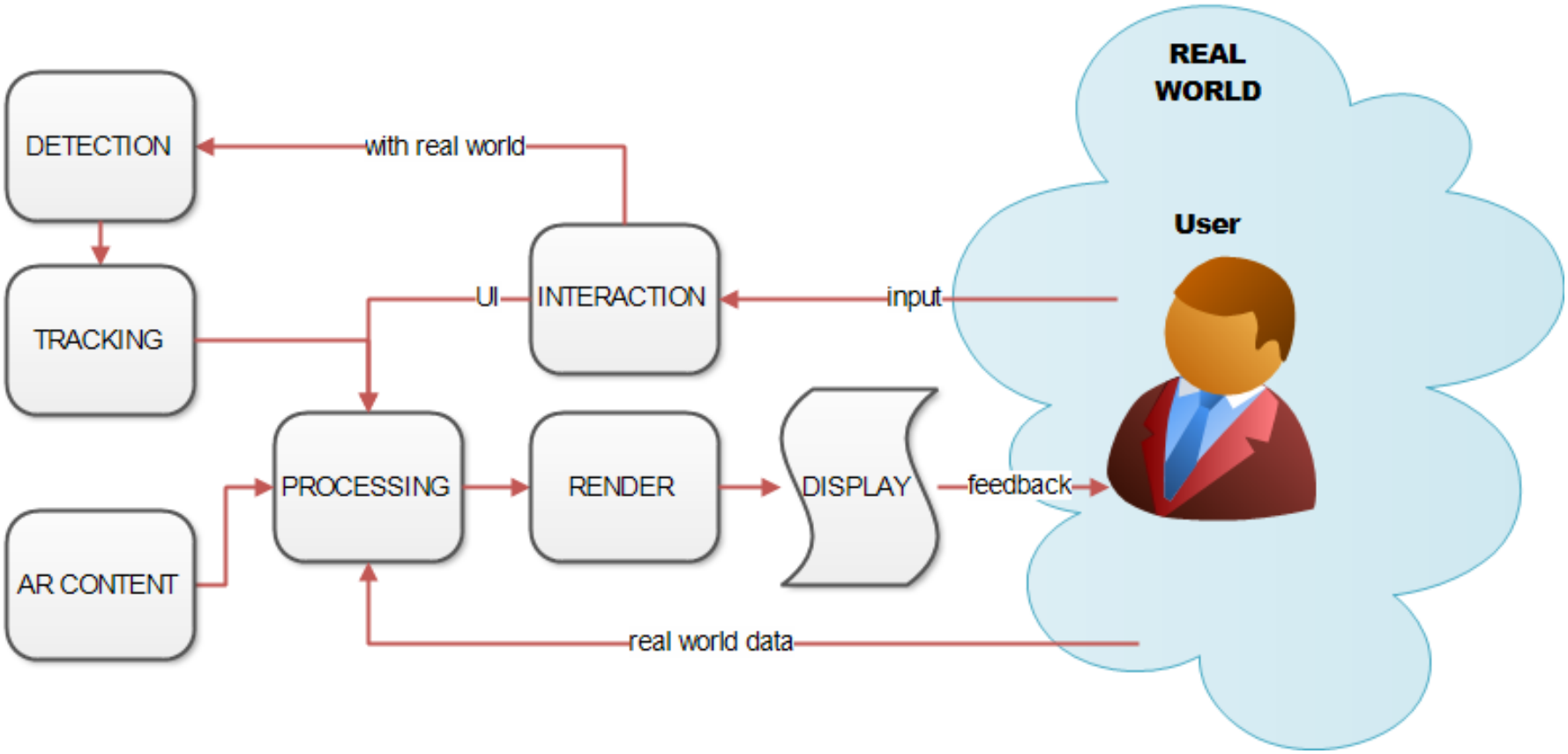
LABORATORIO DI REALTÀ AUMENTATA

Claudio Piciarelli

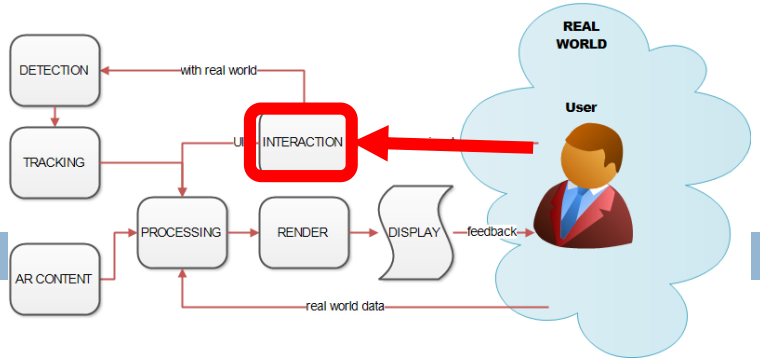
Università degli Studi di Udine
Corso di Laurea in Scienze e Tecnologie Multimediali

