LECTURE 8: AR TECHNOLOGY

COMP 4010 – Virtual Reality
Semester 5 – 2017

Bruce Thomas, Mark Billinghurst
University of South Australia

October 5th 2017

University of South Australia

Augmented Reality

1977 – Star Wars

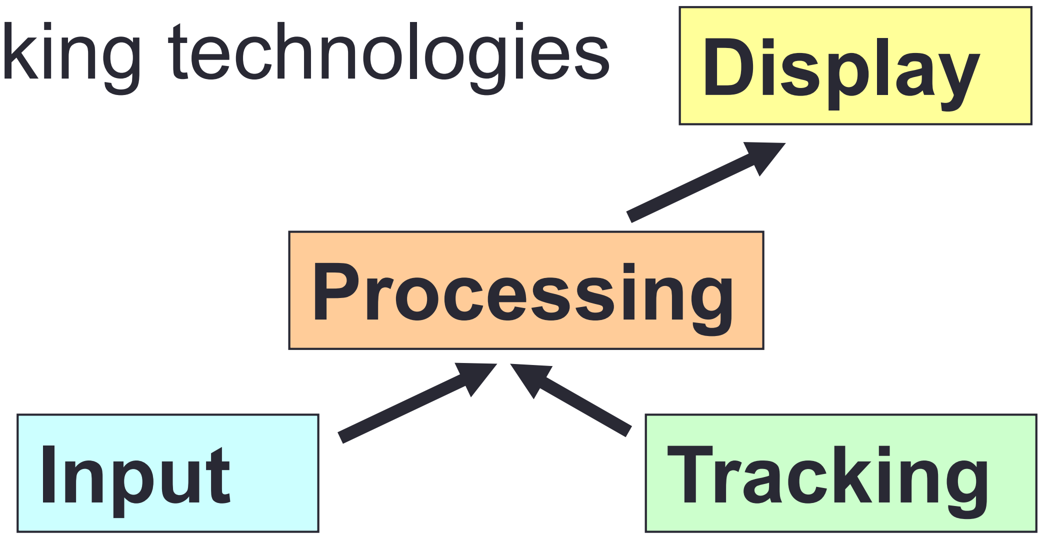
Augmented Reality Definition

- Defining Characteristics [Azuma 97]:
  - Combines Real and Virtual Images
    - Both can be seen at the same time
  - Interactive in real-time
    - The virtual content can be interacted with
  - Registered in 3D
    - Virtual objects appear fixed in space

Azuma, R. T. (1997). A survey of augmented reality. Presence, 6(4), 355-385.

Technology Requirements

- Combining Real and Virtual Images
  - Needs - Display technologies
- Interactive in Real-Time
  - Needs - Input and interactive technologies
- Registered in 3D
  - Needs - Viewpoint tracking technologies

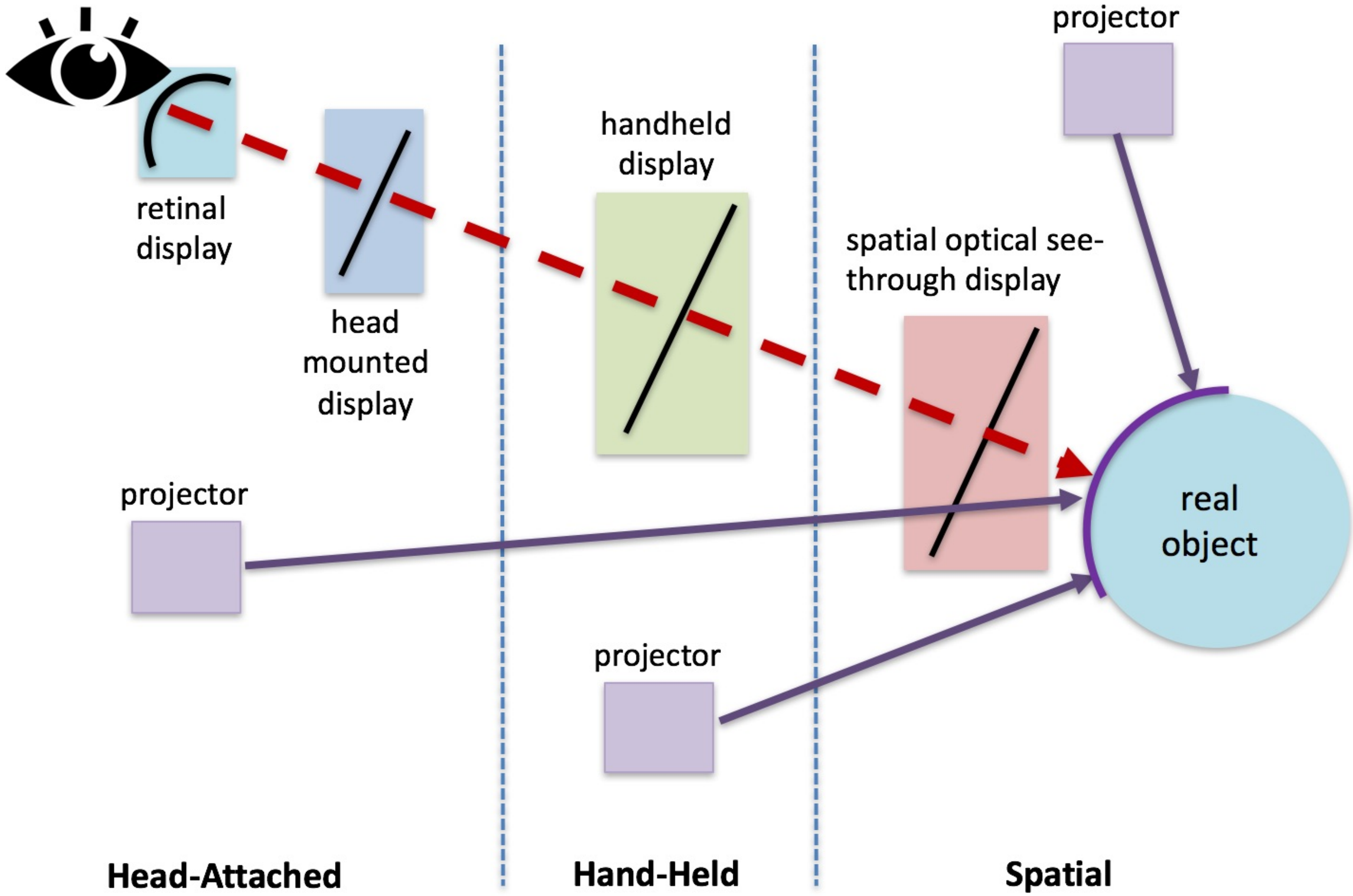


AR DISPLAYS

Display Technologies

- Types (Bimber/Raskar 2005)
- Head attached
  - Head mounted display/projector
- Body attached
  - Handheld display/projector
- Spatial
  - Spatially aligned projector/monitor

Display Taxonomy

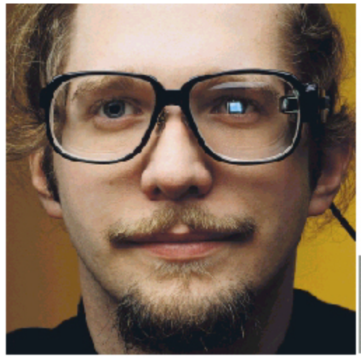


Bimber, O., & Raskar, R. (2005). *Spatial augmented reality: merging real and virtual worlds*. CRC press.

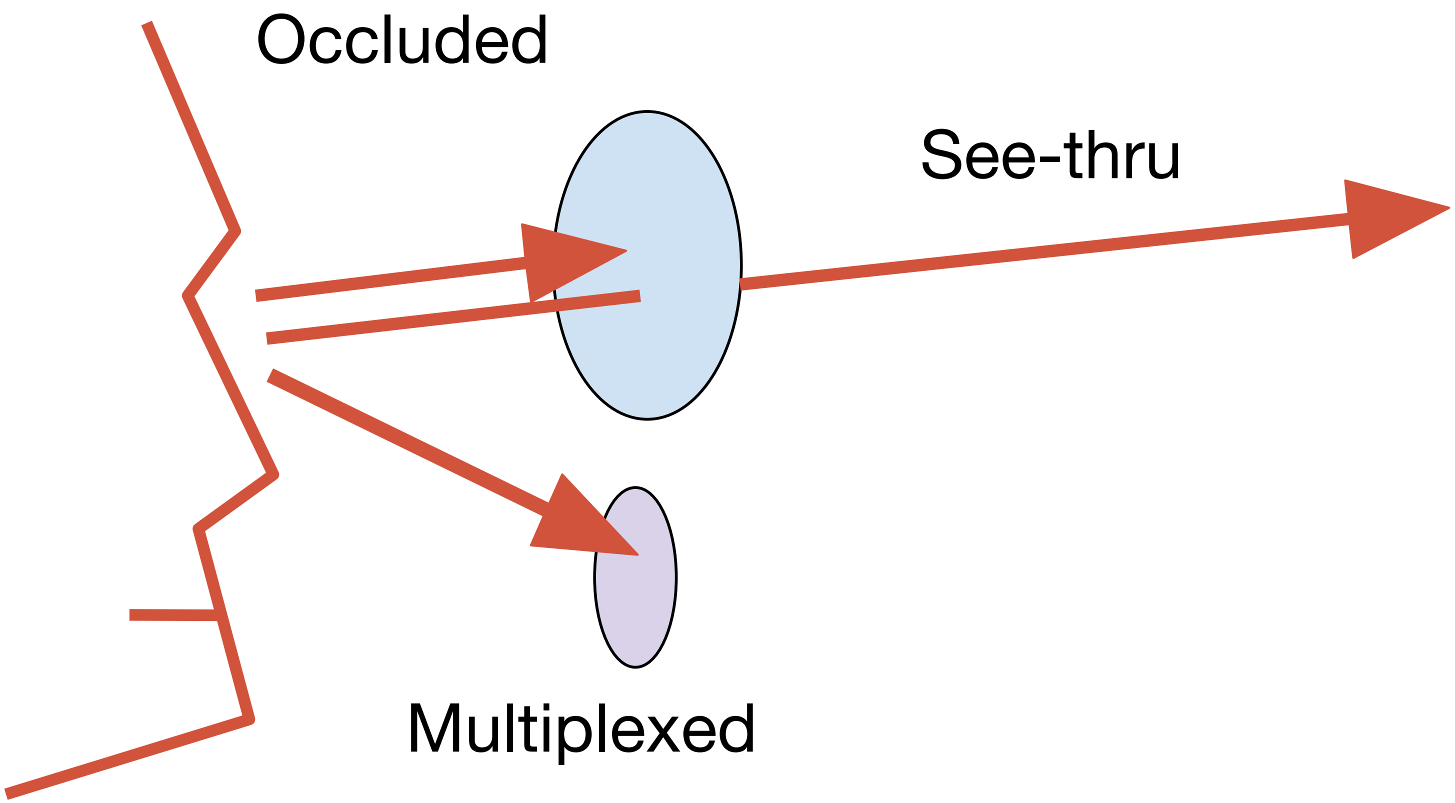
HEAD MOUNTED DISPLAYS

Head Mounted Displays (HMD)

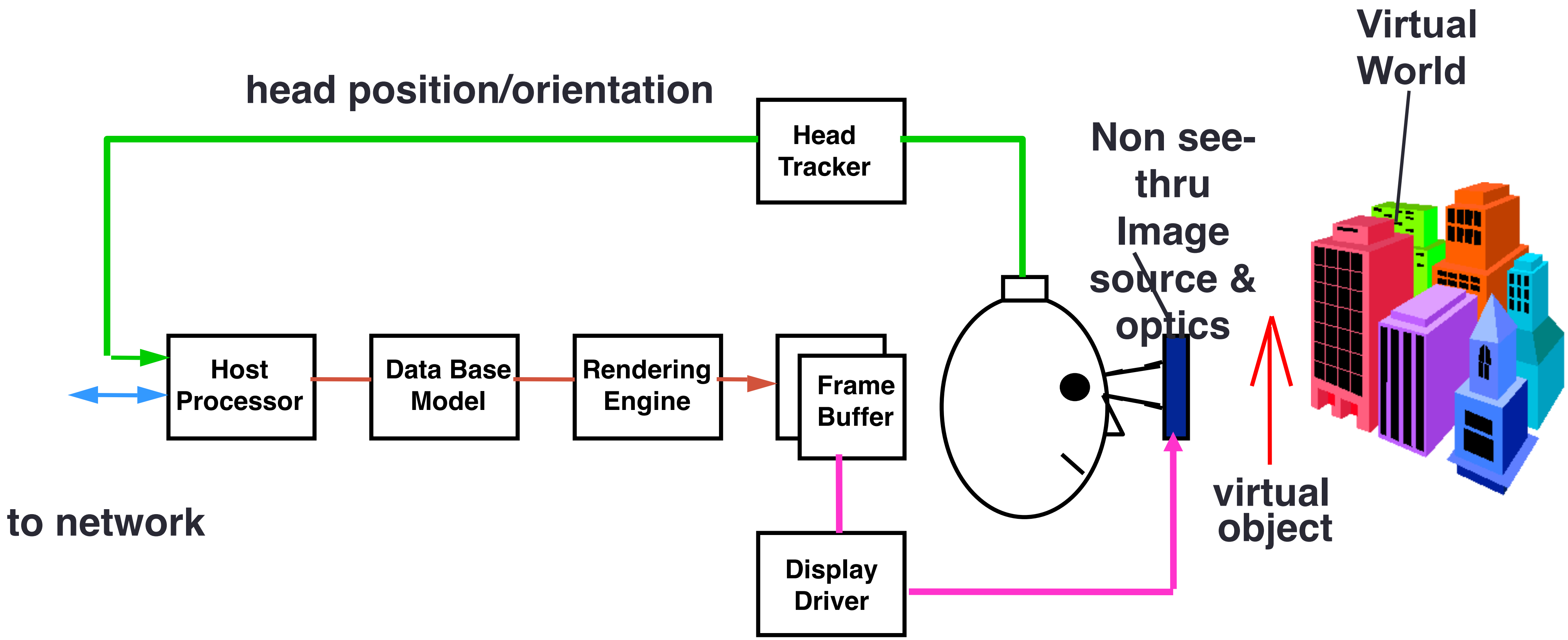
- Display and Optics mounted on Head
- May or may not fully occlude real world
- Provide full-color images
- Considerations
  - Cumbersome to wear
  - Brightness
  - Low power consumption
  - Resolution limited
  - Cost is high?



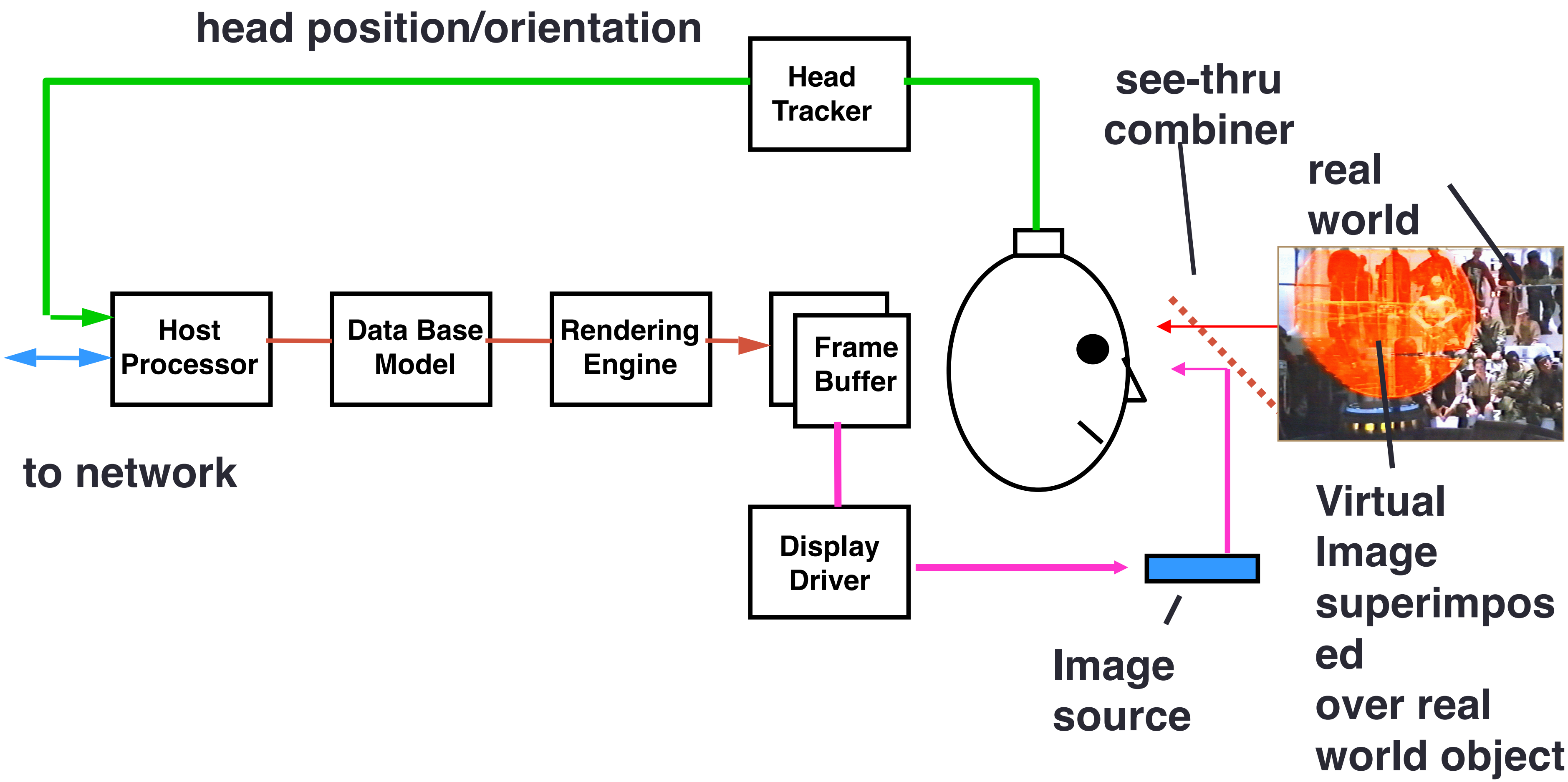
Types of Head Mounted Displays



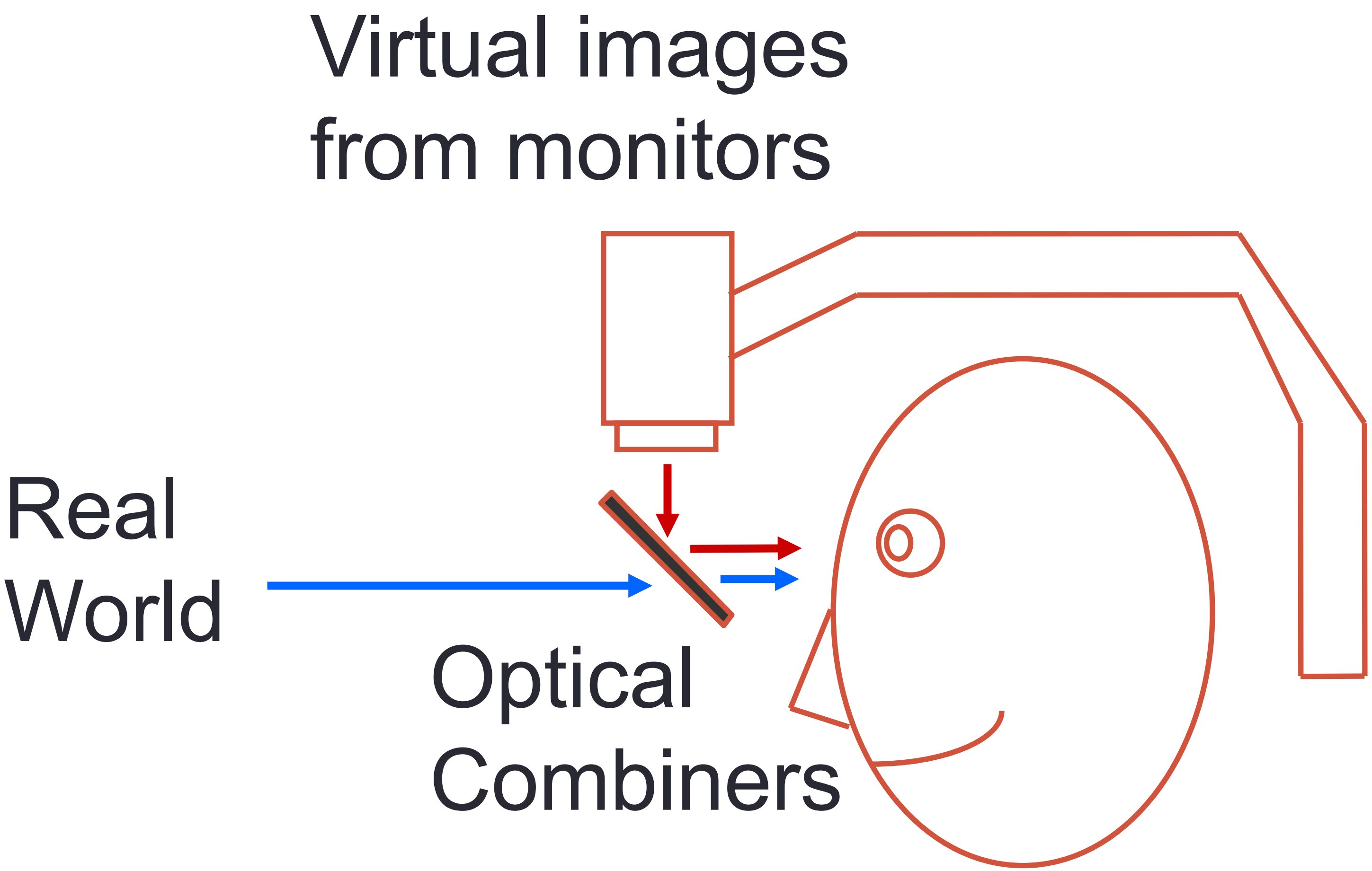
Immersive VR Architecture



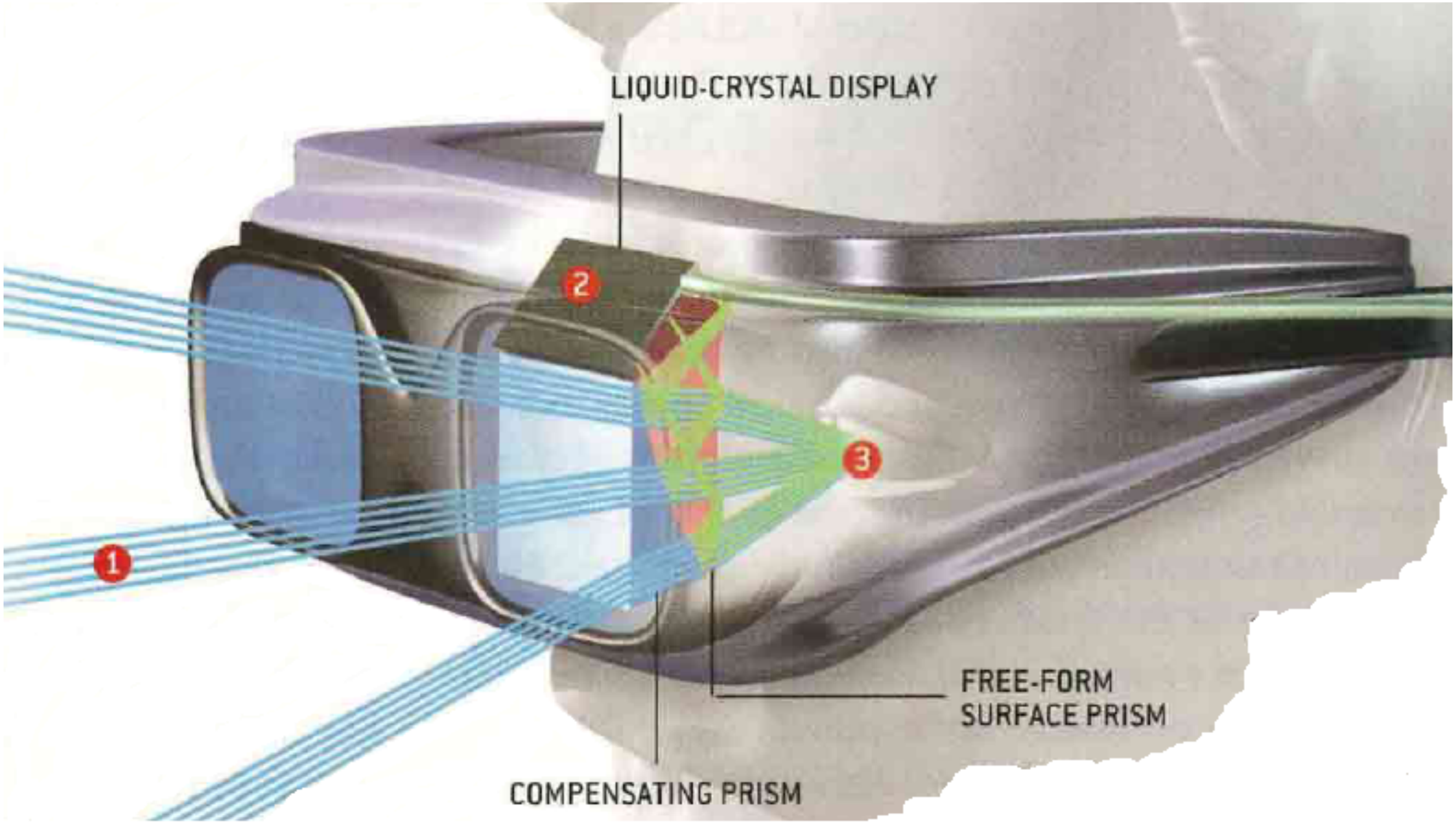
See-thru AR Architecture



Optical see-through Head-Mounted Display



Optical See-Through HMD



Example: Epson Moverio BT-200

- Stereo see-through display ($500)
  - 960 x 540 pixels, 23 degree FOV, 60Hz, 88g
  - Android Powered, separate controller
  - VGA camera, GPS, gyro, accelerometer