Architecture of an AR system

Architecture of an AR system



Interaction



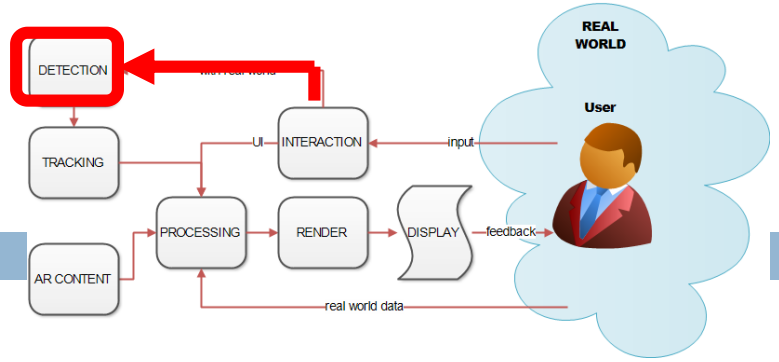
- The user interacts with the system in some way
  (fundamental property of AR!)
- Either directly, with a User Interface (UI)…
  - GUI: buttons, etc. (G stands for Graphical)
  - Vocal commands
  - Gestures
- …or indirectly, acting on the real world
  - Motion (GPS, accelerometers…)
  - Orientation (compass, gyroscope…)
  - Marker detection

Example: navigators

Direct interaction with GUI

Indirect interaction by changing position

Interaction



- Interaction must be **detected**…

- This is particularly challenging for indirect interactions, because:
  - The user does not interact with the AR system, but with the real world
  - Need for specific sensors to detect this form of interaction

Detection of interactions

- In many cases, interactions can be detected by analyzing specific sensors outputs:
  - Movement – GPS, accelerometer
  - Orientation – compass, gyroscope

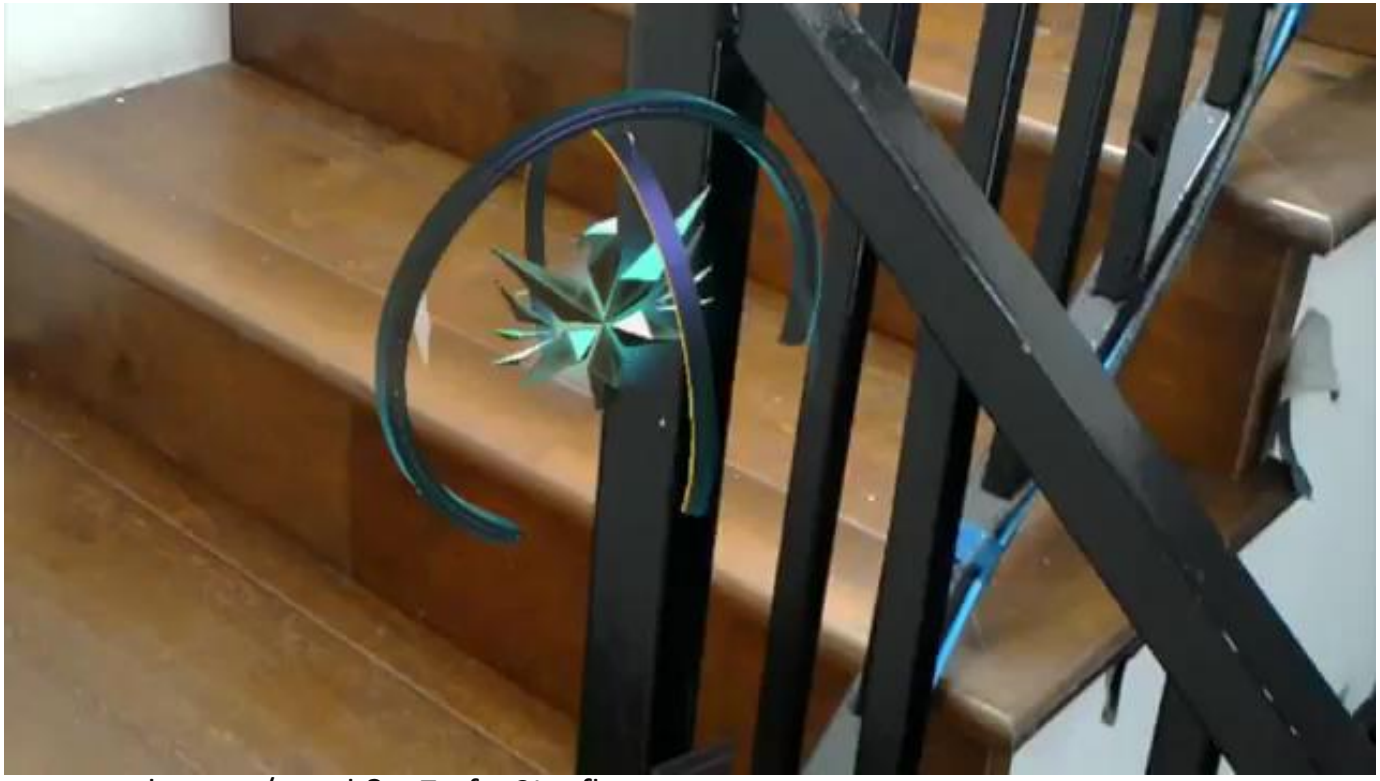
Problems:
- Low precision
- Occlusions, fixation time for GPS

- Sometimes detecting interaction can be a complex process:
  - Audio
  - Video

Audio interaction examples

- Voice commands
- Automatic translation

  (are these direct or indirect interactions?)



https://www.youtube.com/watch?v=7mfge3Iq_fk

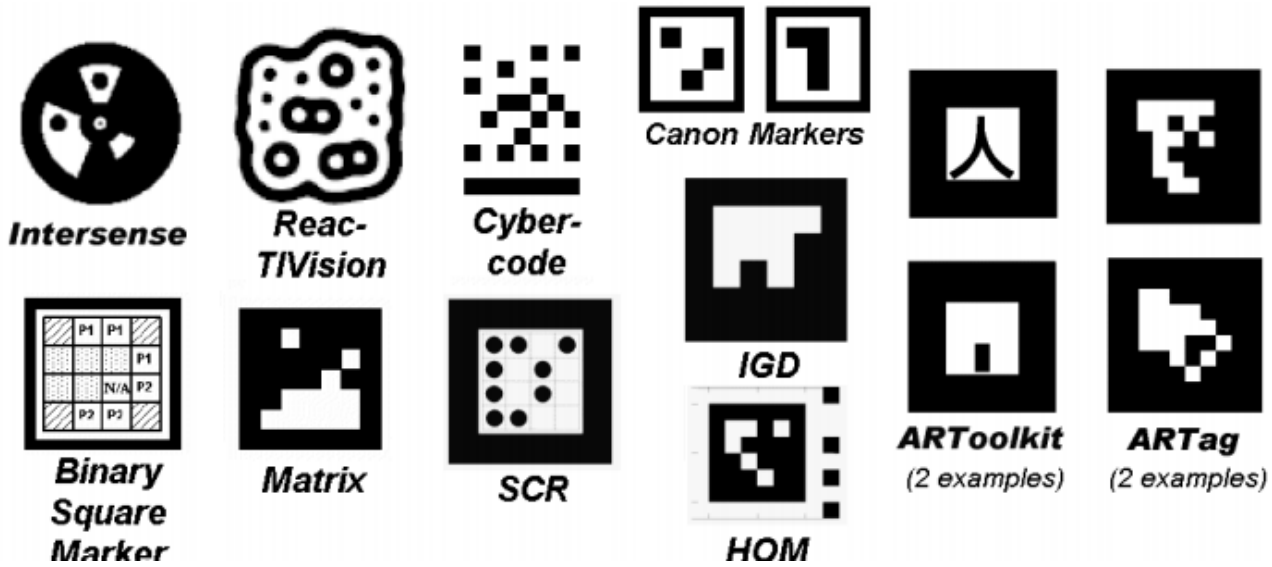
Video interaction

- Gestures
- Detection of specific patterns (markers, features)
  - Fiducial markers detection
  - Natural features detection

https://www.youtube.com/watch?v=kwn9Lh0E_vU

Fiducial Markers

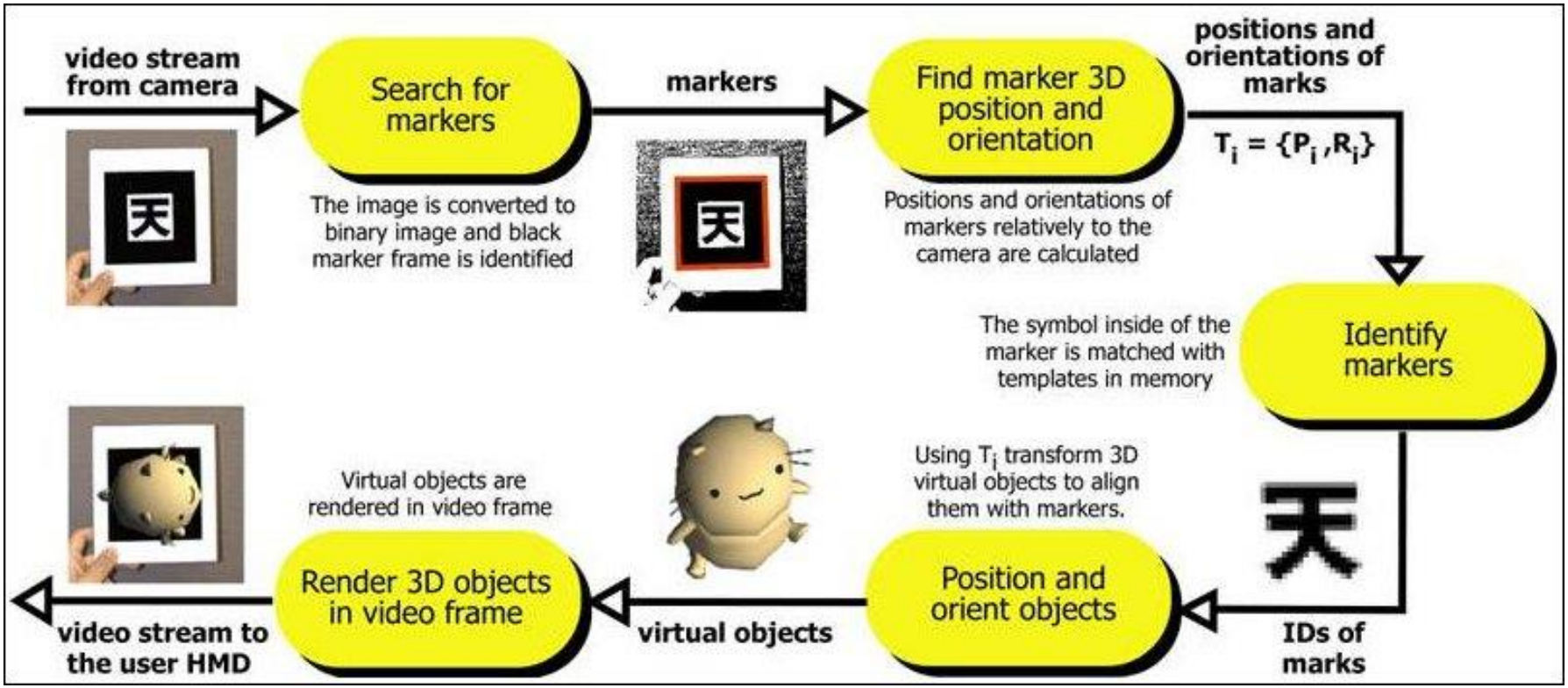
- Fiducial markers are specific patterns artificially added to the scene as a “placeholder” for augmented content
- Typically studied to be easily detectable: high contrast, sharp shapes