View Through Optical See-Through HMD

Other Optical See-Through Displays

- Microsoft HoloLens - $3,000 USD
  - Wearable computer
  - Waveguide displays
  - https://www.microsoft.com/hololens

- Meta Meta2 - $1,500 USD
  - Wide field of view (90 degrees)
  - Tethered display
  - https://www.metavision.com/

- Mira Prism - $100 USD
  - Smart phone based
  - Wide field of view
  - https://www.mirareality.com/

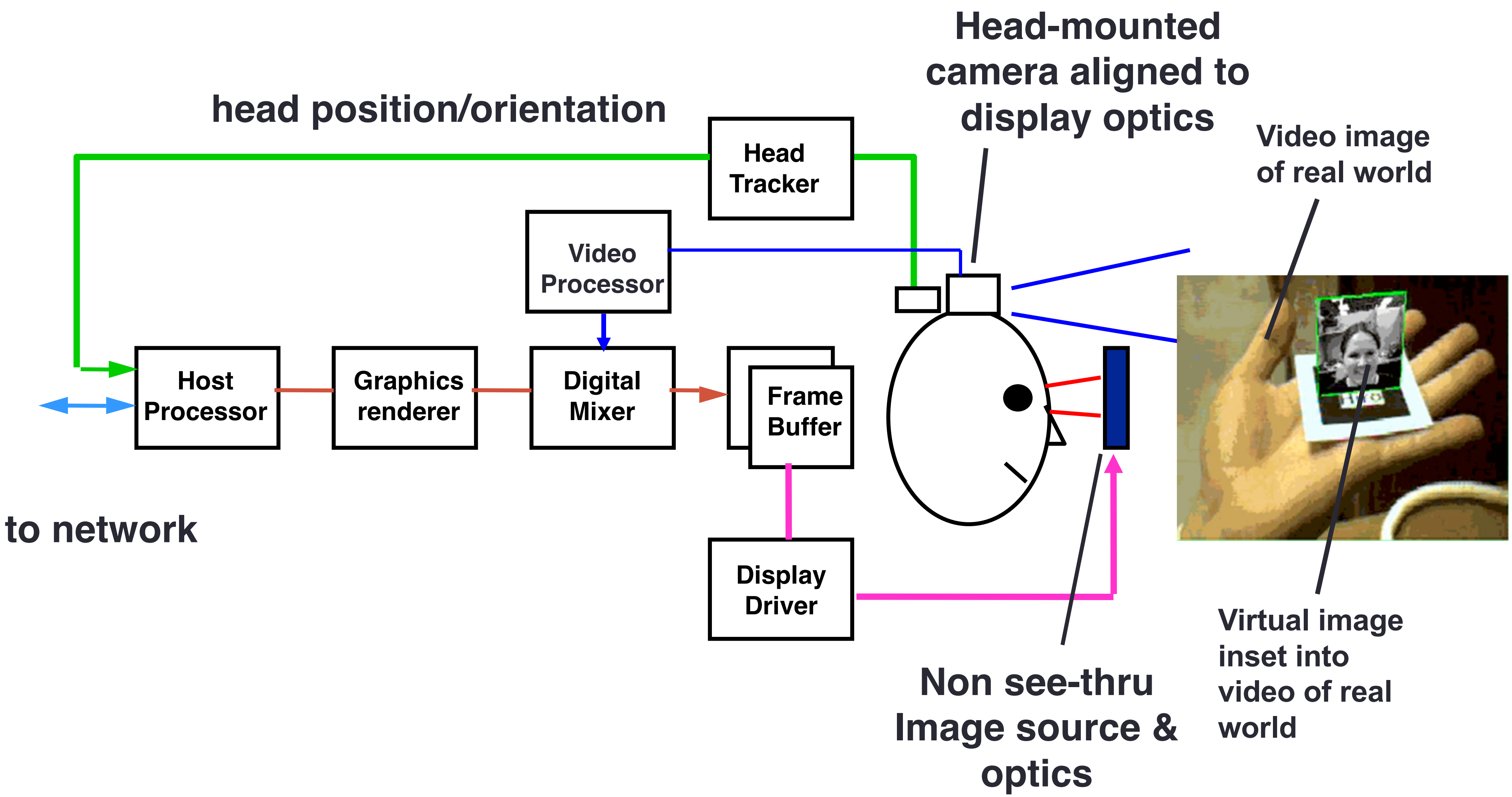
Example: HoloLens

- https://www.youtube.com/watch?v=AaTyeDtht-8

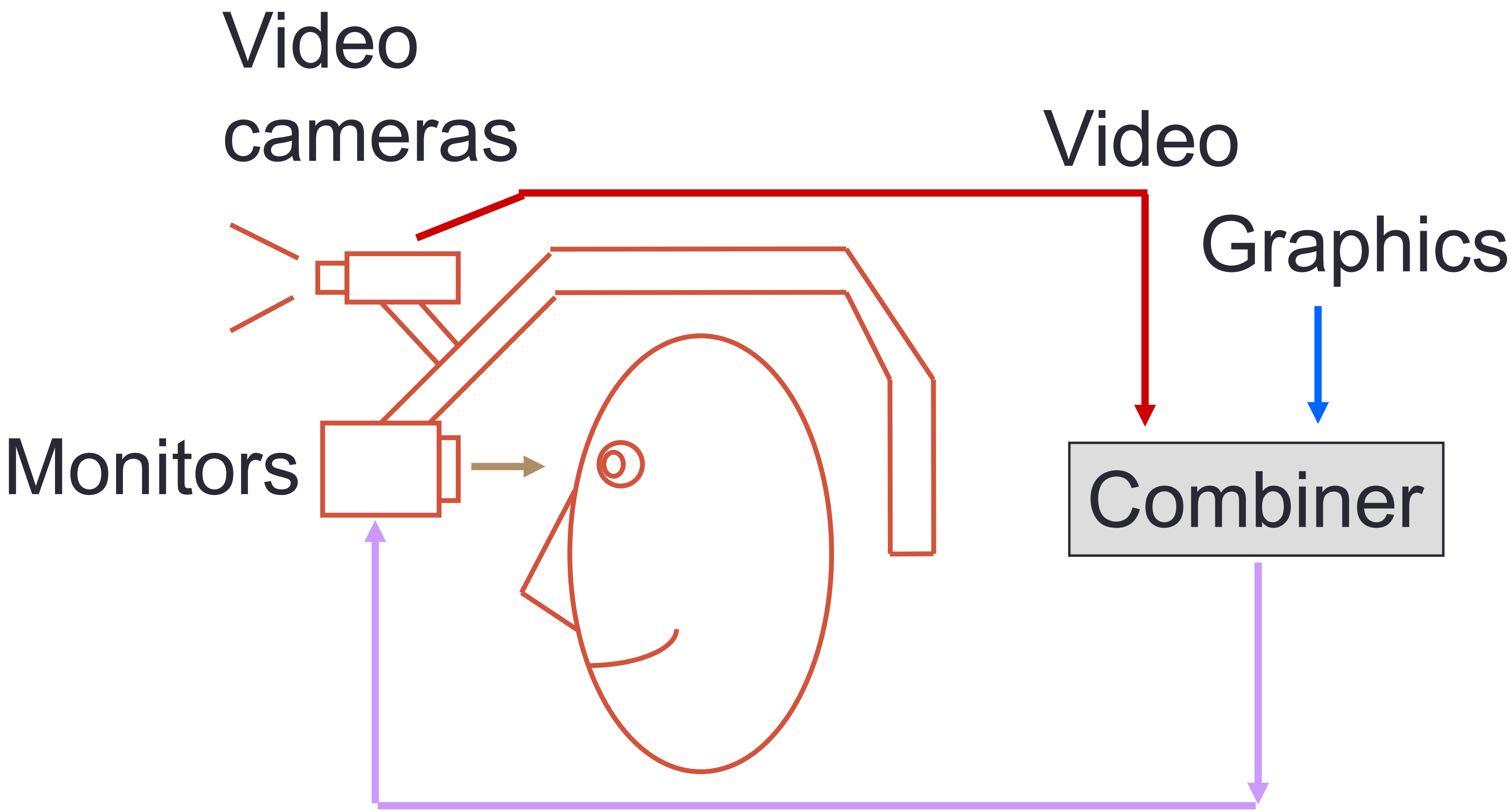
Strengths of Optical see-through AR

- Simpler (cheaper)
- Direct view of real world
  - Full resolution, no time delay (for real world)
  - Safety
  - Lower distortion
- No eye displacement
  - but some video see-through displays avoid this

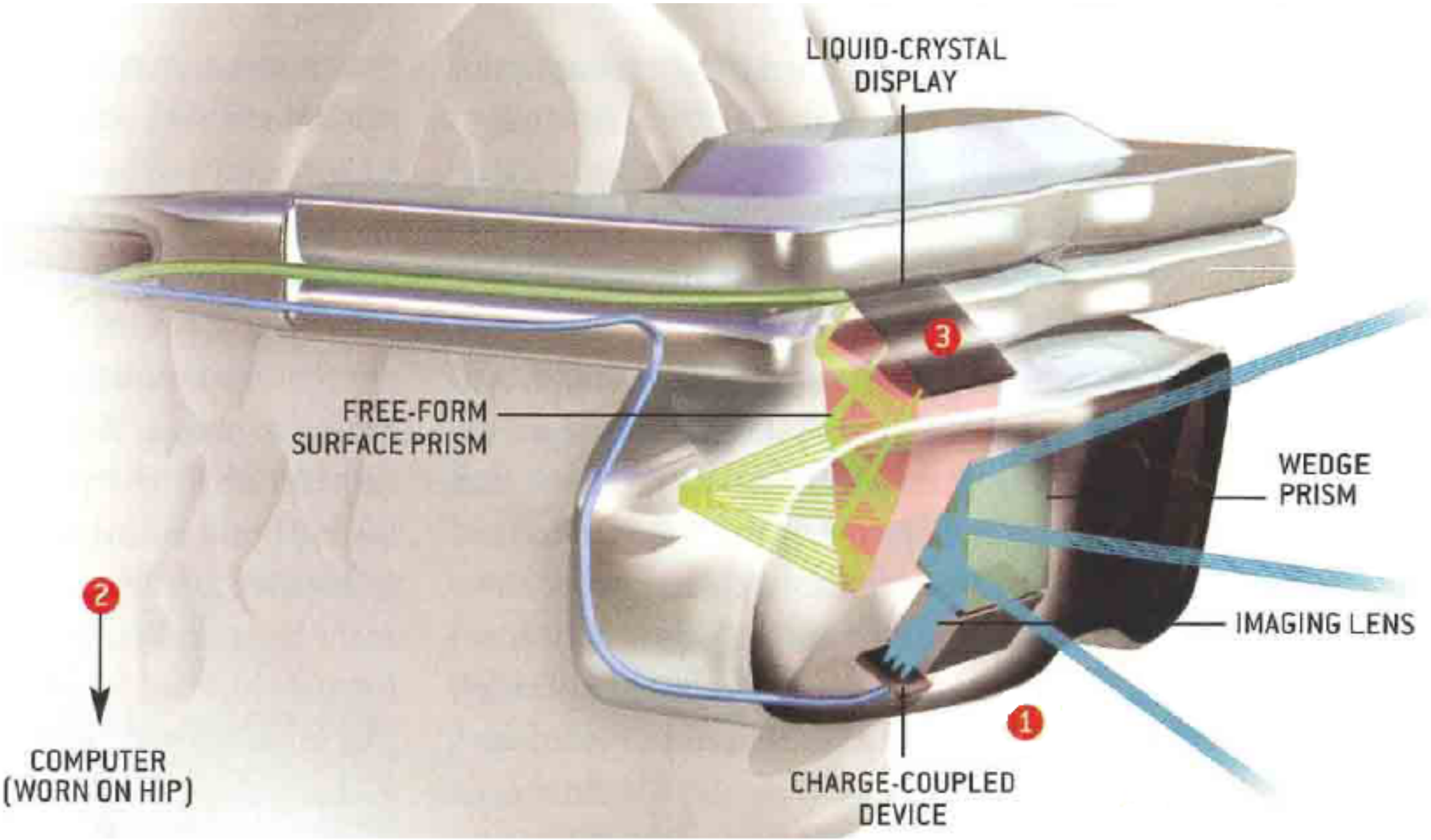
Video AR Architecture



Video see-through HMD



Video See-Through HMD



Vuzix Wrap 1200DXAR

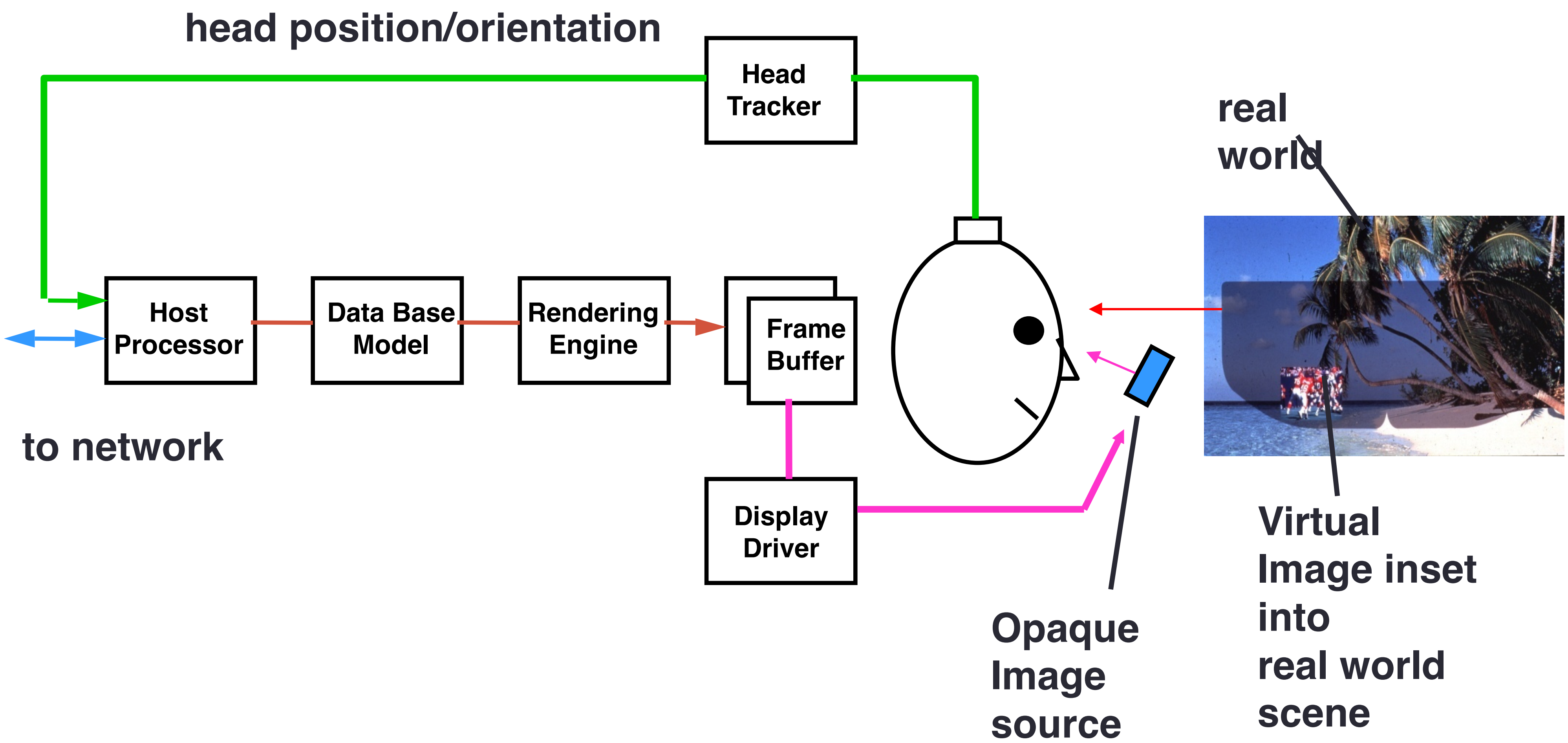
- Stereo video see-through display ($1500)
  - Twin 852 x 480 LCD displays, 35 deg. FOV
  - Stereo VGA cameras
  - 3 DOF head tracking
  - https://www.vuzix.com/Products/LegacyProduct/4

View Through a Video See-Through HMD

Strengths of Video See-Through AR

- True occlusion
  - AR pixels block video pixels
- Digitized image of real world
  - Flexibility in composition
  - Matchable time delays
  - More registration, calibration strategies
- Wide FOV is easier to support

Eye Multiplexed AR Architecture

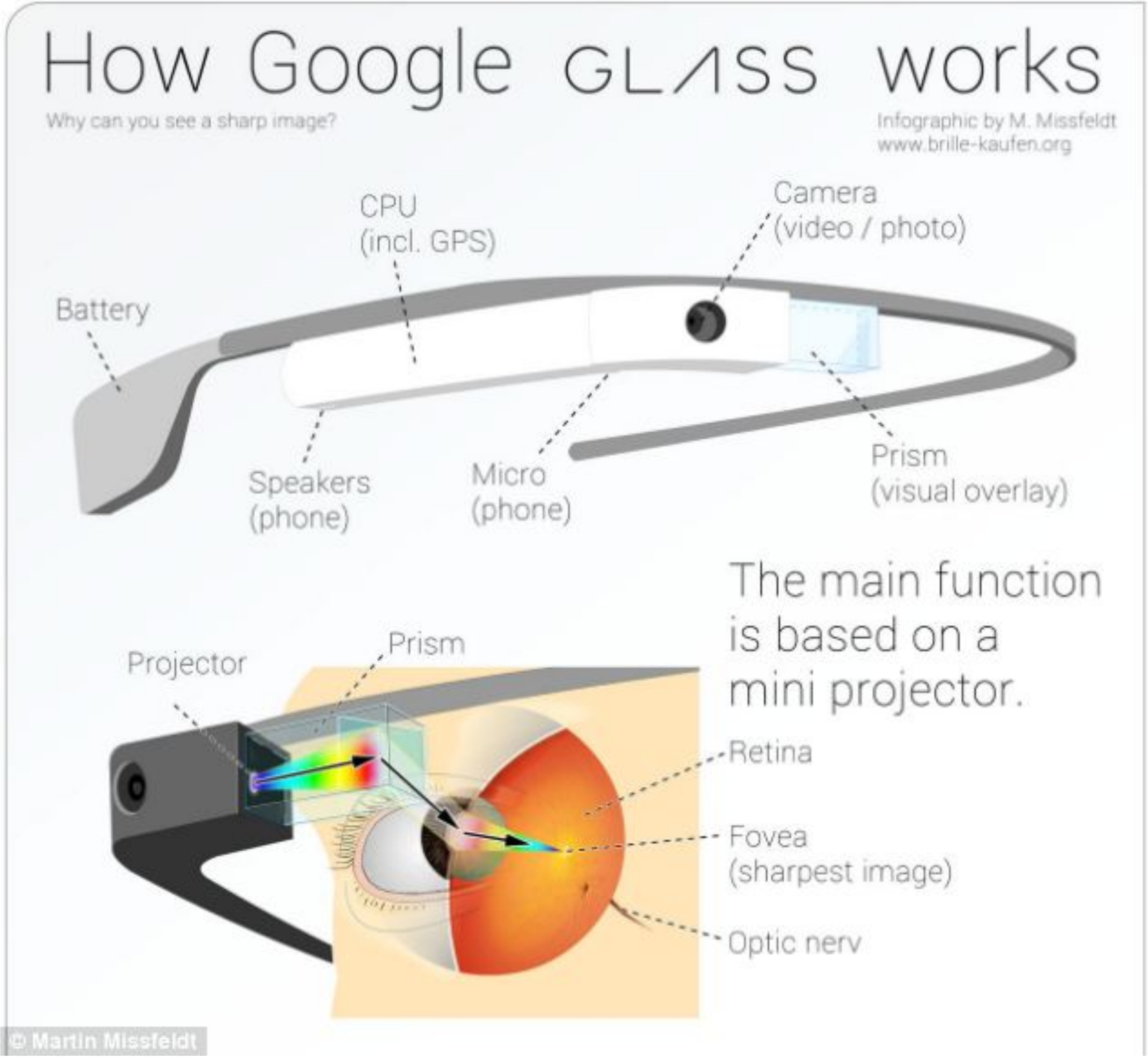


Virtual Image 'inset' into Real World

Example: Google Glass

View Through Google Glass

https://www.youtube.com/watch?v=zKNv505sYAM

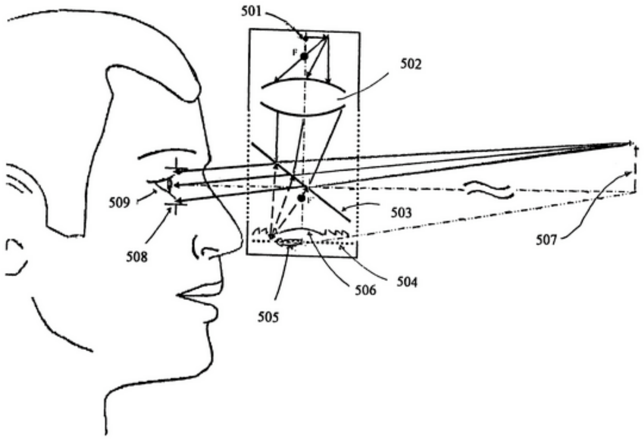

How Google GLASS works
Why can you see a sharp image?
Infographic by M. Missfeldt
www.brille-kaufen.org
CPU
(incl. GPS)
Camera
(video / photo)
Battery
Speakers
(phone)
Micro
(phone)
Prism
(visual overlay)
The main function
is based on a
mini projector.
Projector
Prism
Retina
Fovea
(sharpest image)
Optic nerv
© Martin Missfeldt

Vuzix M-100

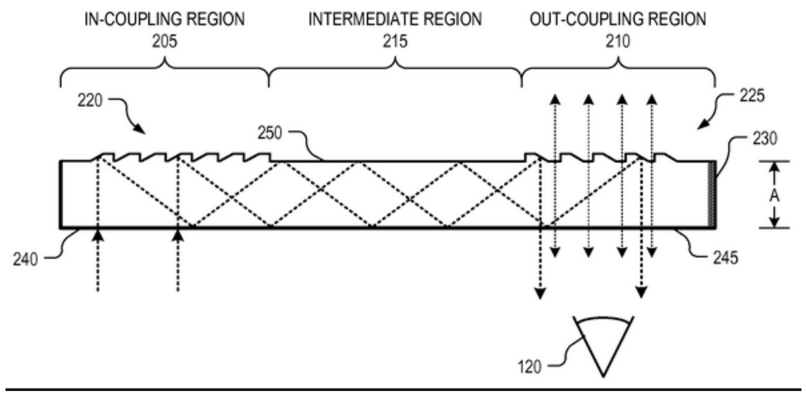
- Monocular multiplexed display ($1000)
  - 852 x 480 LCD display, 15 deg. FOV
  - 5 MP camera, HD video
  - GPS, gyro, accelerometer