Fiducial markers

Interaction is indirectly achieved by moving the marker or the camera

Marker detection in jsARToolKit

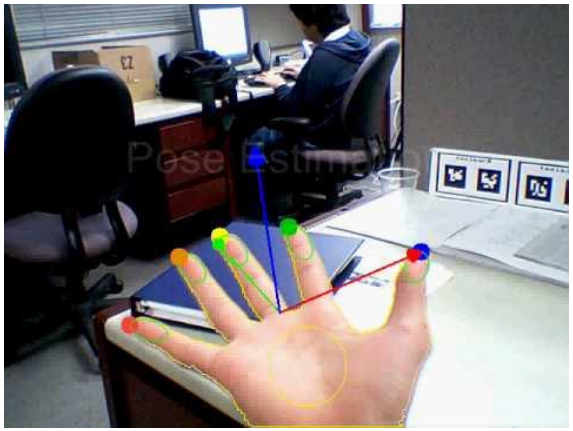


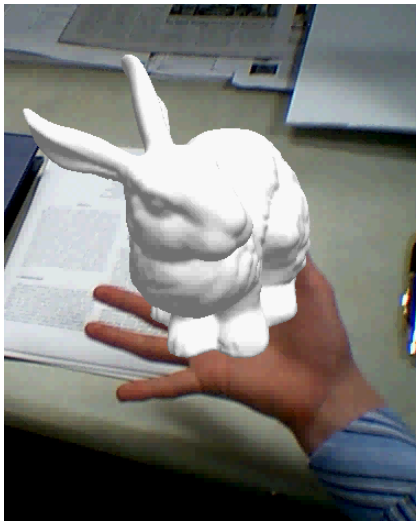
Source: jsartoolkit documentation

Natural features

- Natural features (“markerless”) systems recognize natural images and use them as fiducial markers
- It is an extension of the fiducial marker approach, in which any object can become a marker

Markerless examples

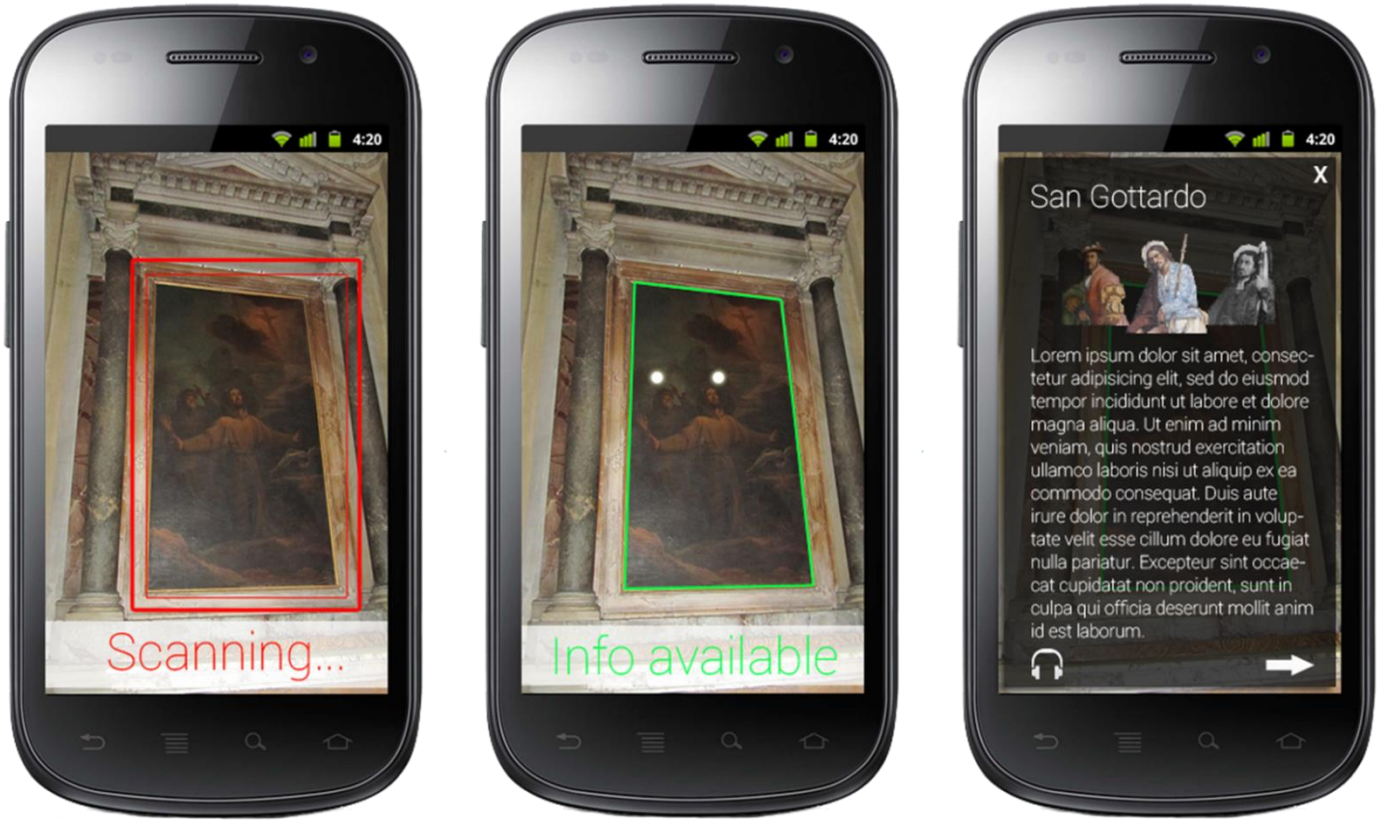




Markerless examples

AR for tourism:
Could be achieved by position/orientation sensors (GPS, compass), by natural features visual detection, or a combination of both

Markerless examples



Natural feature detection

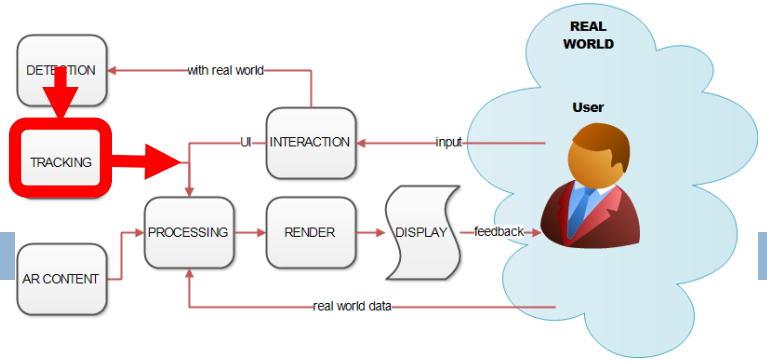
- State-of-the-art image processing techniques
- Much harder than standard marker detection
- 3D objects vs 2D markers
- Changes in appearance (pose)
- More subject to detection errors (occlusions, bad lighting conditions, similar objects…)

Environment detection

- Rather than focusing on a single target, the entire environment structure is computed (spatial mapping)
- E.g. useful for placing objects at specific points

https://www.youtube.com/watch?v=zff2aQ1RaVo https://www.youtube.com/watch?v=BC1k_18JUDk