Display Technology

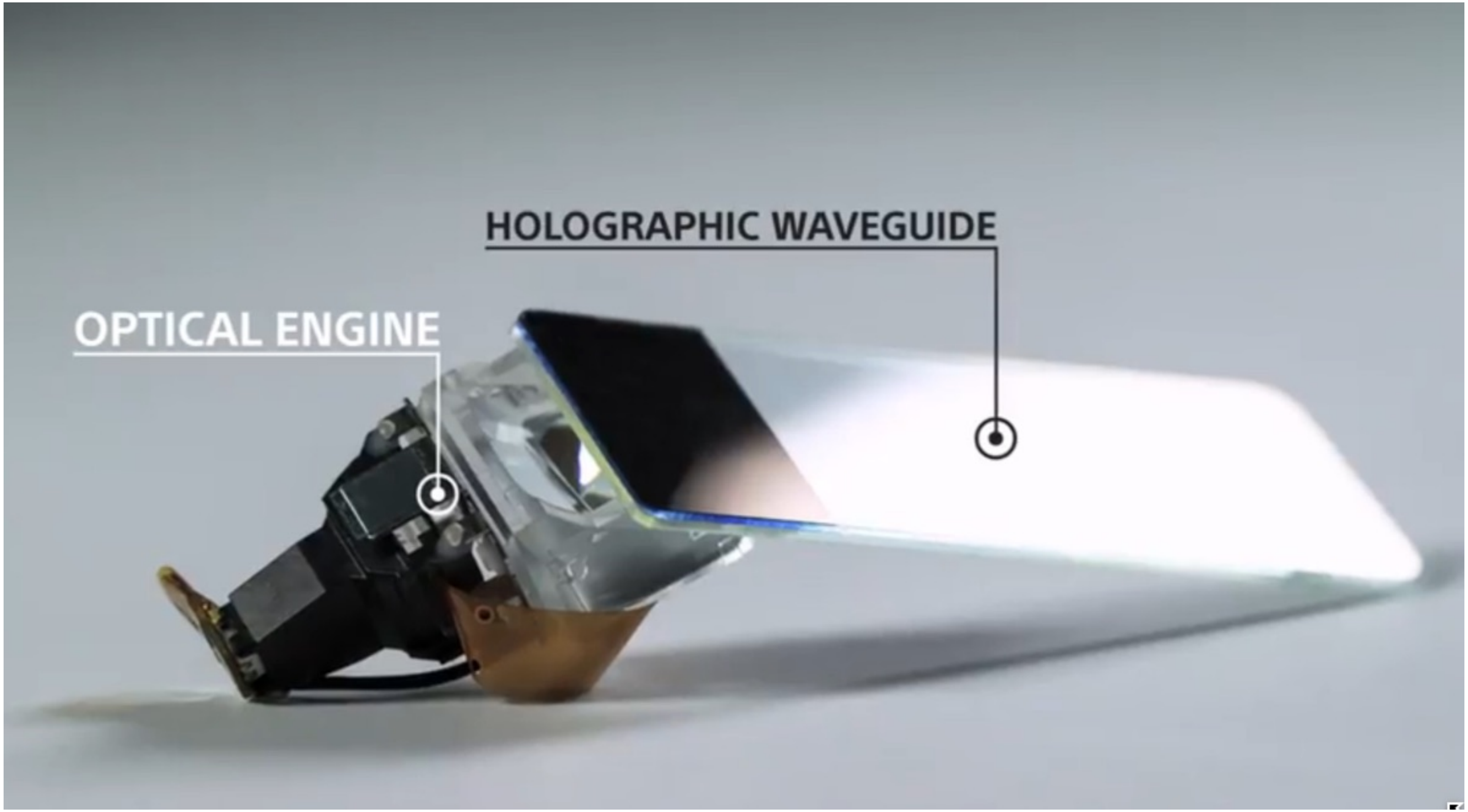
- Curved Mirror
  - off-axis projection
  - curved mirrors in front of eye
  - high distortion, small eye-box



- Waveguide
  - use internal reflection
  - unobstructed view of world
  - large eye-box



Example: Sony Waveguide Display



- https://www.youtube.com/watch?v=G2MtI7asLcA

See-through Thin Displays (Waveguide)

Opinvent Ora

Lumus DK40

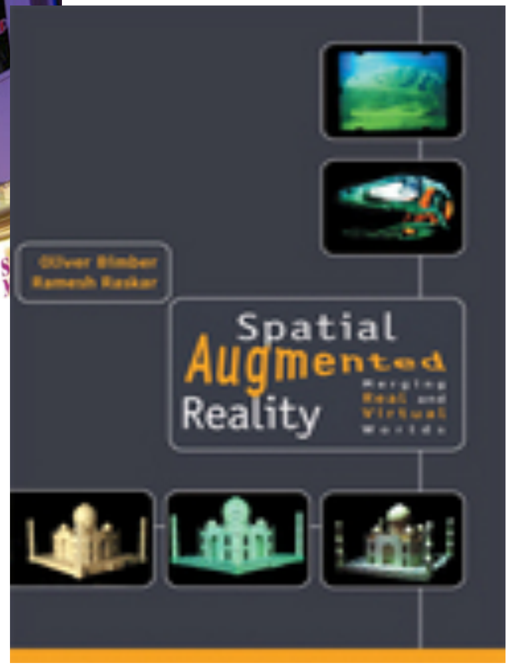
- Waveguide techniques for thin see-through displays
  - Wider FOV, enable AR applications
  - Social acceptability

Example: Sony Smart EyeGlasses

- https://www.youtube.com/watch?v=kYPWaMsarss

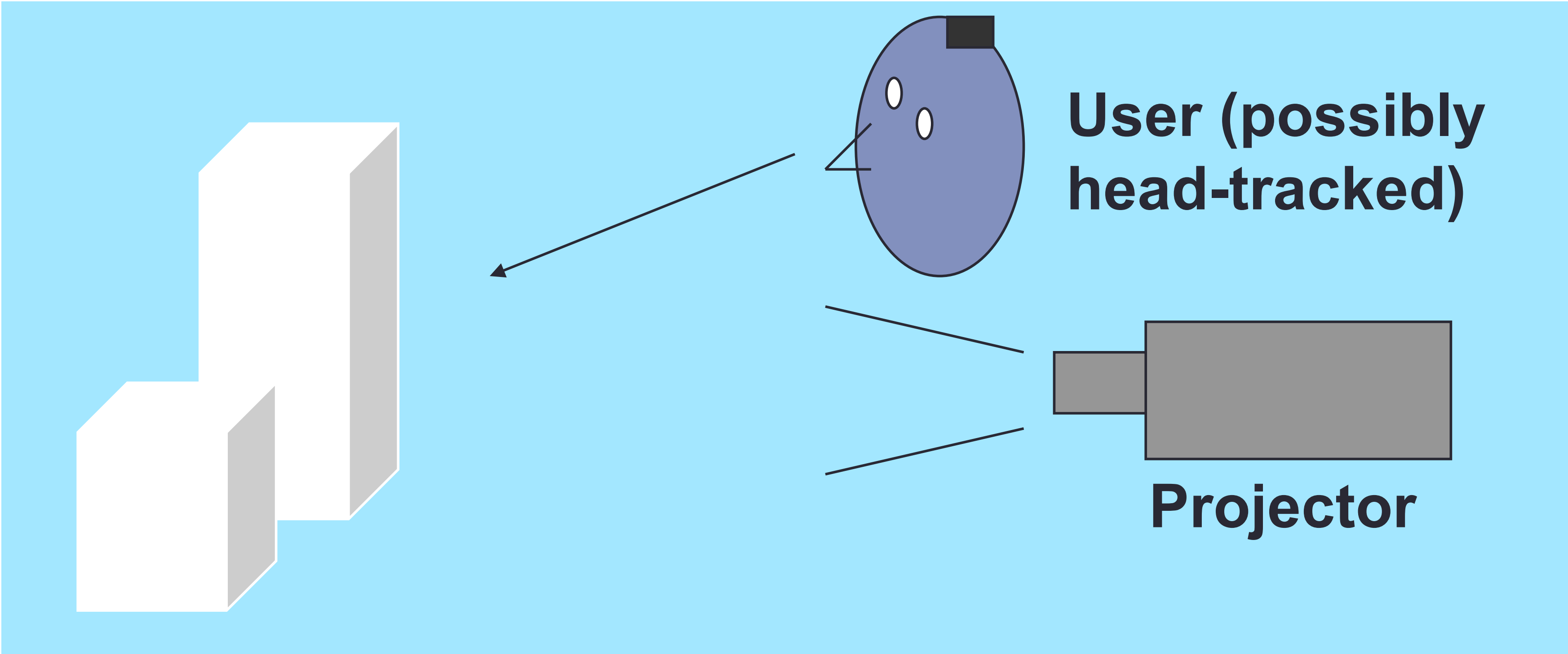
SPATIAL AUGMENTED REALITY

Spatial Augmented Reality



- Project onto irregular surfaces
  - Geometric Registration
  - Projector blending, High dynamic range
- Book: Bimber, Rasker “Spatial Augmented Reality”

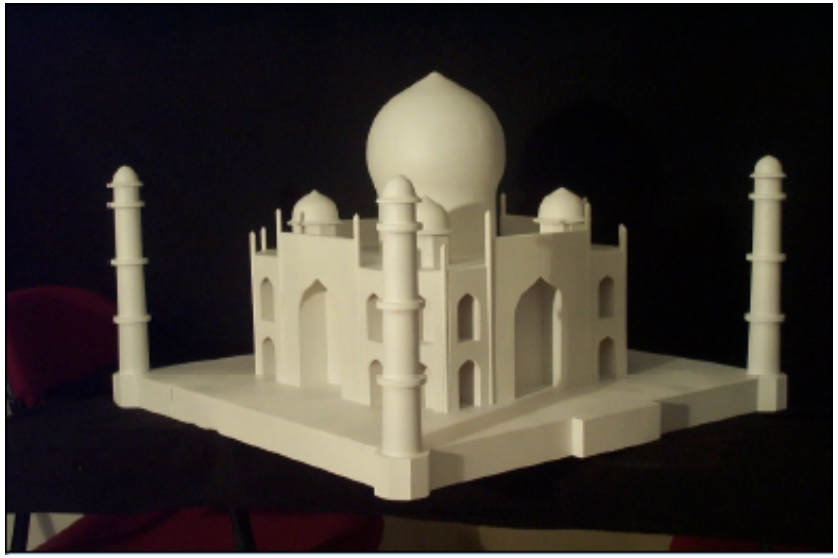
Projector-based AR

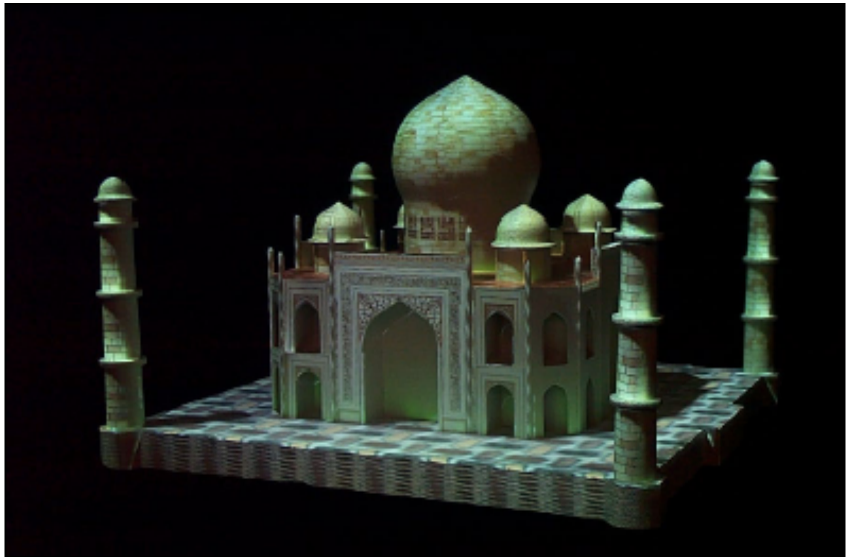


Real objects with retroreflective covering

Examples:
Raskar, MIT Media Lab
Inami, Tachi Lab, U. Tokyo

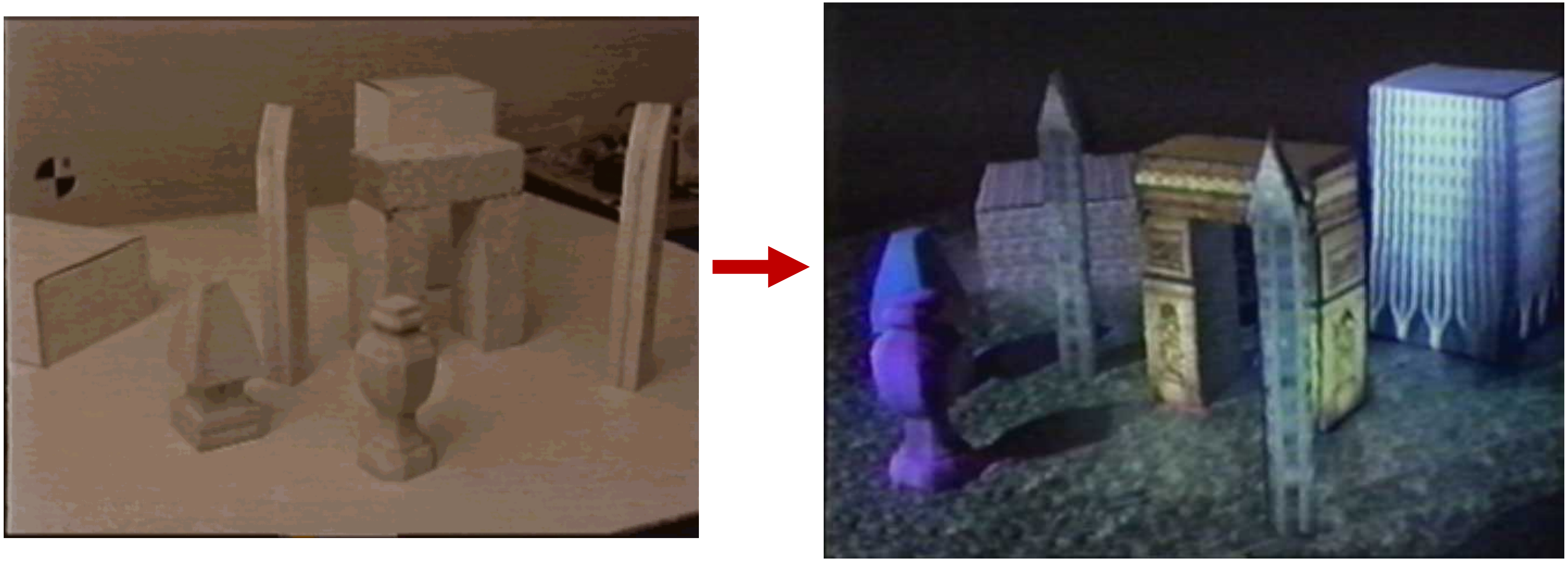
Example of Projector-Based AR





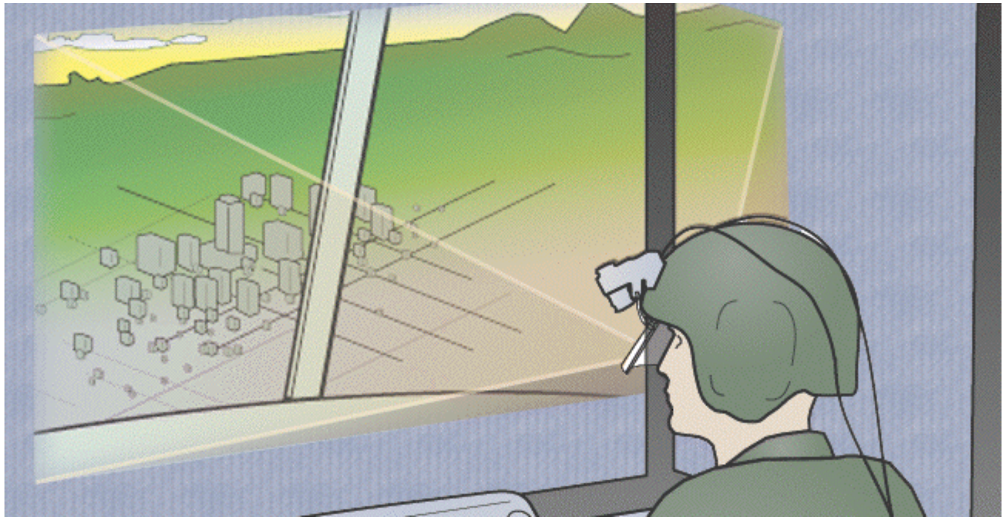
Ramesh Raskar, MIT

Example of Projector-Based AR



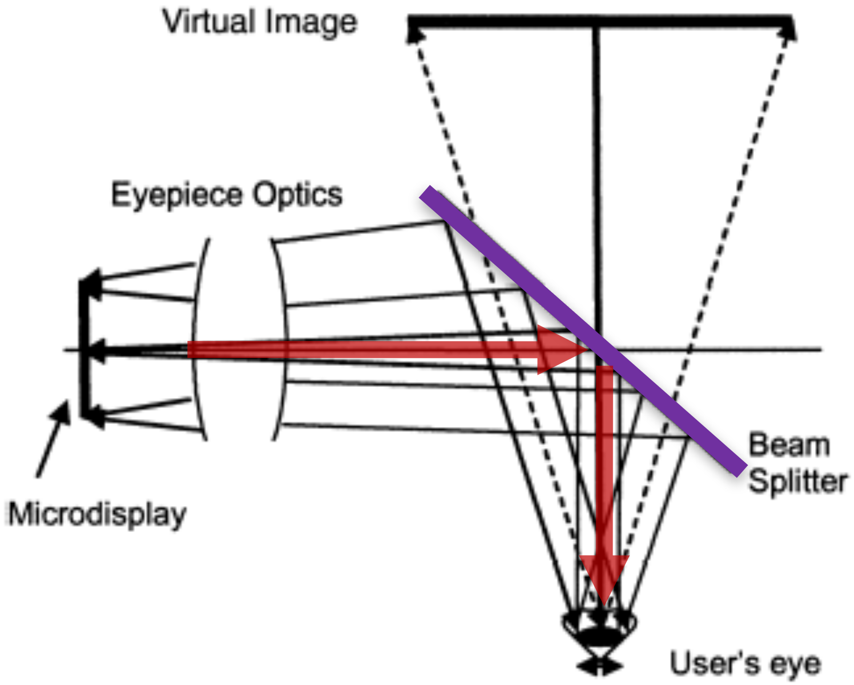
Ramesh Raskar, MIT

Head Mounted Projector

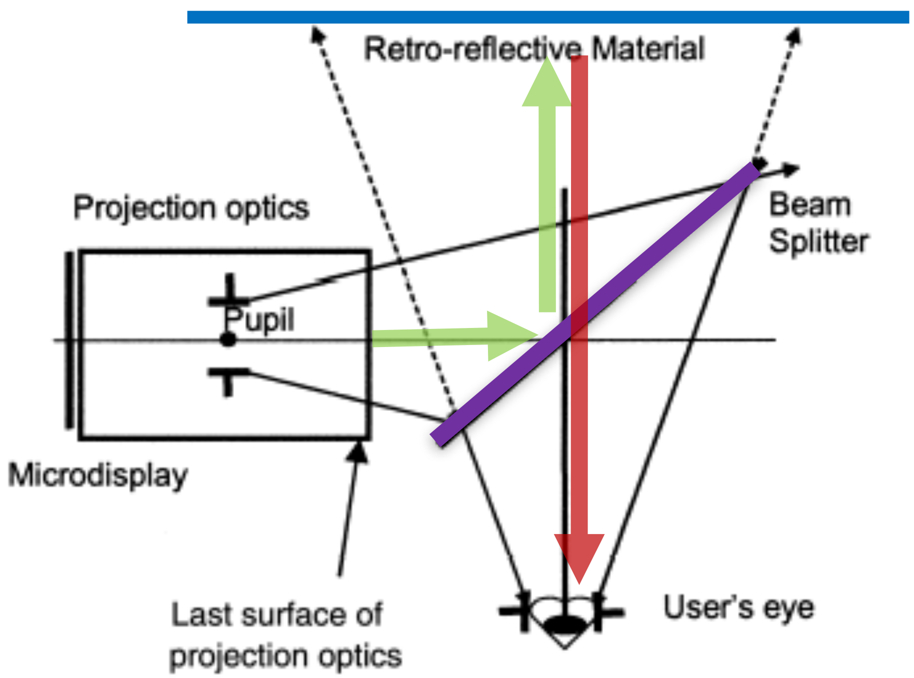


- NVIS P-50 HMPD
  - 1280x1024/eye
  - Stereoscopic
  - 50 degree FOV
  - www.nvis.com