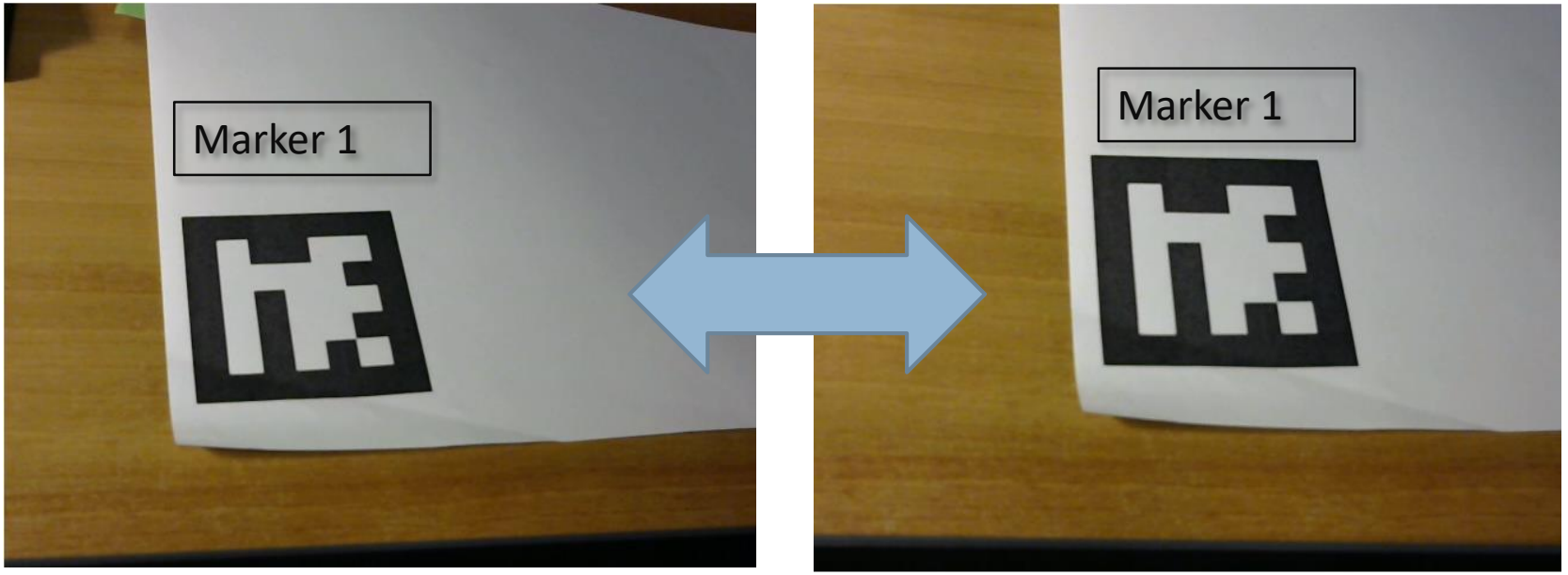
Tracking



- Detection: recognize an input to be processed
- Tracking: keep track of this input to understand the relationship between user and real world

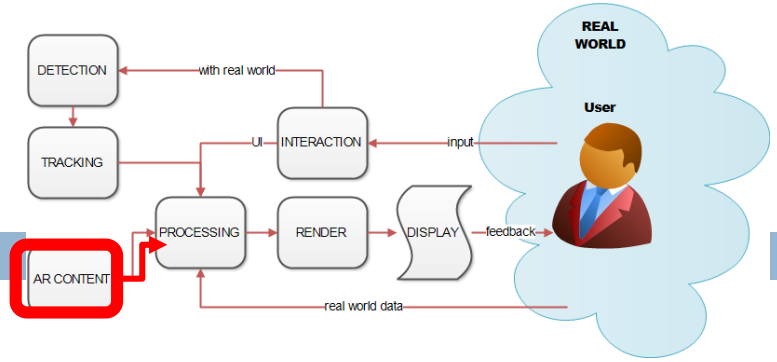
- Tracking adds temporal coherence to detection results

Tracking: example



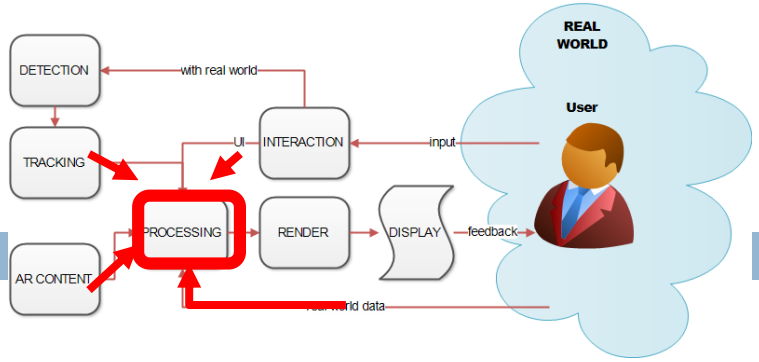
Tracking tells us that these two *detected* markers are actually the same marker, detected at two different positions (or from two different points of view) at different times

AR content



- This is the set of all possible contents that can be added to real world to augment it
- E.g. 3D models, text, graphics, movies, POIs…
- Depending on the case, it can be embedded in the application or be stored in an external source of data (content provider)

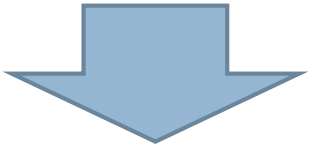
Processing



- The processing module is the core of the AR system
- It takes data from real world (e.g. real-time video) and augments it with AR content, on the basis of user interaction, either direct (with the system) or indirect (by acting and interacting in the real world)

Processing

- A fundamental processing step is **registration**
- AR contents and real world data must be properly registered (aligned) to maintain coherence
- With a poor registration, the user does not perceive the augmented content as part of the environment



AR FAILURE

Sources of registration errors

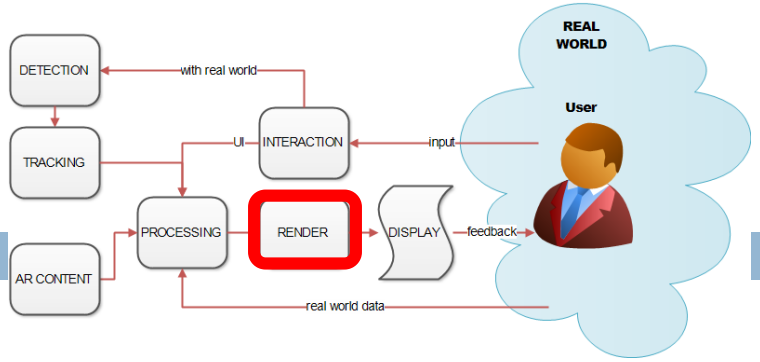
- Static errors
  - Optical distortions
  - Incorrect viewing parameters
- Dynamic errors
  - System delay
  - Tracker errors

Prevention of registration errors

- Optical distortions can be removed by initial (offline) camera calibration
- Incorrect viewing parameters: careful initial setup