HMD vs. HMPD



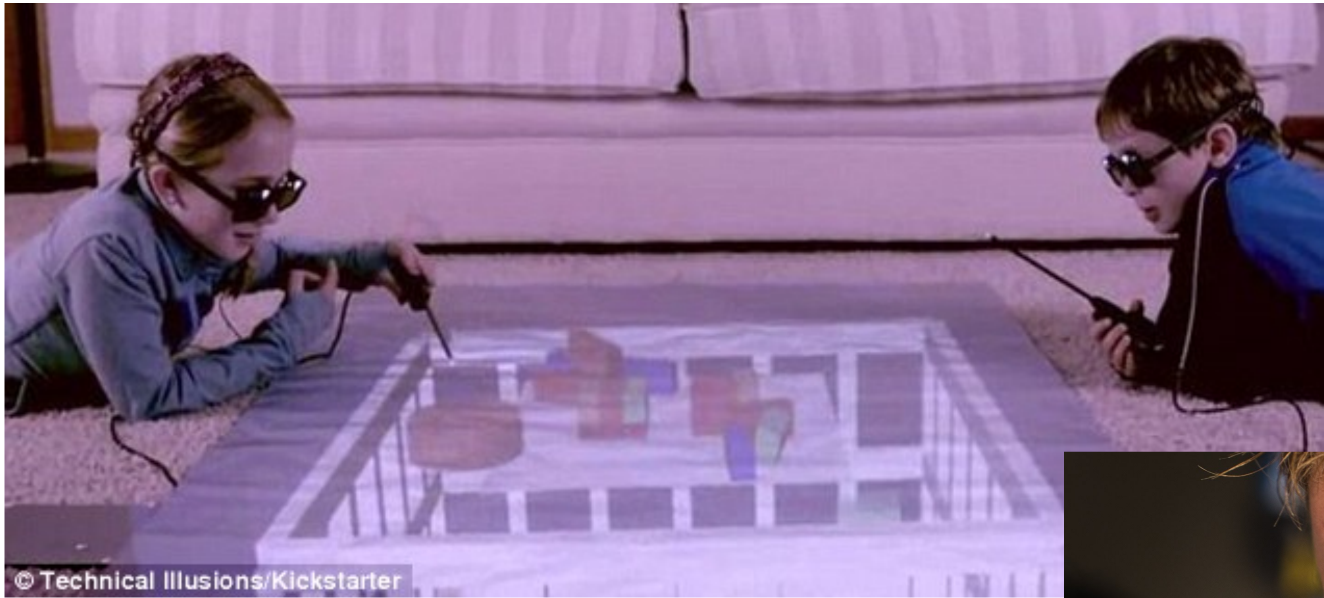
Head Mounted Display



Head Mounted Projected Display

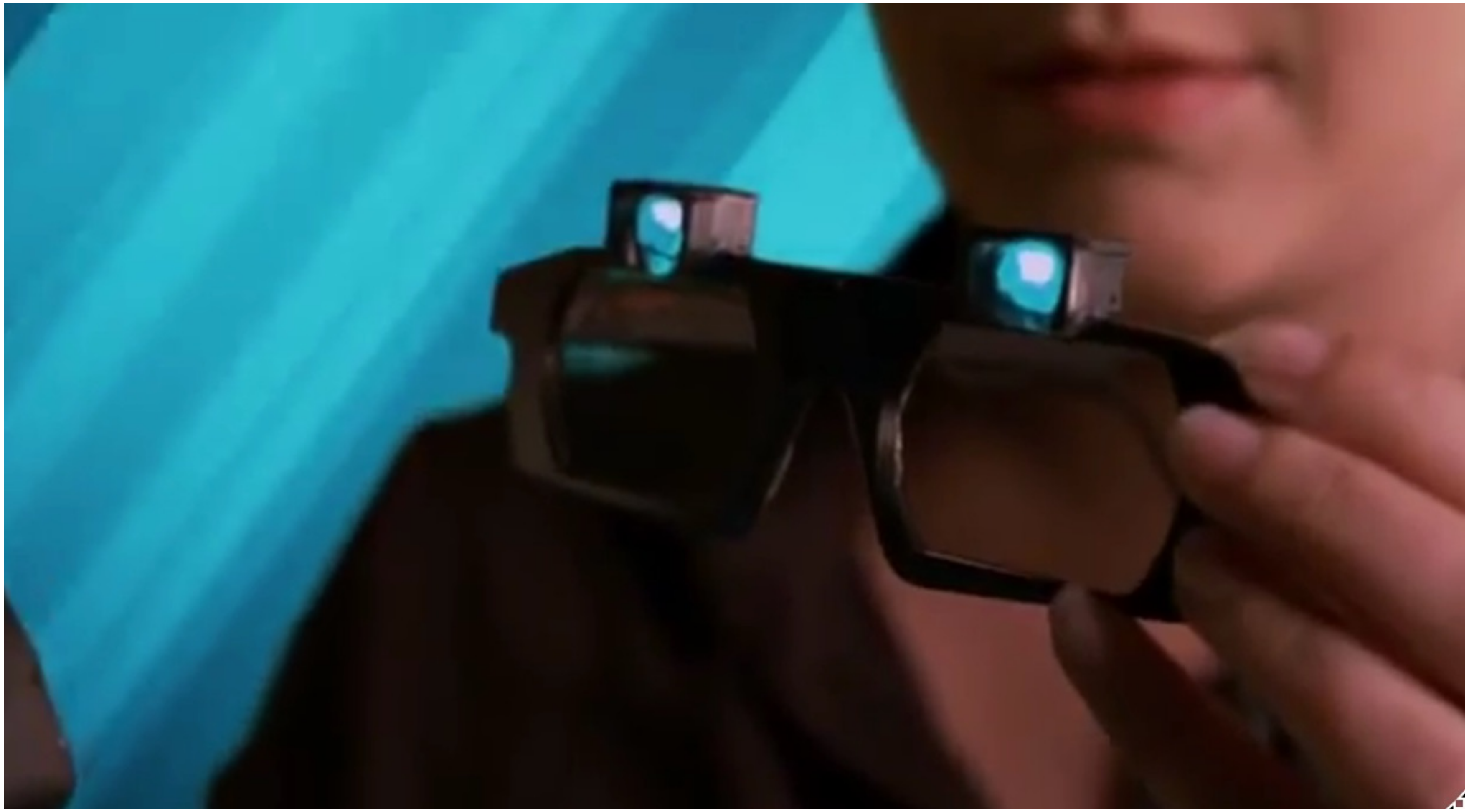
CastAR - http://technicalillusions.com/

- Stereo head worn projectors
- Interactive wand
- Rollable retro-reflective sheet



- Designed for shared interaction

Demo: CastAR



https://www.youtube.com/watch?v=AOI5UW9khoQ#t=47

Pico Projectors

- Microvision - www.mvis.com
- 3M, Samsung, Philips, etc

Demo: Pico Projector

https://www.youtube.com/watch?v=vtdH3CLKEuY

MIT Sixth Sense

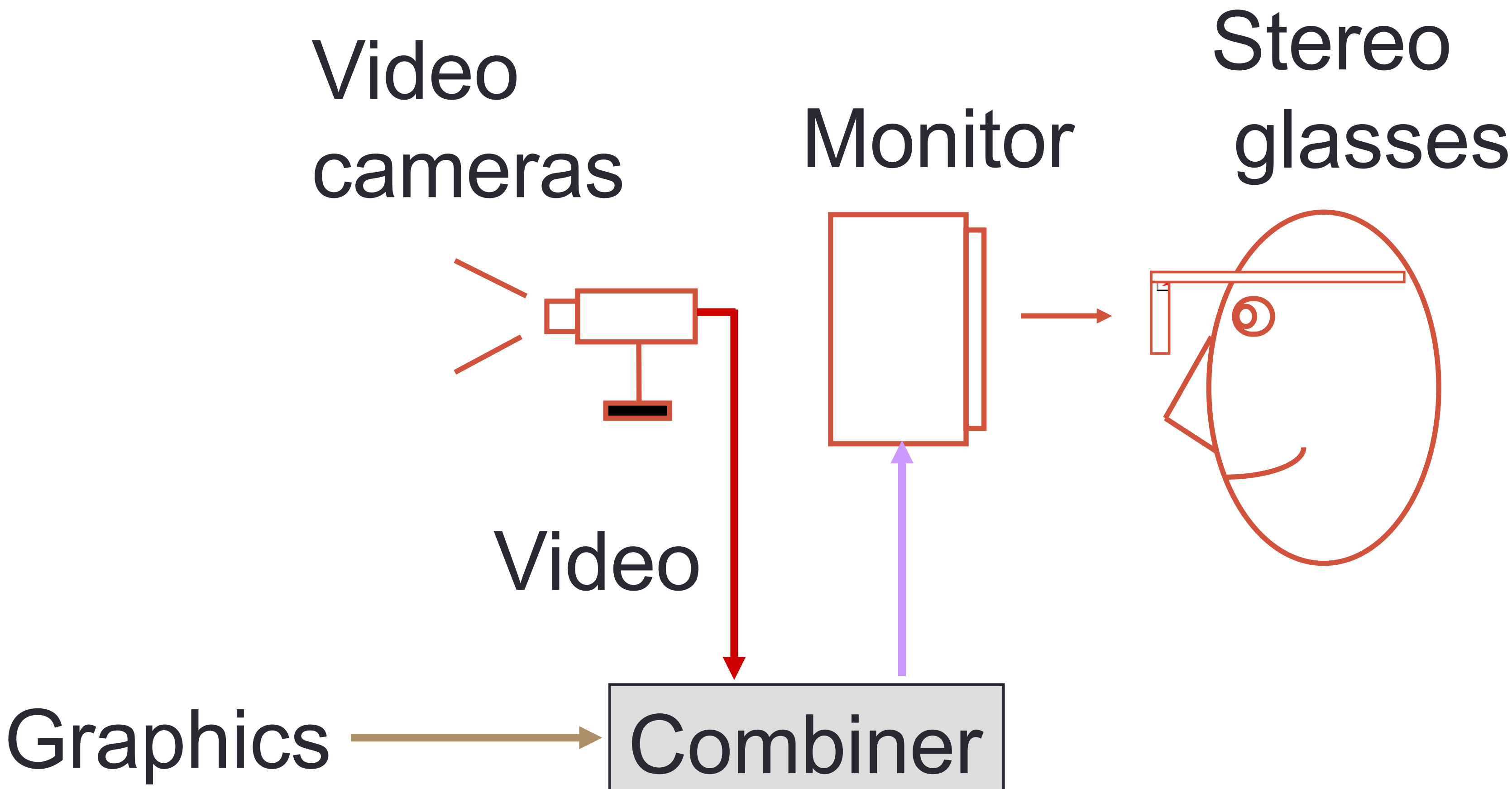
- Body worn camera and projector
- http://www.pranavmistry.com/projects/sixthsense/

Demo: Sixth Sense

https://www.youtube.com/watch?v=Q4Z9sOtiWUY

OTHER AR DISPLAYS

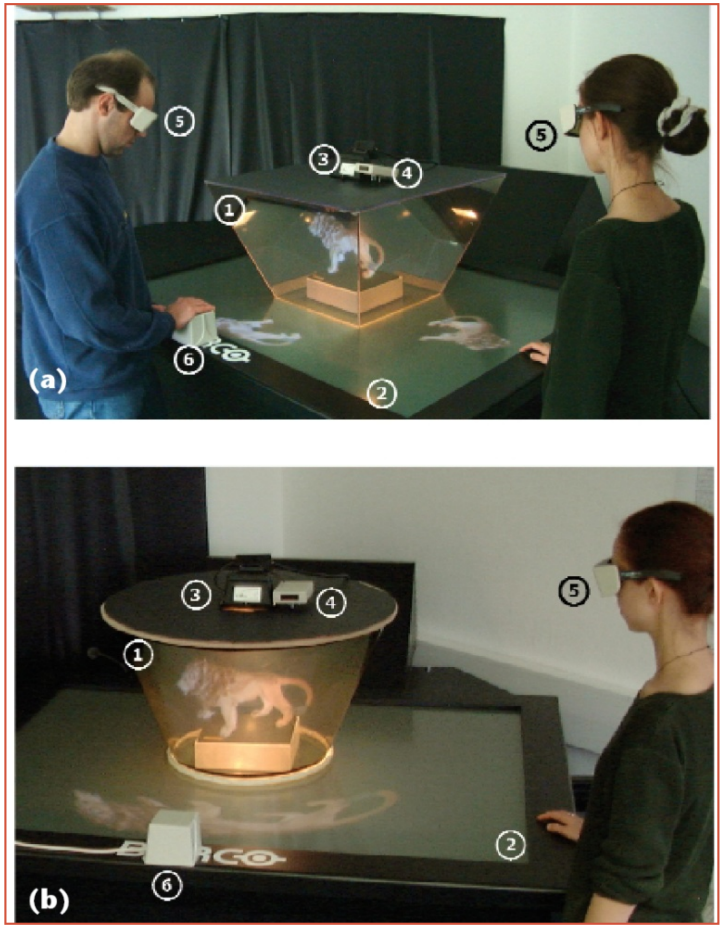
Video Monitor AR



Examples

Virtual Showcase

- Mirrors on a projection table
  - Head tracked stereo
  - Up to 4 users
  - Merges graphic and real objects
  - Exhibit/museum applications
- Fraunhofer Institute (2001)
  - Bimber, Frohlich

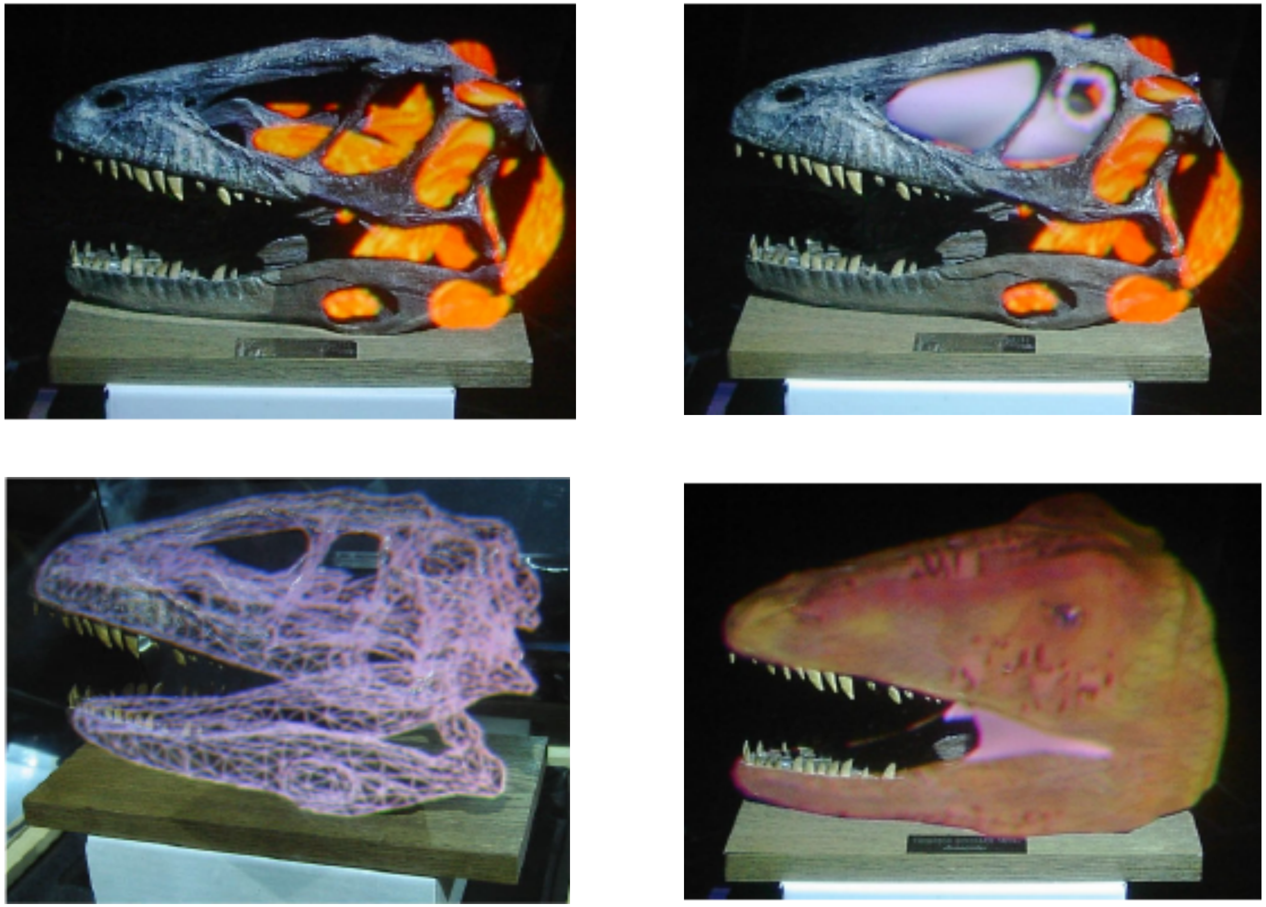


Bimber, O., Fröhlich, B., Schmalstieg, D., & Encarnação, L. M. (2006, July). The virtual showcase. In *ACM SIGGRAPH 2006 Courses* (p. 9). ACM.

Demo: Virtual Showcase

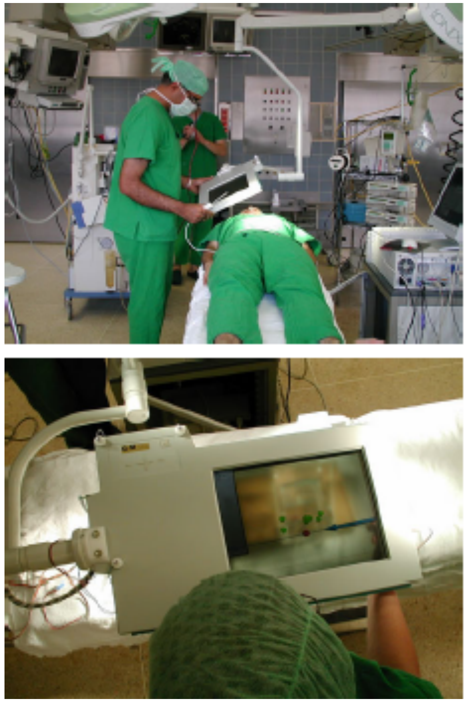
https://www.youtube.com/watch?v=iXl4FolFUzc

Augmented Paleontology



Bimber et. al. IEEE Computer Sept. 2002

Alternate Displays



LCD Panel

Laptop

PDA

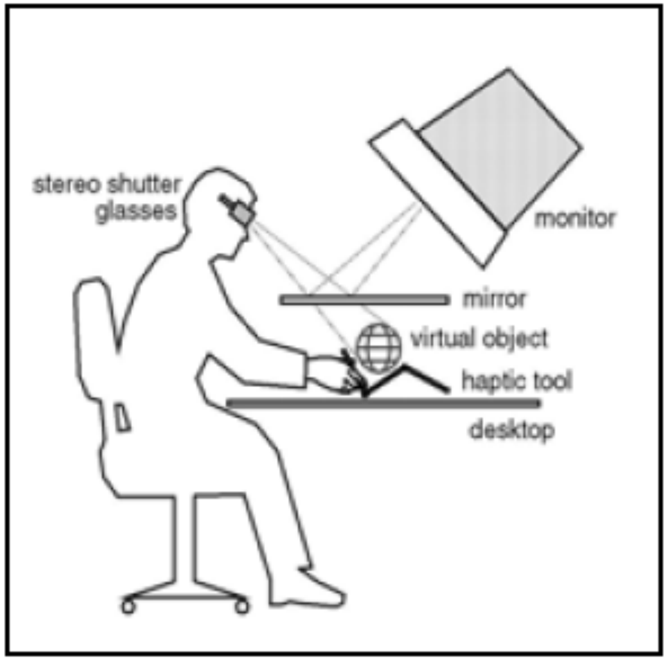
Handheld Displays

- Mobile Phones
  - Camera
  - Display
  - Input

Other Types of AR Display

- Audio
  - spatial sound
  - ambient audio
- Tactile
  - physical sensation
- Haptic
  - virtual touch

Haptic Input



- AR Haptic Workbench
  - CSIRO 2003 – Adcock et. al.

Phantom

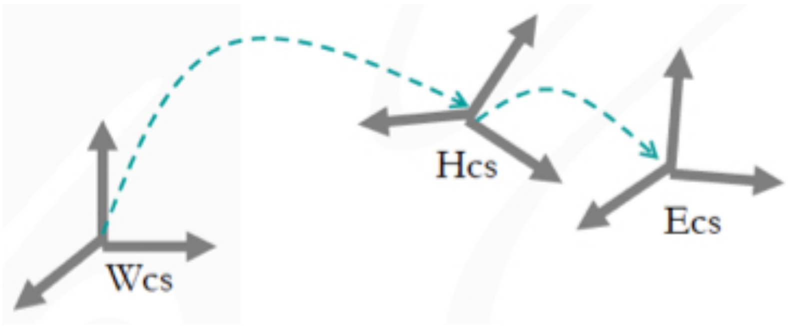
- Sensable Technologies (www.sensable.com)
- 6 DOF Force Feedback Device

Example: Haptic Augmented Reality

- https://www.youtube.com/watch?v=3meAlle8kZs

AR TRACKING AND REGISTRATION

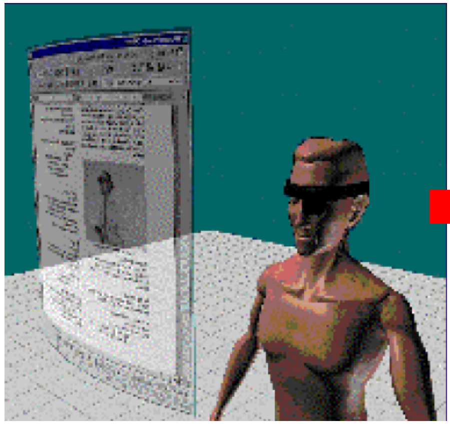
AR Requires Tracking and Registration



- Registration
  - Positioning virtual object wrt real world
    - Fixing virtual object on real object when view is fixed
- Tracking
  - Continually locating the users viewpoint when view moving
    - Position (x,y,z), Orientation (r,p,y)

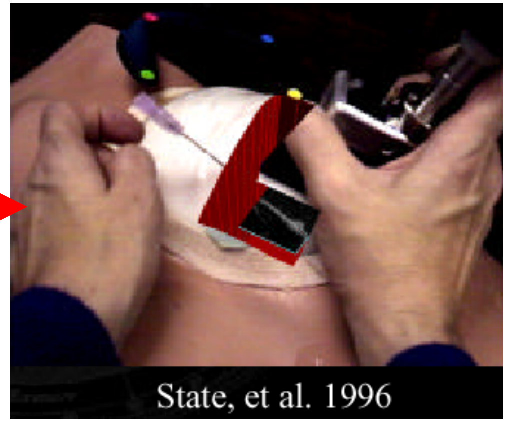
AR TRACKING

Tracking Requirements



Head Stabilized

Body Stabilized



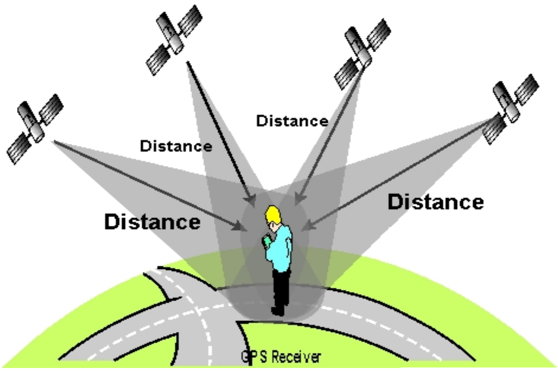
World Stabilized

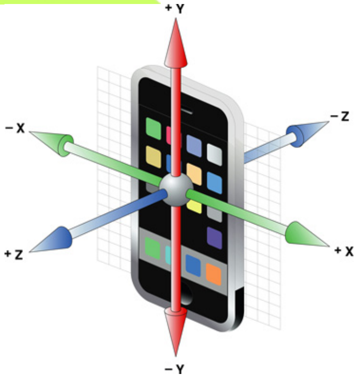
- Augmented Reality Information Display
  - World Stabilized
  - Body Stabilized
  - Head Stabilized

Increasing Tracking Requirements

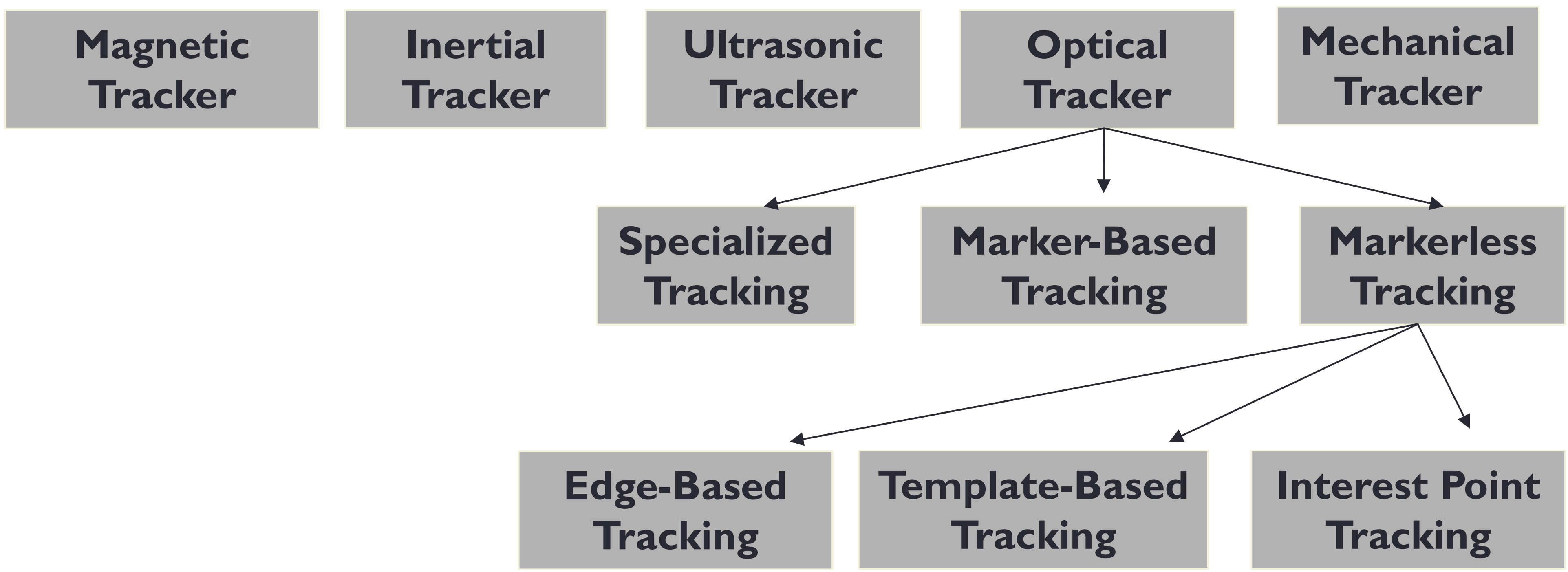
Tracking Technologies

- Active
  - Mechanical, Magnetic, Ultrasonic
  - GPS, Wifi, cell location
- Passive
  - Inertial sensors (compass, accelerometer, gyro)
  - Computer Vision
    - Marker based, Natural feature tracking
- Hybrid Tracking
  - Combined sensors (eg Vision + Inertial)





Tracking Types



Mechanical Tracker

- Idea: mechanical arms with joint sensors

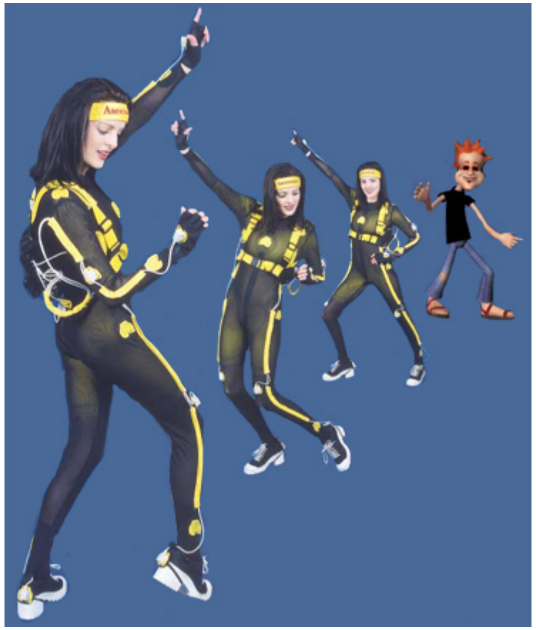
Microscribe

- ++: high accuracy, haptic feedback
- -- : cumbersome, expensive

Magnetic Tracker

- Idea: coil generates current when moved in magnetic field. Measuring current gives position and orientation relative to magnetic source.