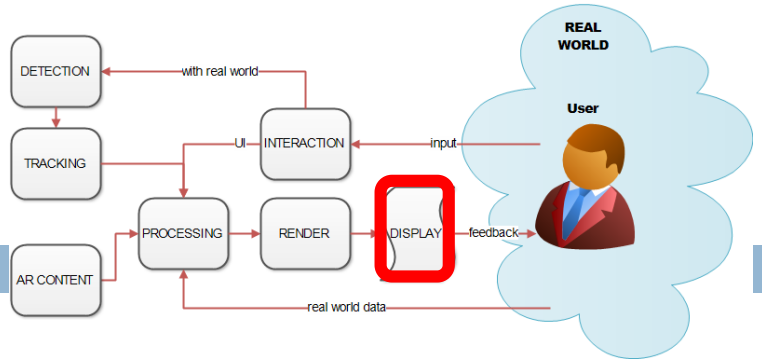
- Tracker errors: use better trackers ☺
- Delay: careful timing for real-time systems

Rendering



- "Rendering" means to create the final augmented image
- Example: if we want to apply a 3D model on the top of a fiducial marker, the ***processing*** module compute the 3D object position, while the ***rendering*** module actually draws it (with textures, shadows, etc.)

Displays



- The display is the device the user uses to perceive the augmented world
  - Monitor
  - Smartphone displays
  - Eyeglasses
  - Head-Mounted Displays (HMD)
  - HUD (head-up displays)
  - …

Displays

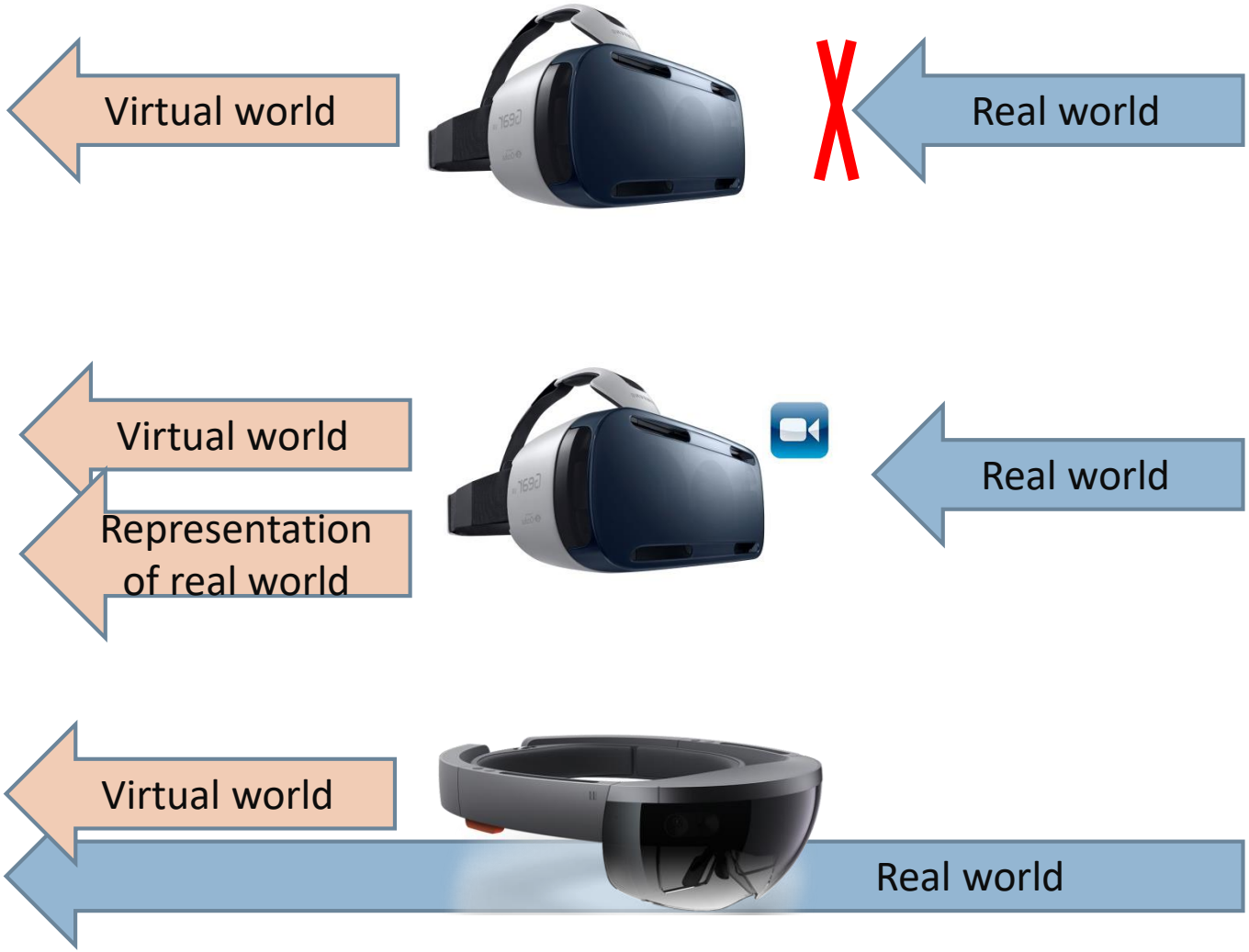
- Monitors offer a low integration with the real world, but are ok for specific applications (e.g. virtual mirror)
- Smartphones give a better integration (can be freely moved and pointed at objects, can implement a fake “see-through” effect)
  - Moreover, smartphones are excellent AR devices because of good cameras, displays and availability of heterogeneous sensors – GPS, compass, etc.