Flock of Birds (Ascension)

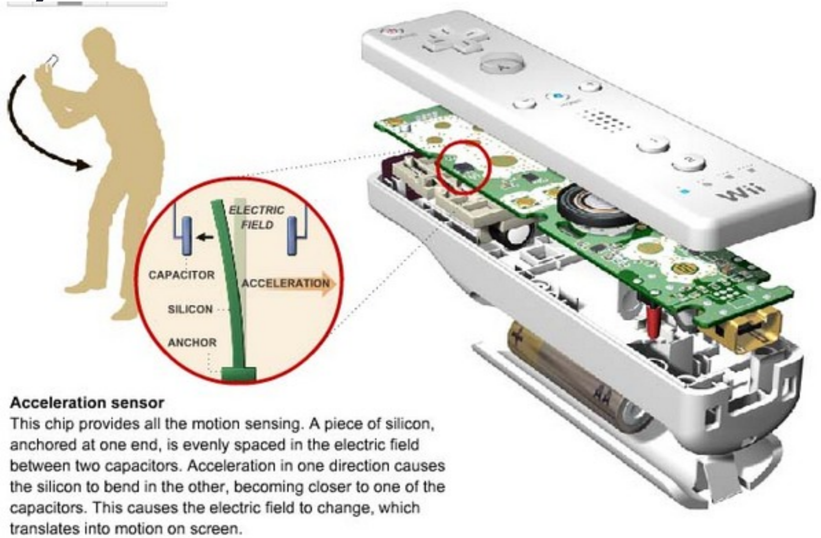


- ++: 6DOF, robust
- -- : wired, sensible to metal, noisy, expensive

Inertial Tracker

- Idea: measuring linear and angular orientation rates (accelerometer/gyroscope)

IS300 (Intersense)



Wii Remote

- ++: no transmitter, cheap, small, high frequency, wireless
- -- : drifts over time, hysteresis effect, only 3DOF

Ultrasonic Tracker

- Idea: time of Flight or phase-Coherence Sound Waves

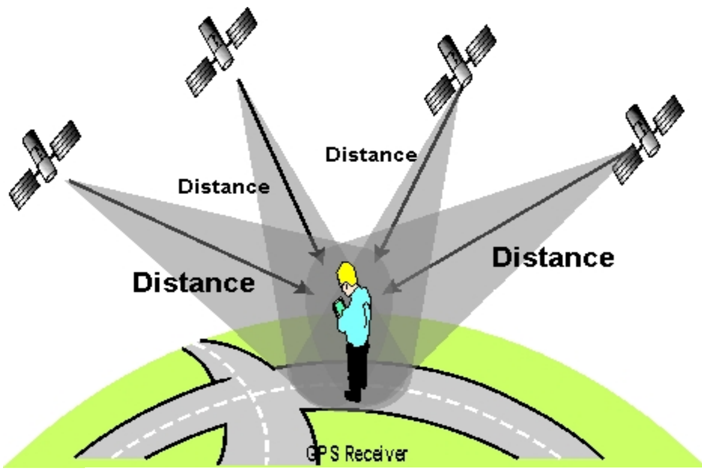
Ultrasonic Logitech

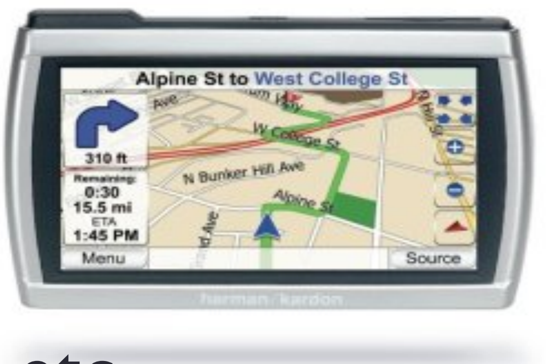
IS600

- ++: Small, Cheap
- -- : 3DOF, Line of Sight, Low resolution, Affected by environmental conditons (pressure, temperature)

Global Positioning System (GPS)

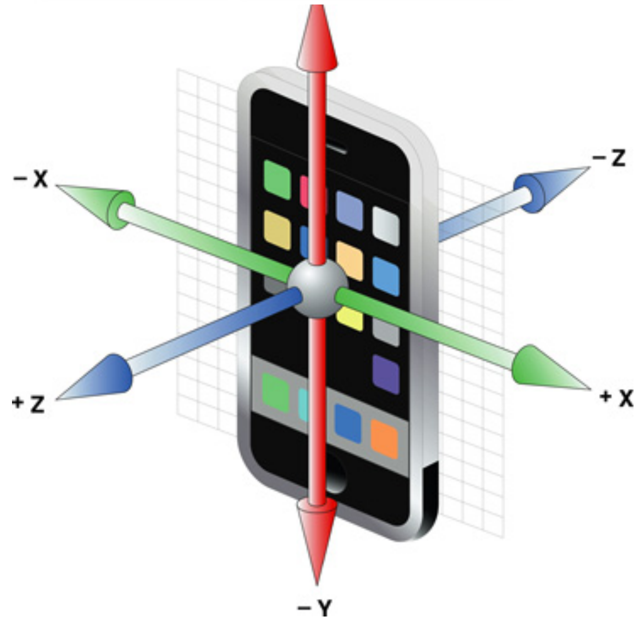
- Created by US in 1978
  - Currently 29 satellites
- Satellites send position + time
- GPS Receiver positioning
  - 4 satellites need to be visible
  - Differential time of arrival
  - Triangulation
- Accuracy
  - 5-30m+, blocked by weather, buildings etc.





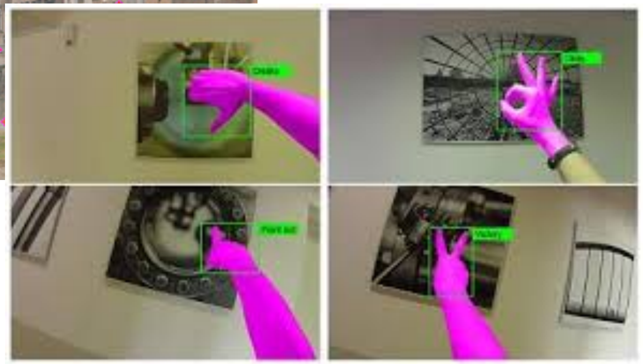
Mobile Sensors

- Inertial compass
  - Earth's magnetic field
  - Measures absolute orientation
- Accelerometers
  - Measures acceleration about axis
  - Used for tilt, relative rotation
  - Can drift over time



OPTICAL TRACKING

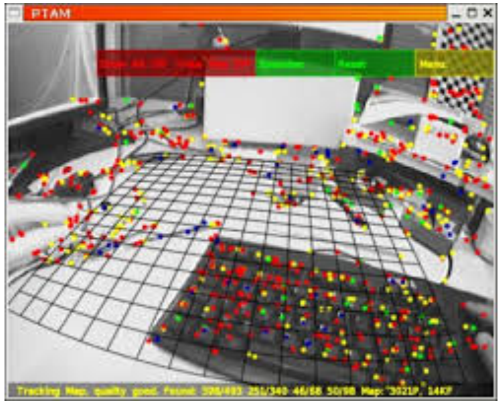
Why Optical Tracking for AR?



- Many AR devices have cameras
  - Mobile phone/tablet, Video see-through display
- Provides precise alignment between video and AR overlay
  - Using features in video to generate pixel perfect alignment
  - Real world has many visual features that can be tracked from
- Computer Vision well established discipline
  - Over 40 years of research to draw on
  - Old non real time algorithms can be run in real time on todays devices

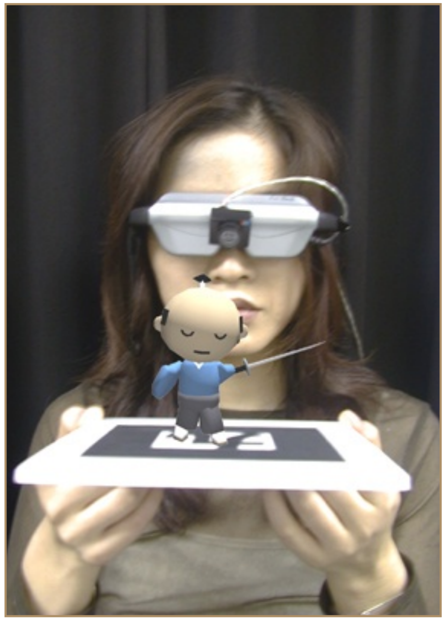
Common AR Optical Tracking Types

- Marker Tracking
  - Tracking known artificial markers/images
    - e.g. ARToolKit square markers
- Markerless Tracking
  - Tracking from known features in real world
    - e.g. Vuforia image tracking
- Unprepared Tracking
  - Tracking in unknown environment
    - e.g. SLAM tracking

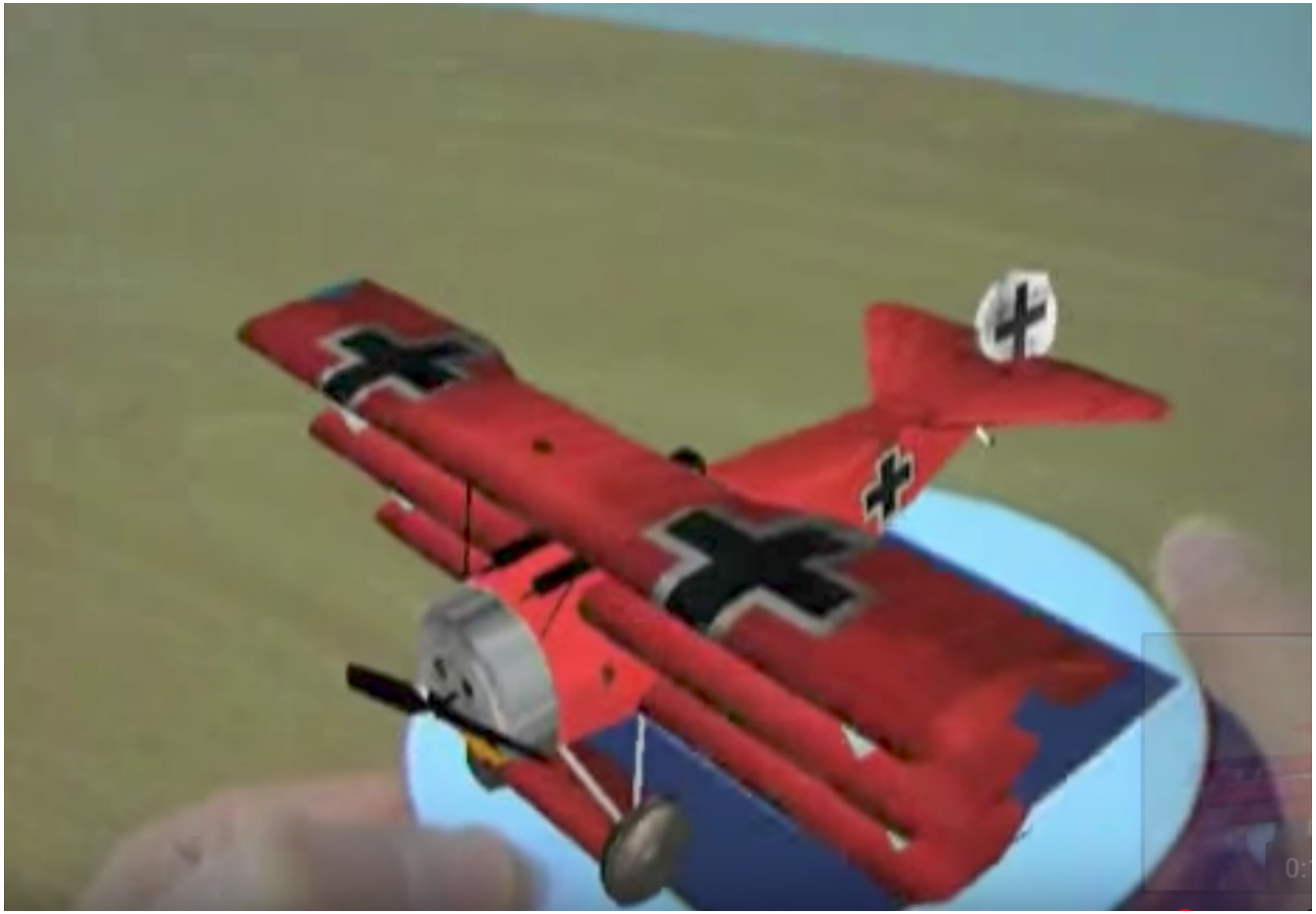


Marker tracking

- Available for more than 10 years
- Several open source solutions exist
  - ARToolKit, ARTag, ATK+, etc
- Fairly simple to implement
  - Standard computer vision methods
- A rectangle provides 4 corner points
  - Enough for pose estimation!

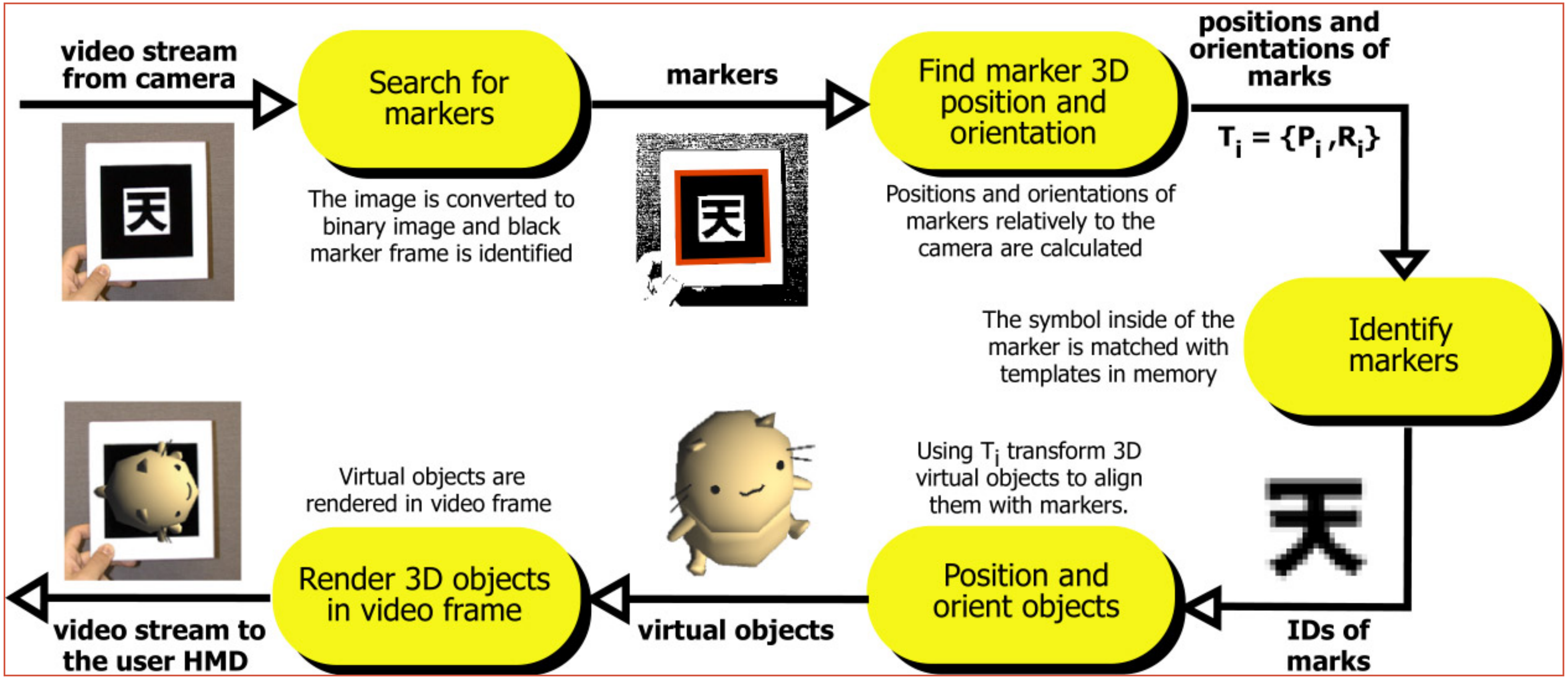


Demo: ARToolKit



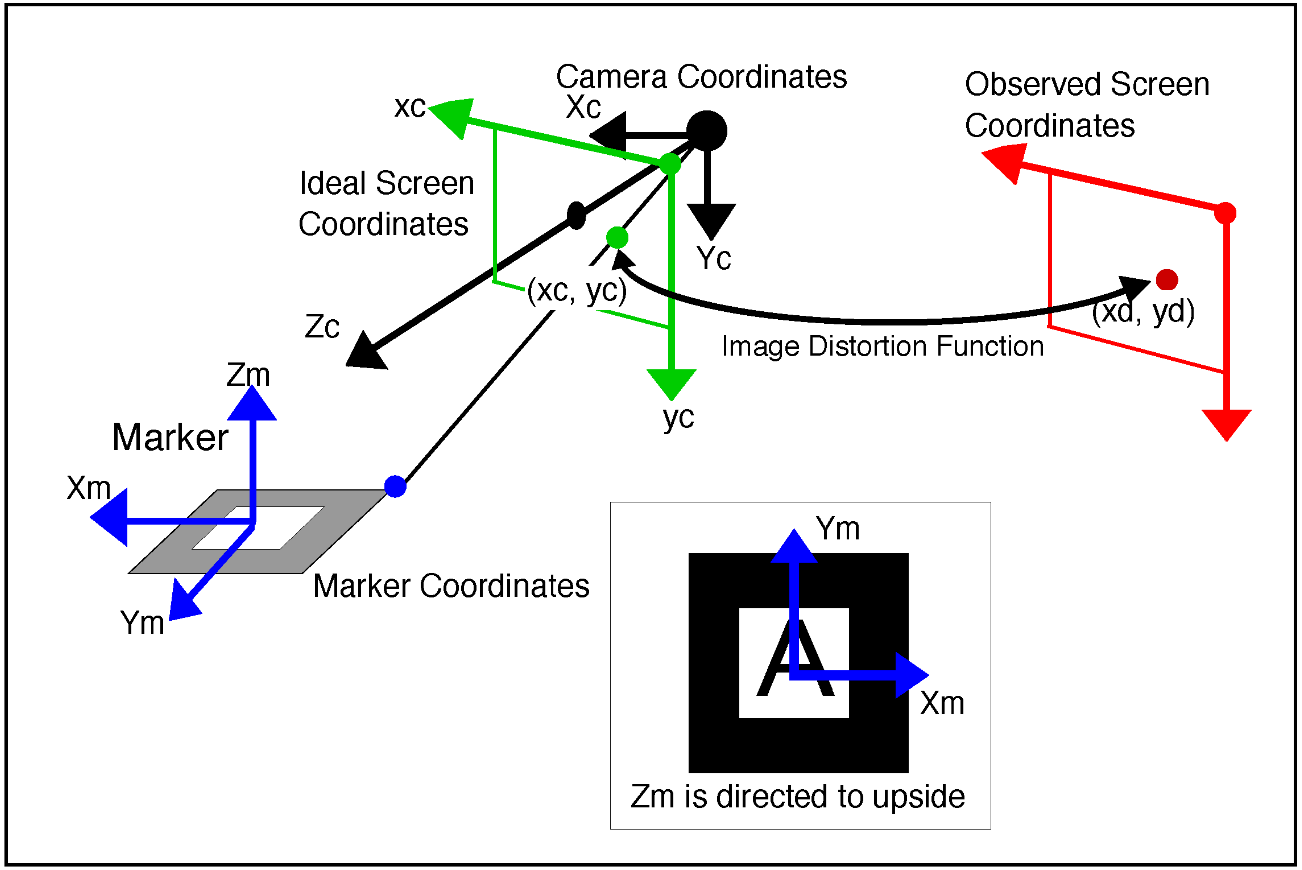
- https://www.youtube.com/watch?v=TqGAqAFlGg0

Marker Based Tracking: ARToolKit



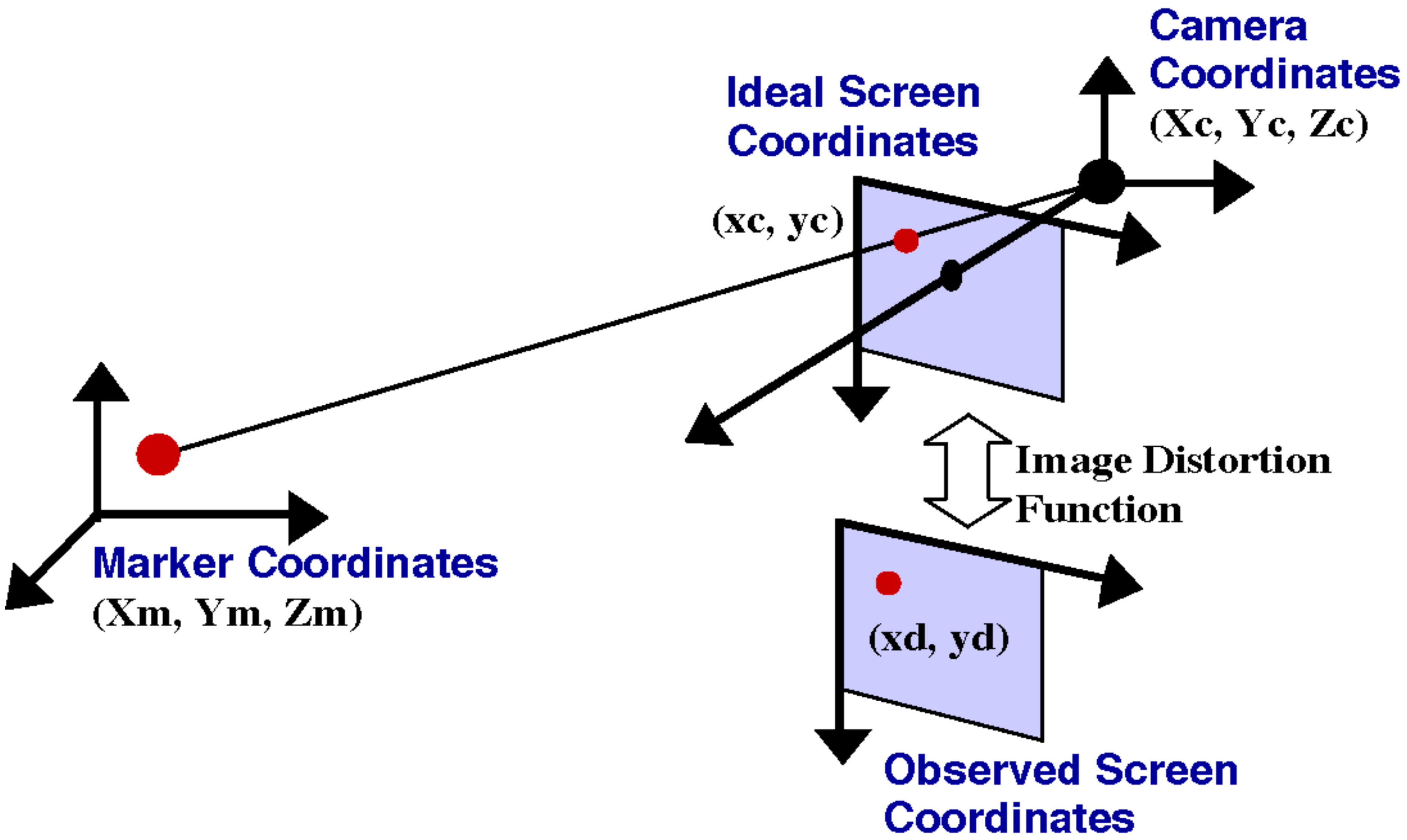
http://www.artoolkit.org

Goal: Find Camera Pose



- Goal is to find the camera pose in maker coordinate frame
- Knowing:
  - Position of key points in on-screen video image
  - Camera properties (focal length, image distortion)

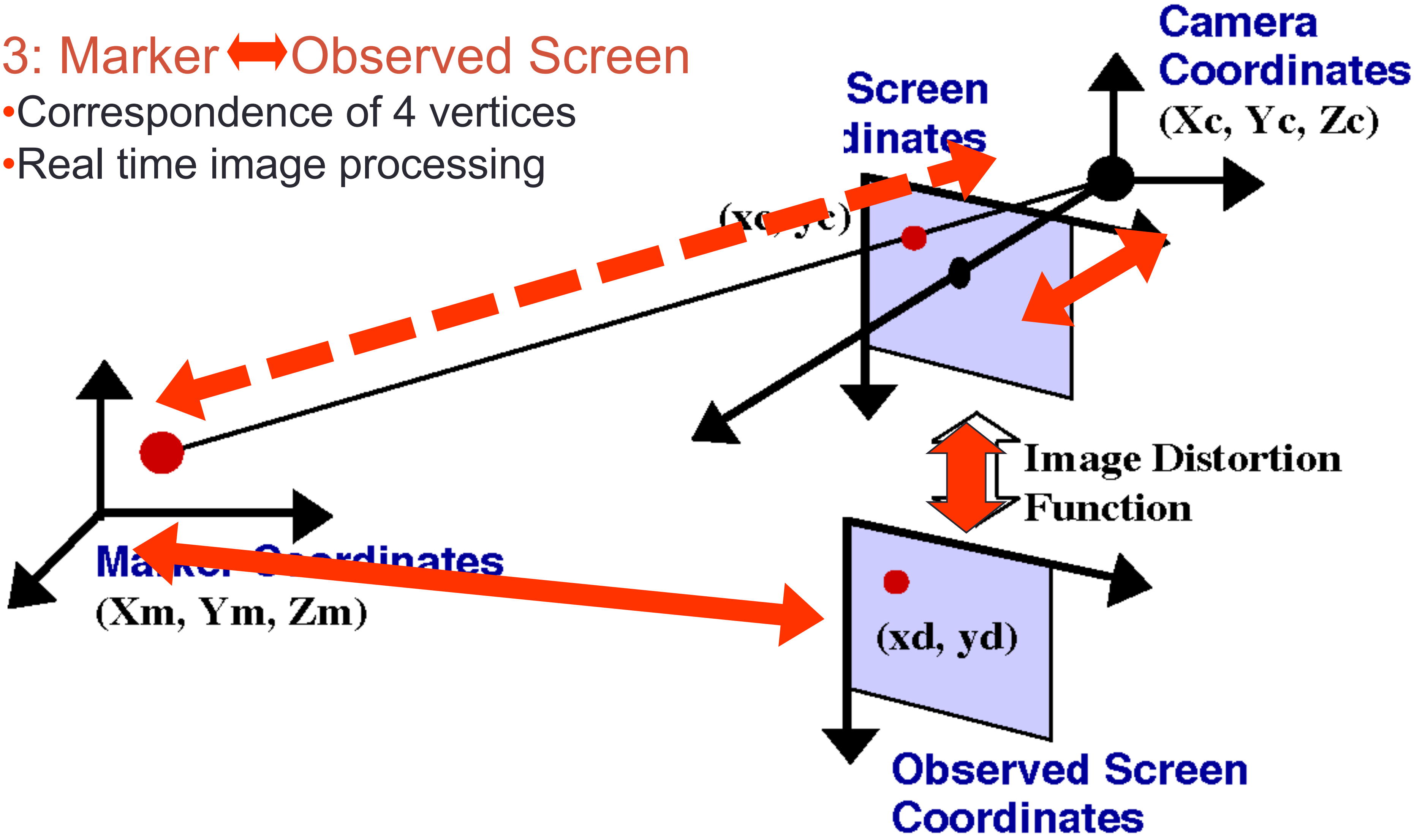
Coordinates for Marker Tracking



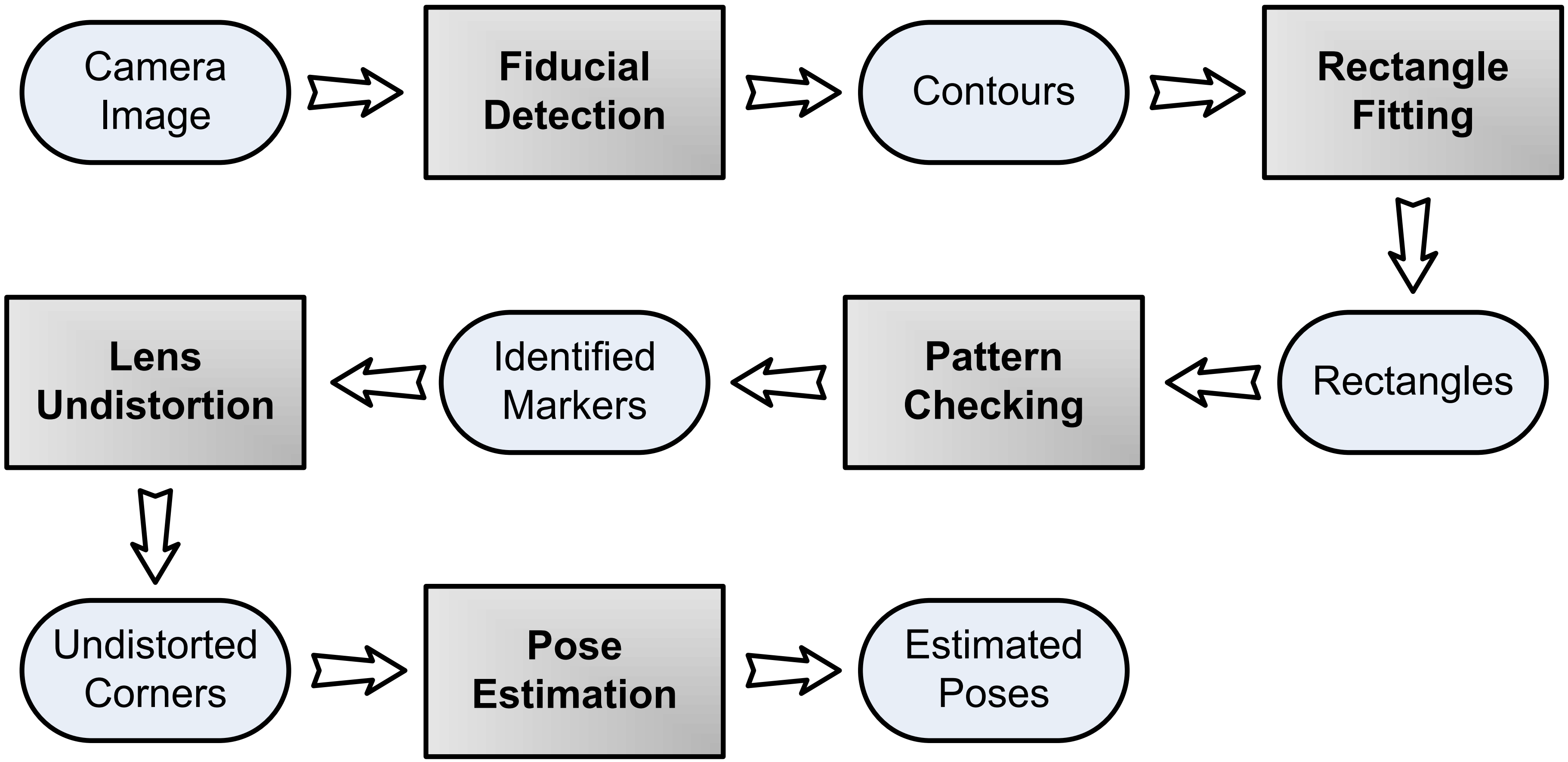
Coordinates for Marker Tracking

3: Marker ⟷ Observed Screen

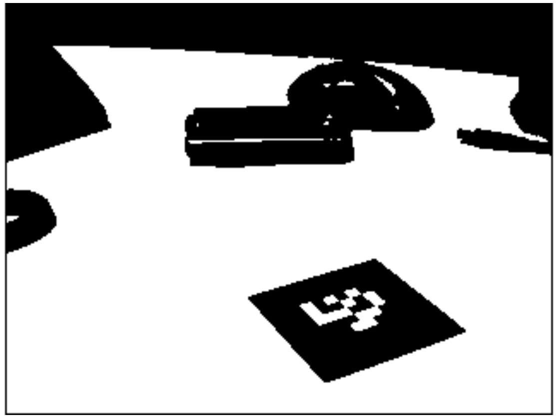
- Correspondence of 4 vertices
- Real time image processing

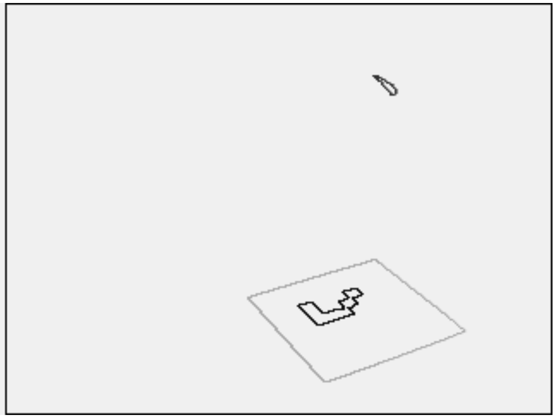


Marker Tracking – Overview



Marker Tracking – Fiducial Detection

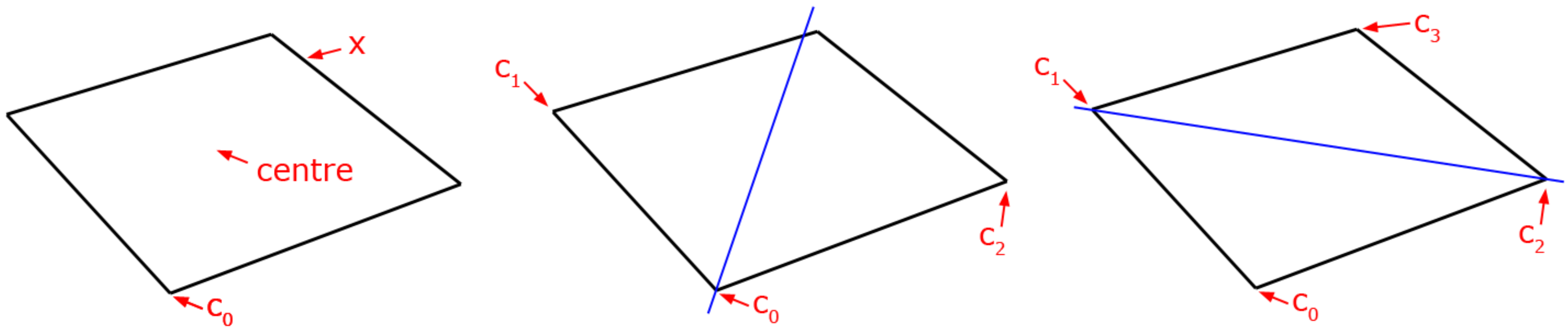




- Threshold the whole image to black and white
- Search scanline by scanline for edges (white to black)
- Follow edge until either
  - Back to starting pixel
  - Image border
- Check for size
  - Reject fiducials early that are too small (or too large)

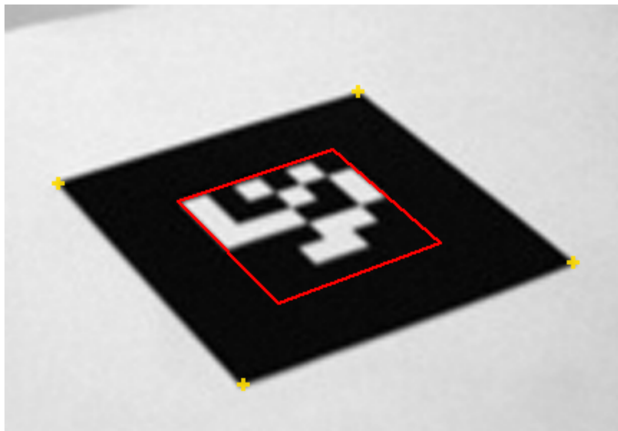
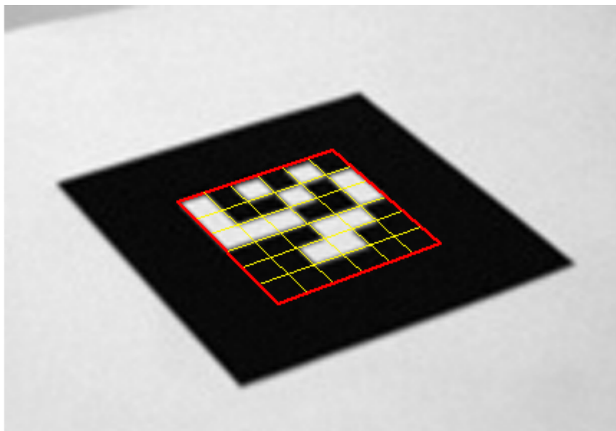
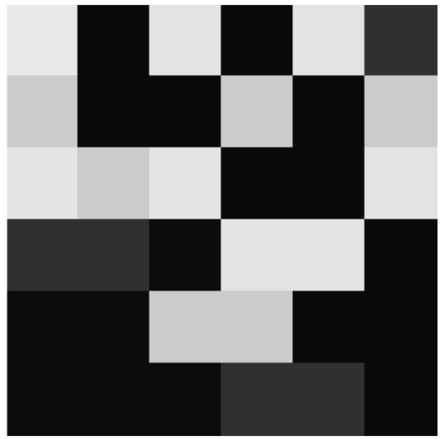
Marker Tracking – Rectangle Fitting

- Start with an arbitrary point "x" on the contour
- The point with maximum distance must be a corner c0
- Create a diagonal through the center
- Find points c1 & c2 with maximum distance left and right of diag.
- New diagonal from c1 to c2
- Find point c3 right of diagonal with maximum distance



Marker Tracking – Pattern checking

- Calculate homography using the 4 corner points
  - "Direct Linear Transform" algorithm
  - Maps normalized coordinates to marker coordinates (simple perspective projection, no camera model)
- Extract pattern by sampling and check
  - Id (implicit encoding)
  - Template (normalized cross correlation)

Marker Tracking – Corner refinement

- Refine corner coordinates
  - Critical for high quality tracking
  - Remember: 4 points is the bare minimum!
  - So these 4 points should better be accurate…
- Detect sub-pixel coordinates
  - E.g. Harris corner detector
    - Specialized methods can be faster and more accurate
  - Strongly reduces jitter!
- Undistort corner coordinates
  - Remove radial distortion from lens

Marker tracking – Pose estimation

- Calculates marker pose relative to the camera
- Initial estimation directly from homography
  - Very fast, but coarse with error
  - Jitters a lot…
- Iterative Refinement using Gauss-Newton method
  - 6 parameters (3 for position, 3 for rotation) to refine
  - At each iteration we optimize on the error
  - Iterat

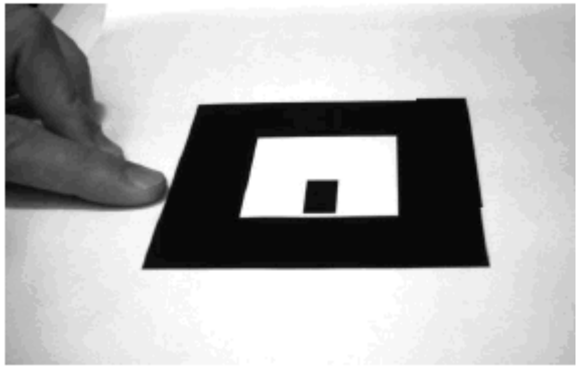
From Marker To Camera

- Rotation & Translation

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \mathbf{T}_{\mathbf{CM}} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

$\mathbf{T}_{\mathbf{CM}}$: 4x4 transformation matrix
from marker coord. to camera coord.

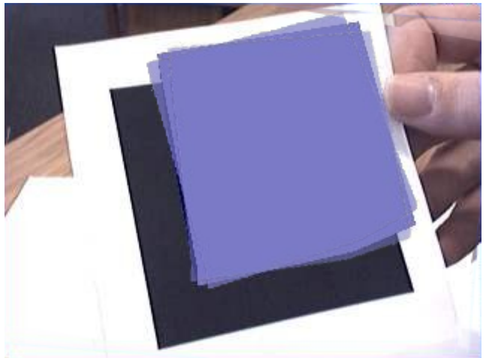
Tracking challenges in ARToolKit



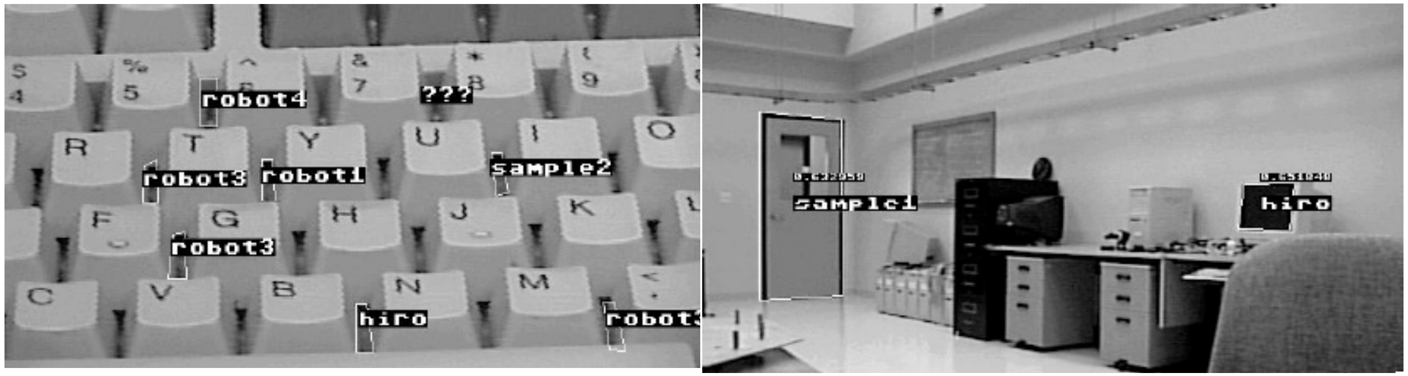
Occlusion
(image by M. Fiala)

Unfocused camera, motion blur

Dark/unevenly lit scene, vignetting



Jittering
(Photoshop illustration)



False positives and inter-marker confusion
(image by M. Fiala)

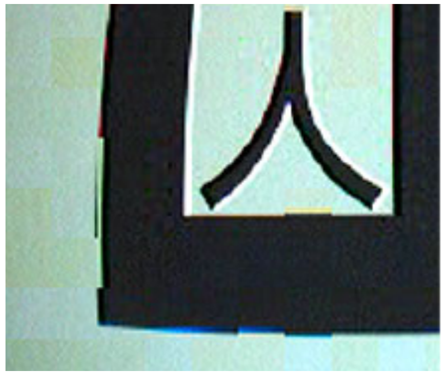
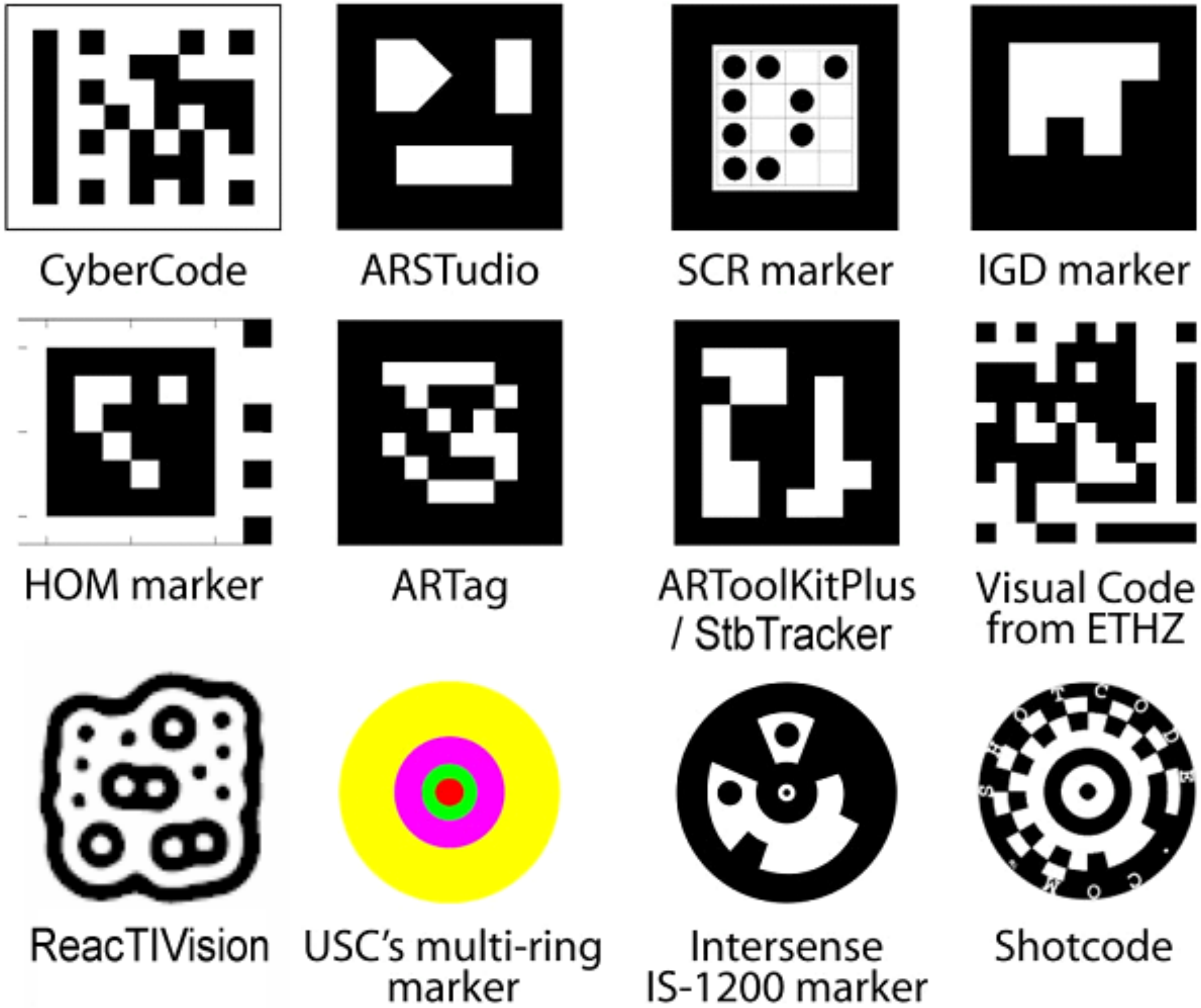
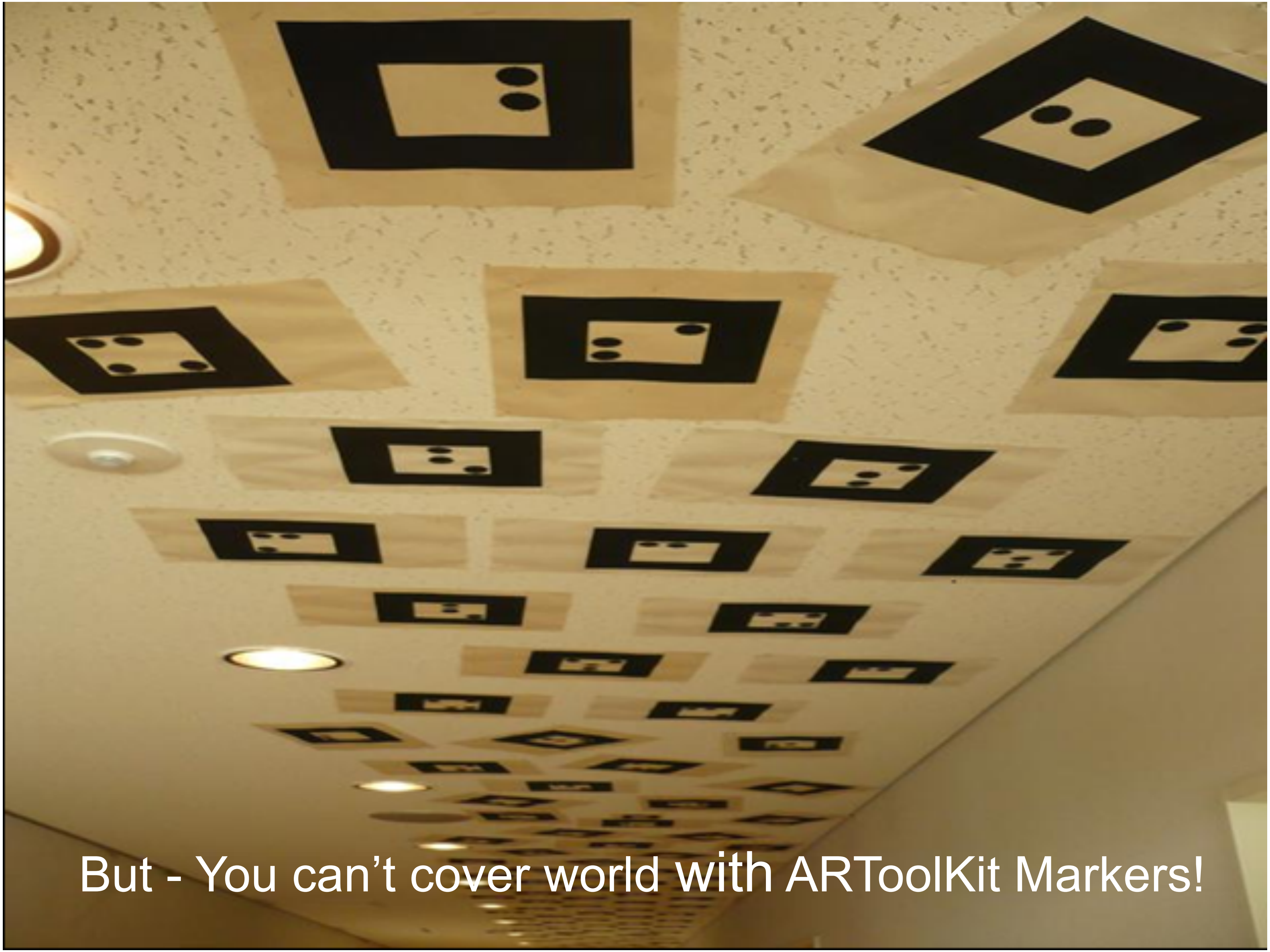