Head Mounted Displays

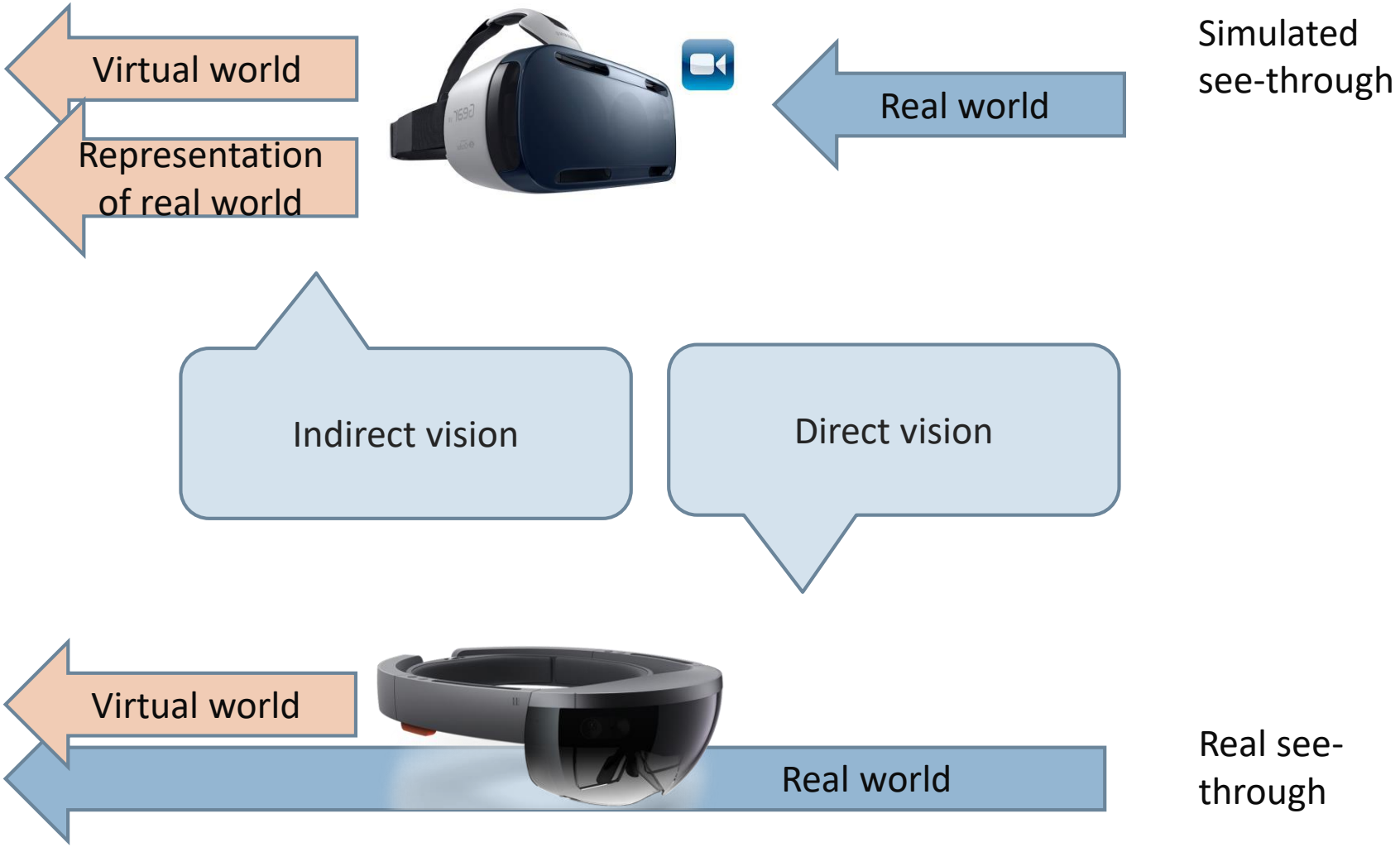
- HMD first appeared in VR contexts
- The same technology can be used for AR if see-through is simulated (indirect vision)
- Some HMD actually implement real see-through… (direct vision)

- Immersive, but cumbersome
- Simulated see-through can introduce delays
- Interaction by a head tracker

HMD


Virtual world
Real world
VR HMD
Virtual world
Representation of real world
Real world
Simulated see-through
Virtual world
Real world
Real see-through

Direct / indirect vision of the real world



Eyeglasses

- A particular type of HMD, designed to be small and comfortable as a pair of real eyeglasses
- Always real see-through augmentation
- Do not require external processing unit (everything is integrated in the glasses)

Eyeglasses and HMD examples

Google Glass

Microsoft hololens

Altri esempi:
http://www.tomshw.it/articoli/i-migliori-occhiali-per-la-realta-aumentata-del-momento-72198

Eyeglasses: an application example

- Sony Entertainment Access Glasses
- Currently used in some cinemas
- Show subtitles synchronized with the movie for deaf people

https://pro.sony.com/bbsc/ssr/product-STWC140GI/