Image noise
(e.g. poor lens, block coding /

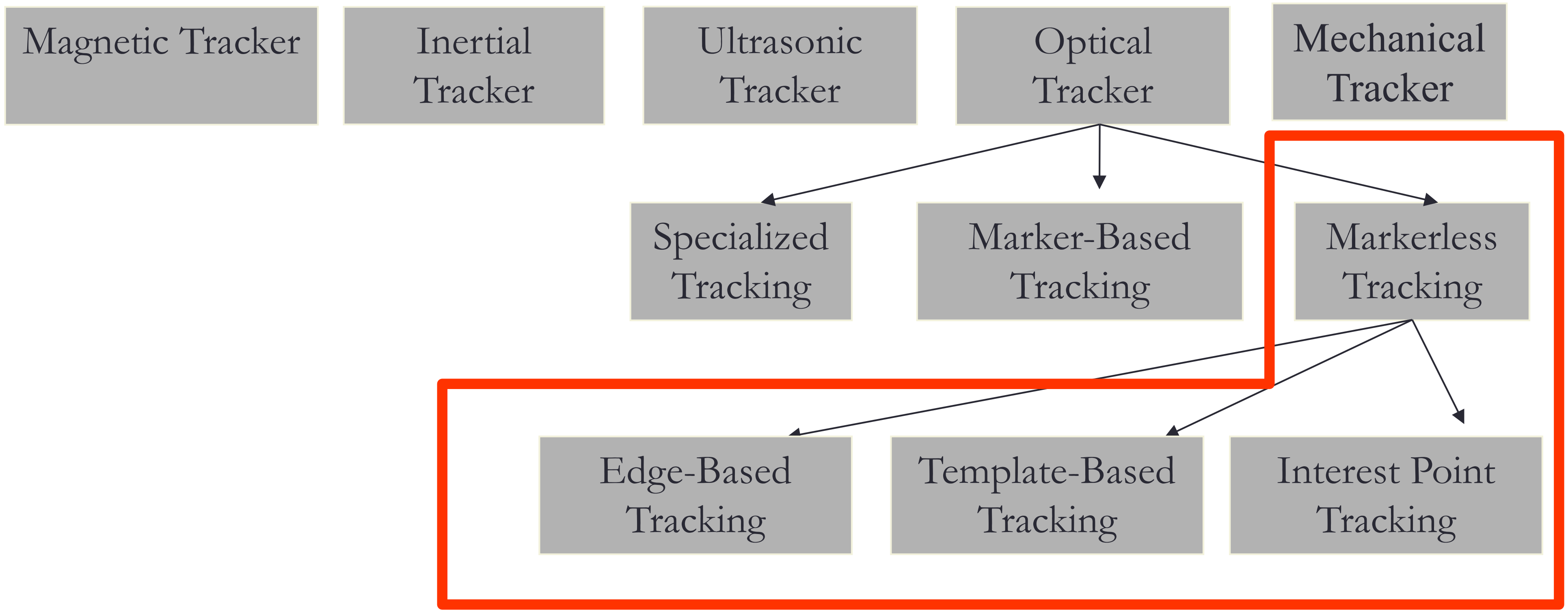
Other Marker Tracking Libraries




But - You can’t cover world with ARToolKit Markers!

Markerless Tracking

- No more Markers! ➔Markerless Tracking



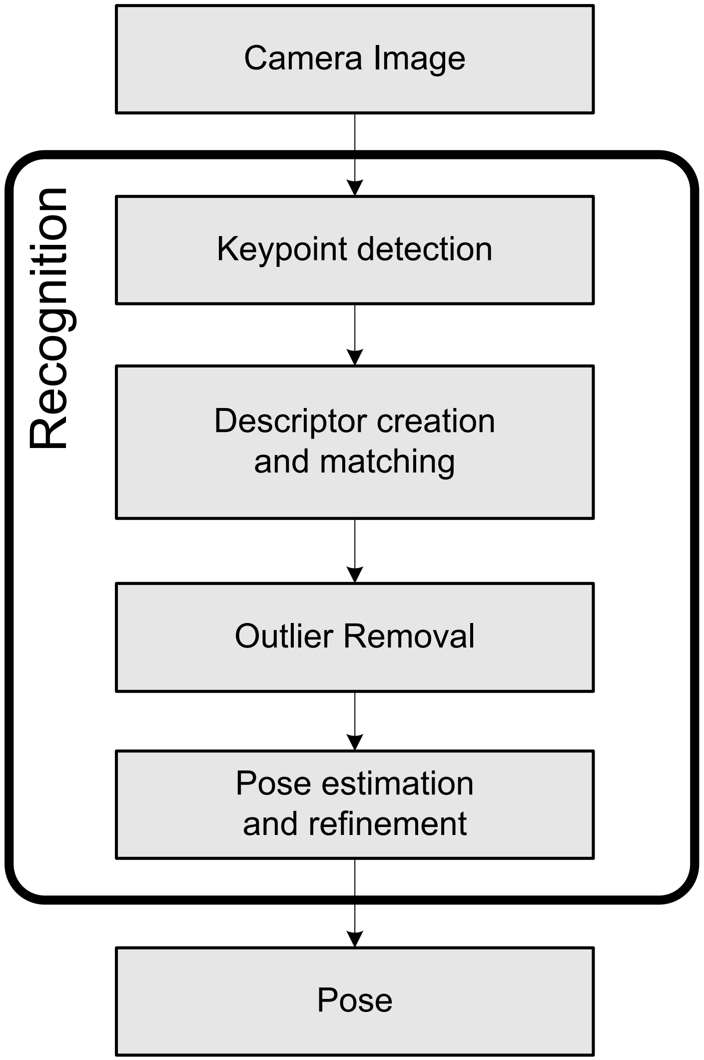
Natural Feature Tracking

- Use Natural Cues of Real Elements
  - Edges
  - Surface Texture
  - Interest Points
- Model or Model-Free
- No visual pollution

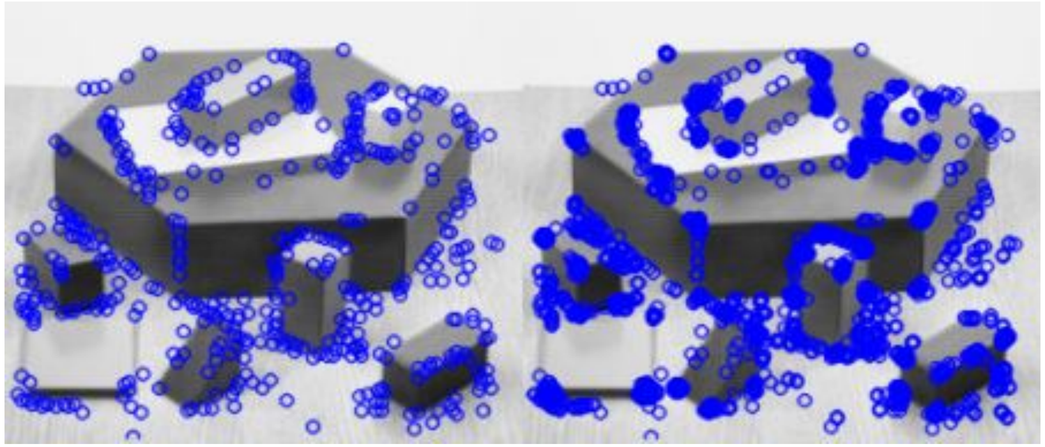
Texture Tracking

Tracking by Keypoint Detection

- This is what most trackers do…
- Targets are detected every frame
- Popular because
  tracking and detection
  are solved simultaneously



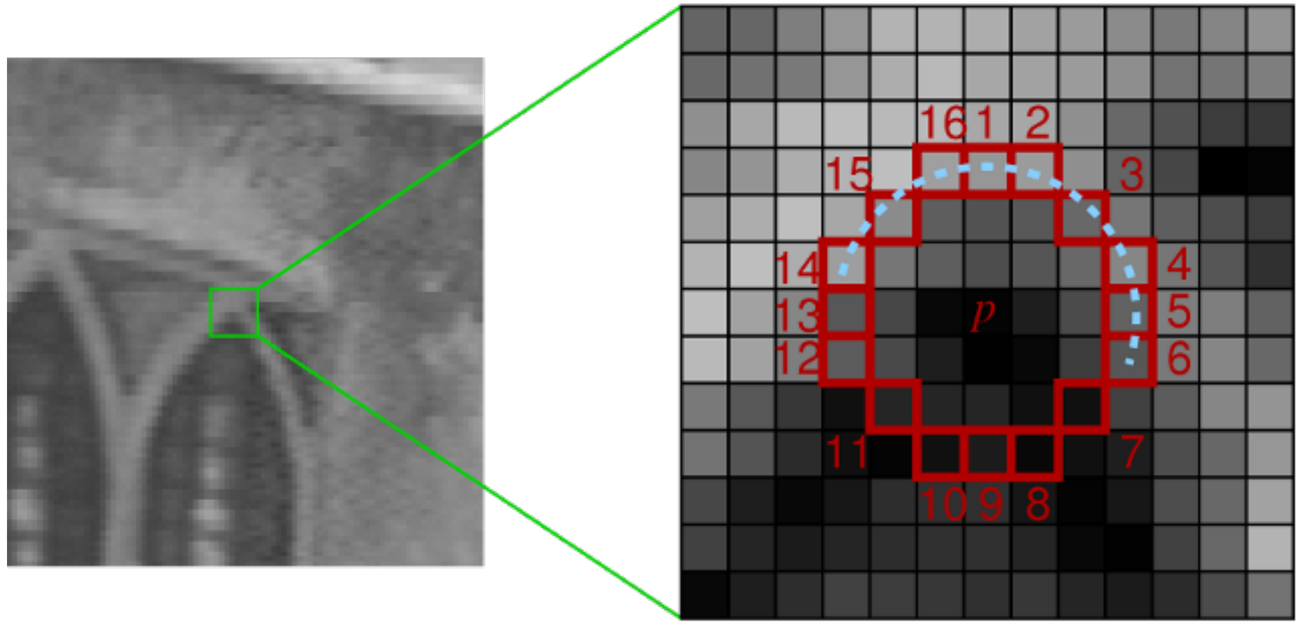
What is a Keypoint?

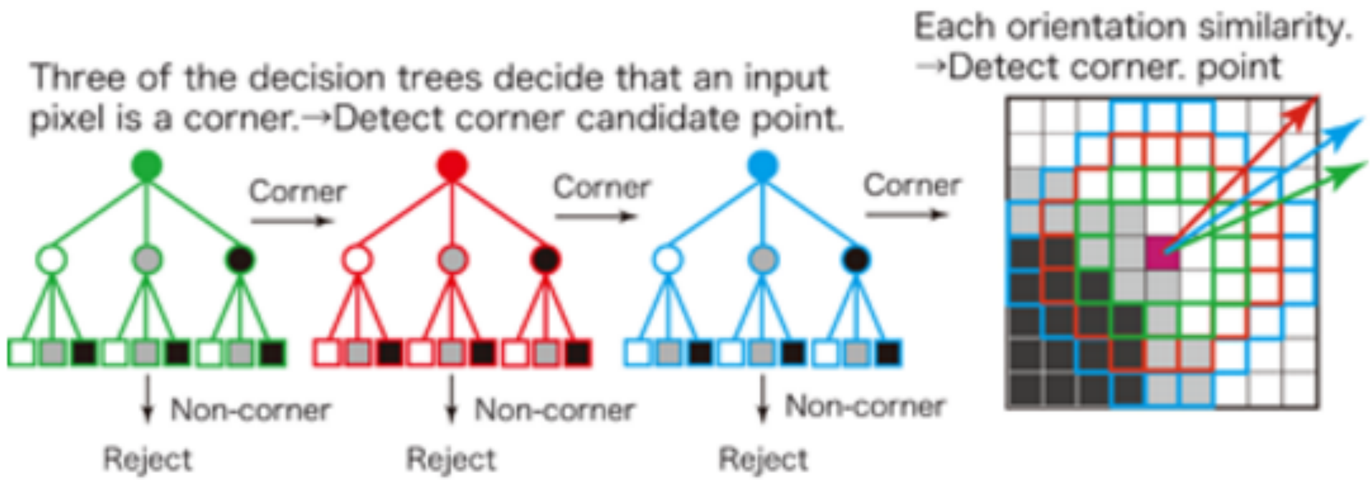


- It depends on the detector you use!
- For high performance use the FAST corner detector
  - Apply FAST to all pixels of your image
  - Obtain a set of keypoints for your image
  - *Describe* the keypoints

Rosten, E., & Drummond, T. (2006, May). Machine learning for high-speed corner detection. In *European conference on computer vision* (pp. 430-443). Springer Berlin Heidelberg.

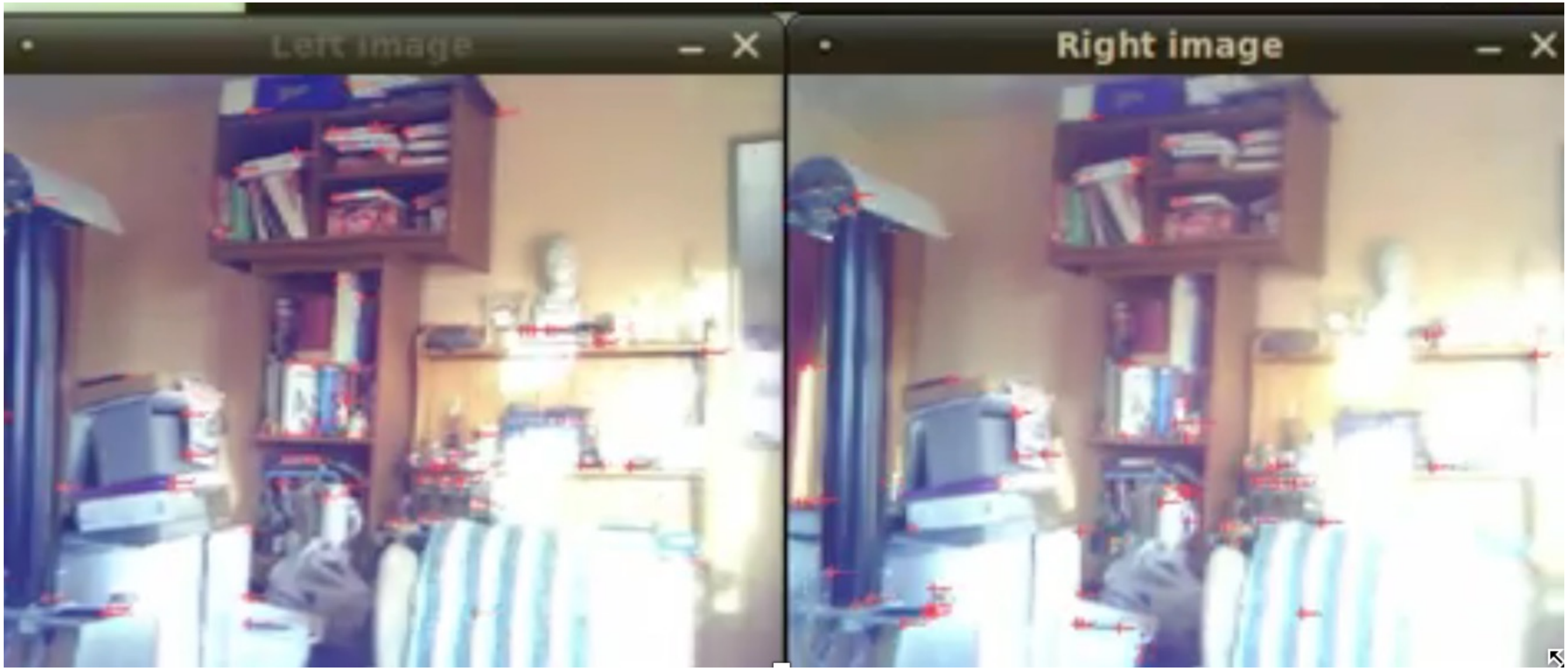
FAST Corner Keypoint Detection





Keypoint detection flow in Cascaded FAST.

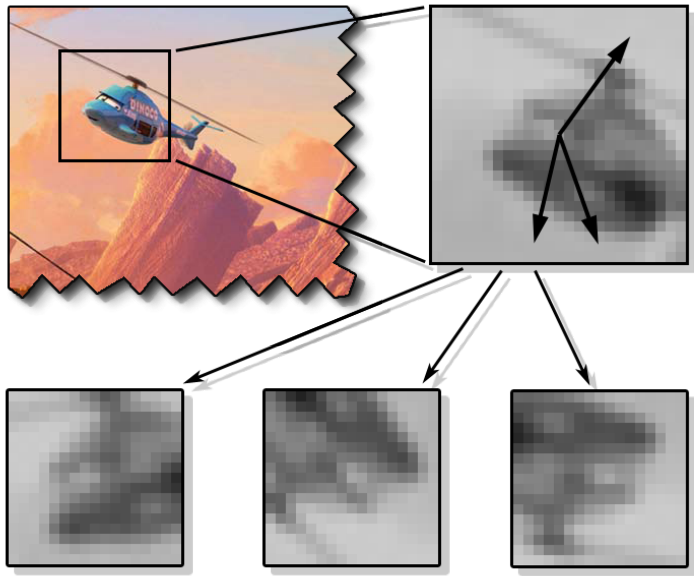
Example: FAST Corner Detection



https://www.youtube.com/watch?v=pJ2xSrIXy_s

Descriptors

- Describe the Keypoint features
- Can use SIFT
  - Estimate the dominant keypoint orientation using gradients
  - Compensate for detected orientation
  - Describe the keypoints in terms of the gradients surrounding it



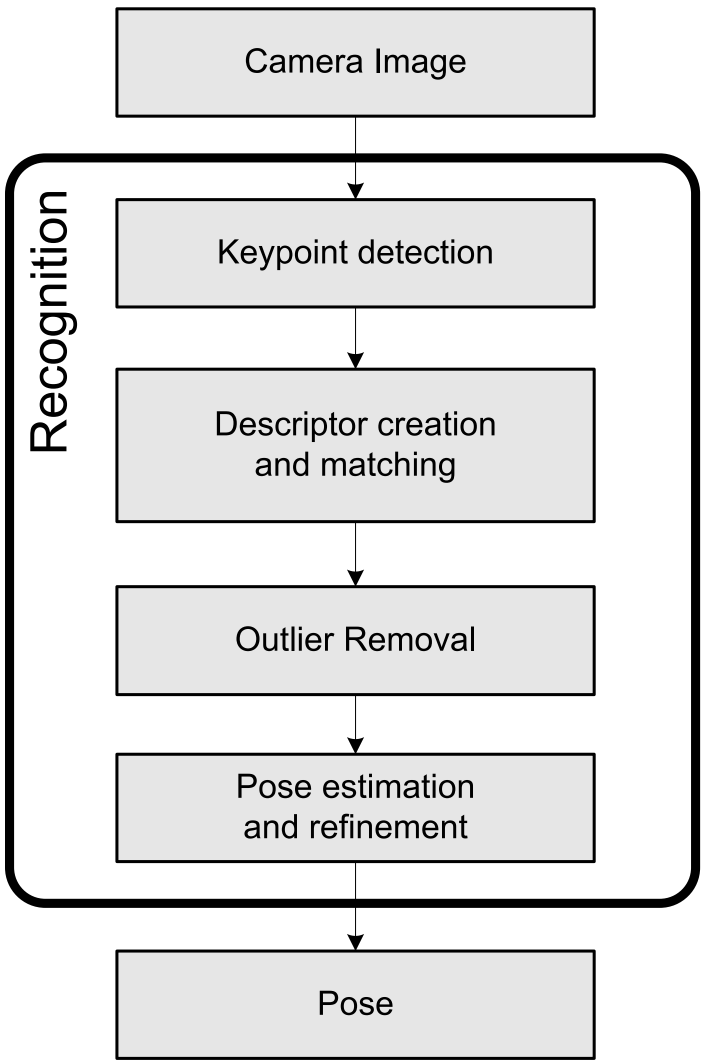
Wagner D., Reitmayr G., Mulloni A., Drummond T., Schmalstieg D.,
Real-Time Detection and Tracking for Augmented Reality on Mobile Phones.
IEEE Transactions on Visualization and Computer Graphics, May/June, 2010

Database Creation

- Offline step – create database of known features
- Searching for corners in a static image
- For robustness look at corners on multiple scales
  - Some corners are more descriptive at larger or smaller scales
  - We don't know how far users will be from our image
- Build a database file with all descriptors and their position on the original image

Real-time tracking

- Search for known keypoints in the video image
- Create the descriptors
- Match the descriptors from the live video against those in the database
  - Brute force is not an option
  - Need the speed-up of special data structures
    - E.g., we use multiple spill trees



NFT – Outlier removal

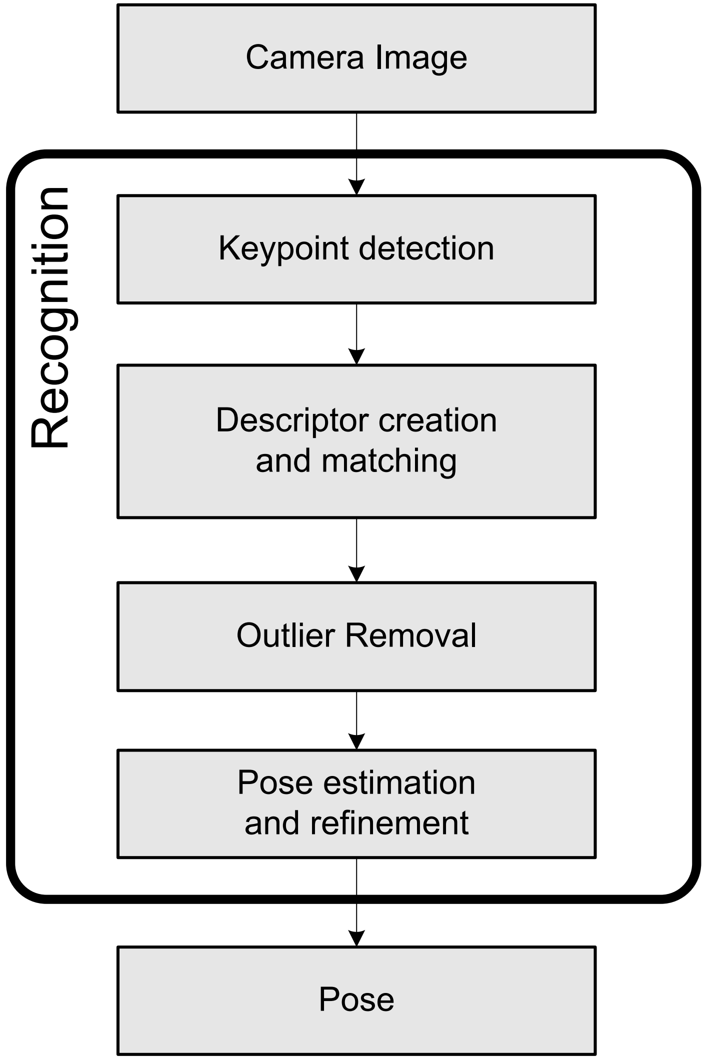
- Removing outlining features
- Cascade of removal techniques
- Start with cheapest, finish with most expensive…
  - First simple geometric tests
    - E.g., line tests
      - Select 2 points to form a line
      - Check all other points being on correct side of line
  - Then, homography-based tests

NFT – Pose refinement

- Pose from homography makes good starting point
- Based on Gauss-Newton iteration
  - Try to minimize the re-projection error of the keypoints
- Part of tracking pipeline that mostly benefits from floating point usage
- Can still be implemented effectively in fixed point
- Typically 2-4 iterations are enough…

NFT – Real-time tracking

- Search for keypoints in the video image
- Create the descriptors
- Match the descriptors from the live video against those in the database
- Remove the keypoints that are outliers
- Use the remaining keypoints to calculate the pose of the camera

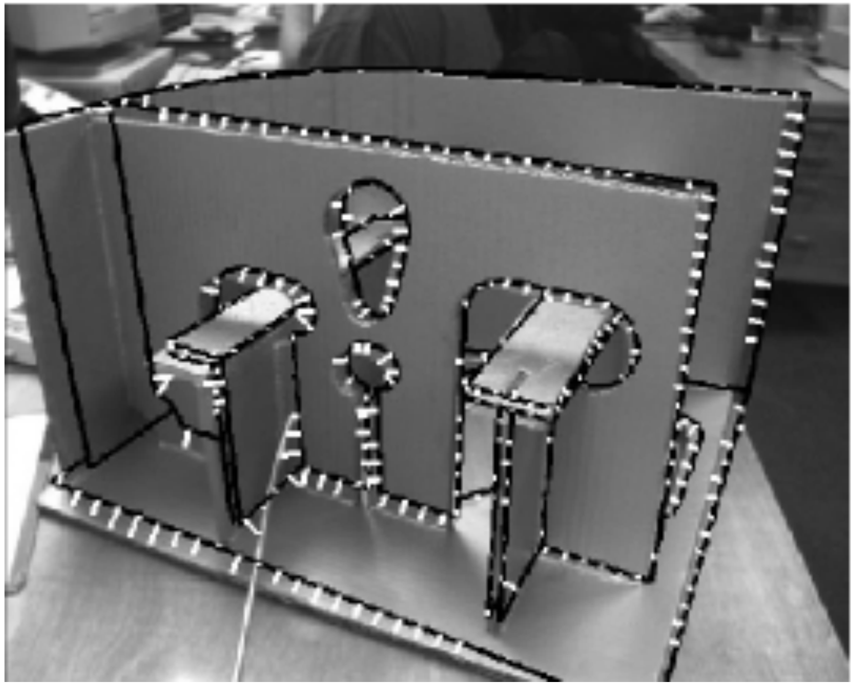


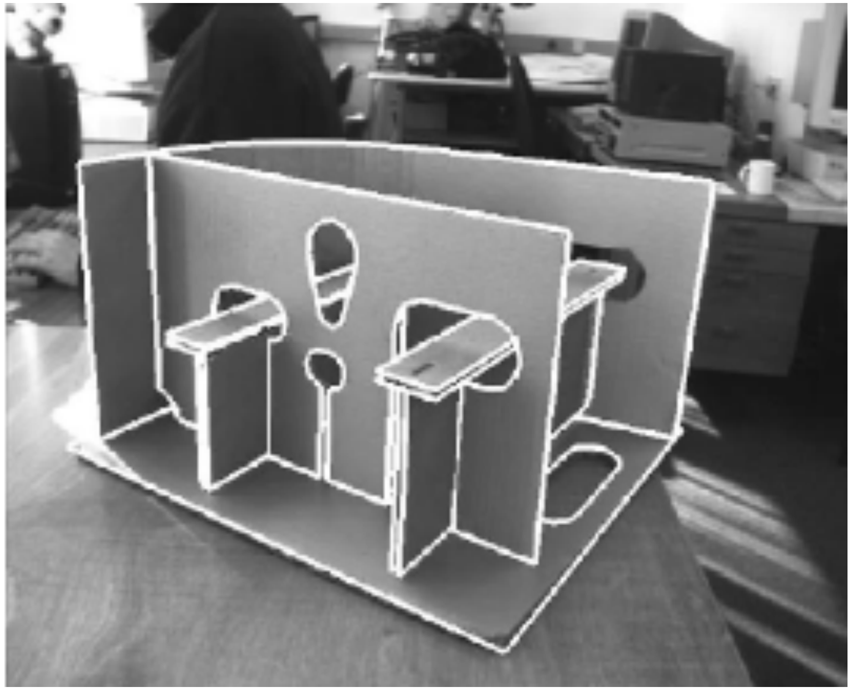
Demo: Vuforia Texture Tracking

https://www.youtube.com/watch?v=1Qf5Qew5zSU

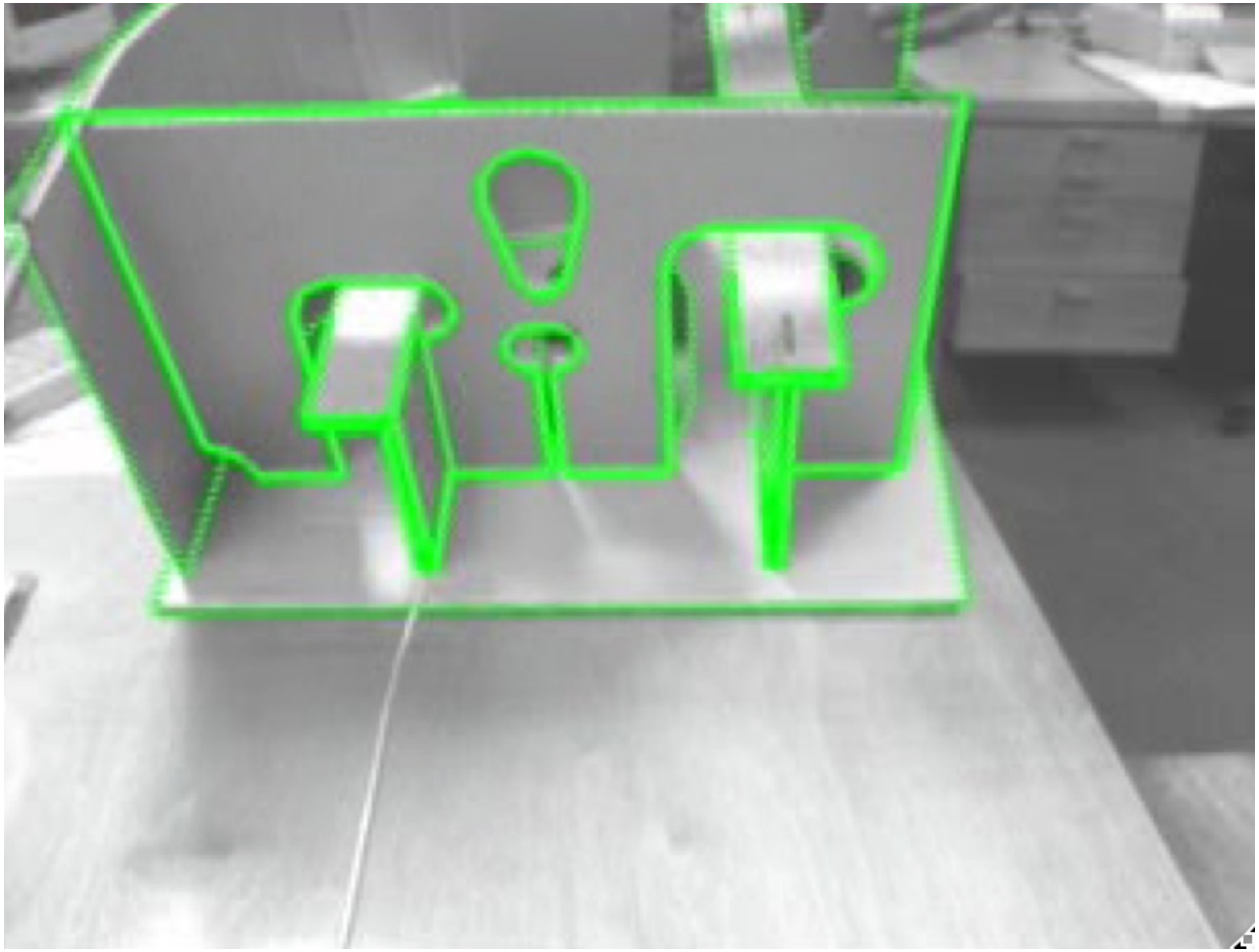
Edge Based Tracking

- Example: RAPiD [Drummond et al. 02]
  - Initialization, Control Points, Pose Prediction (Global Method)





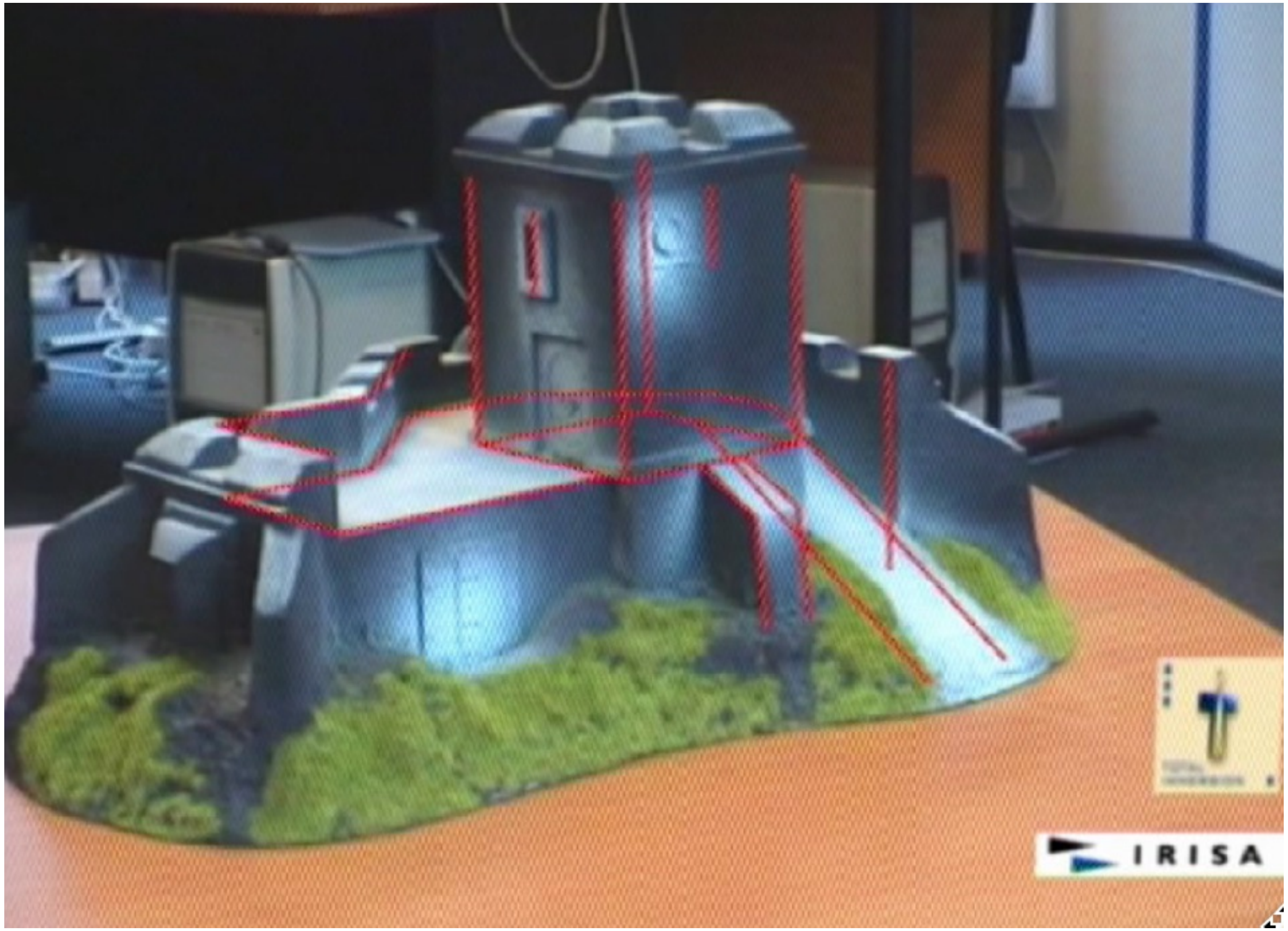
Demo: Edge Based Tracking



https://www.youtube.com/watch?v=Z1J0VLizVDs

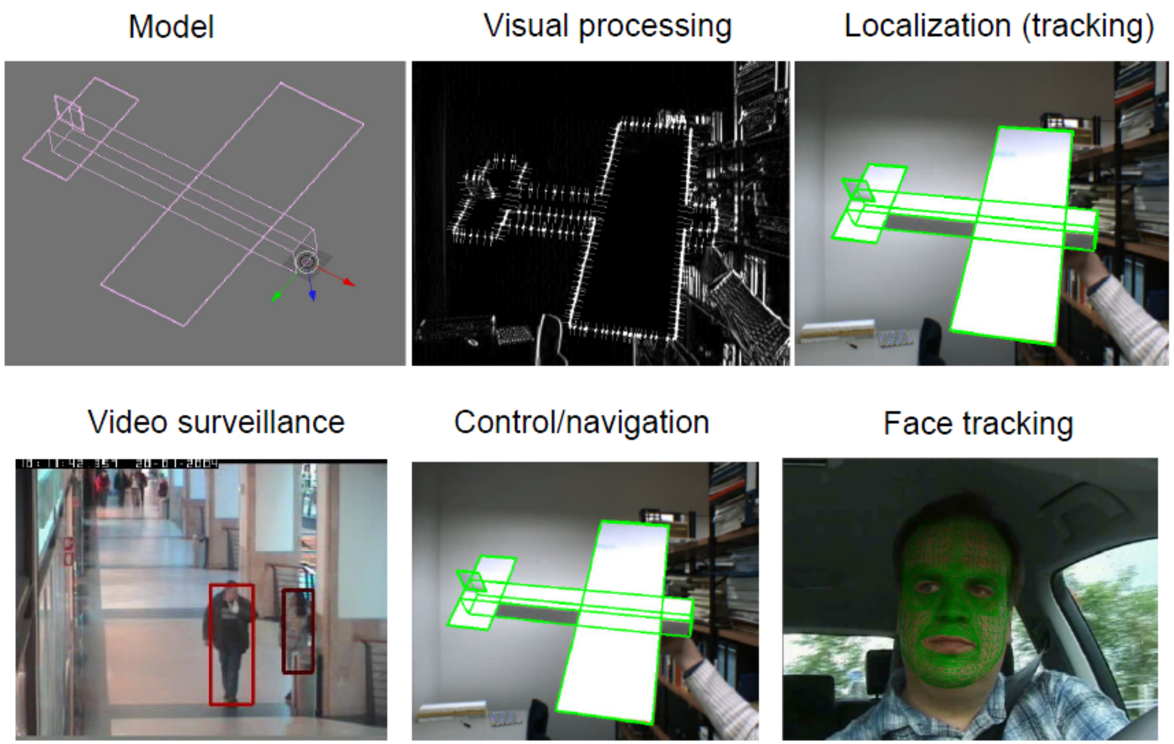
Line Based Tracking

- Visual Servoing [Comport et al. 2004]

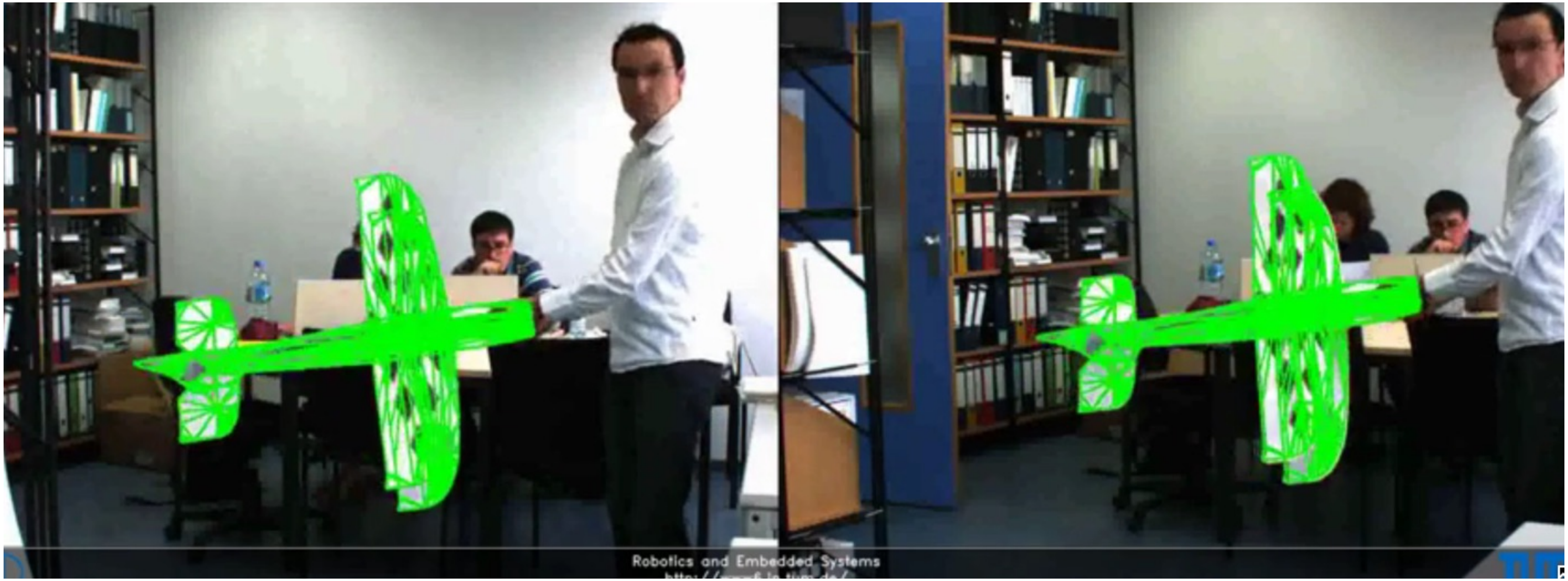


Model Based Tracking

- Tracking from 3D object shape
- Example: OpenTL - www.opentl.org
  - General purpose library for model based visual tracking

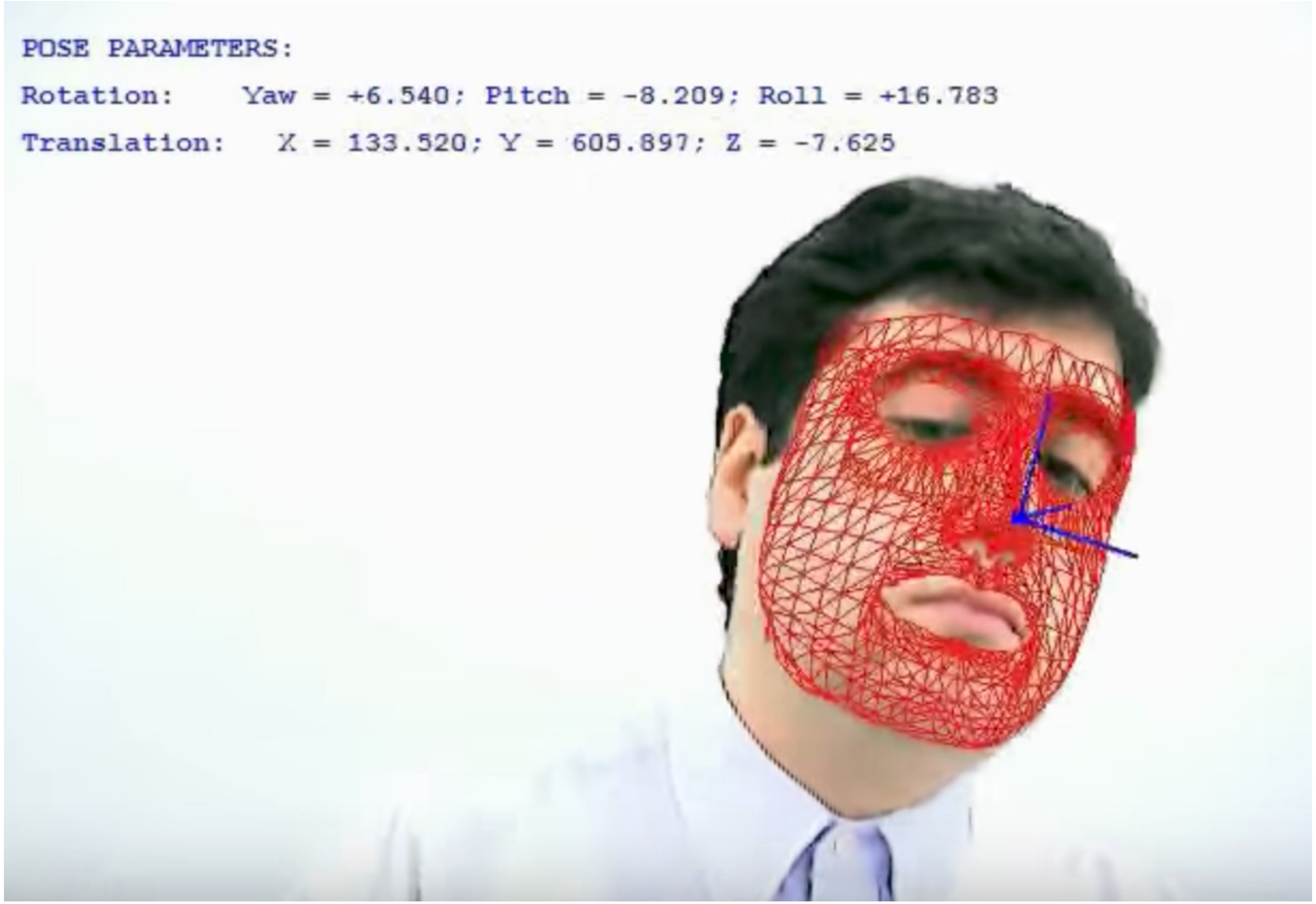


Demo: OpenTL Model Tracking



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

Demo: OpenTL Face Tracking



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

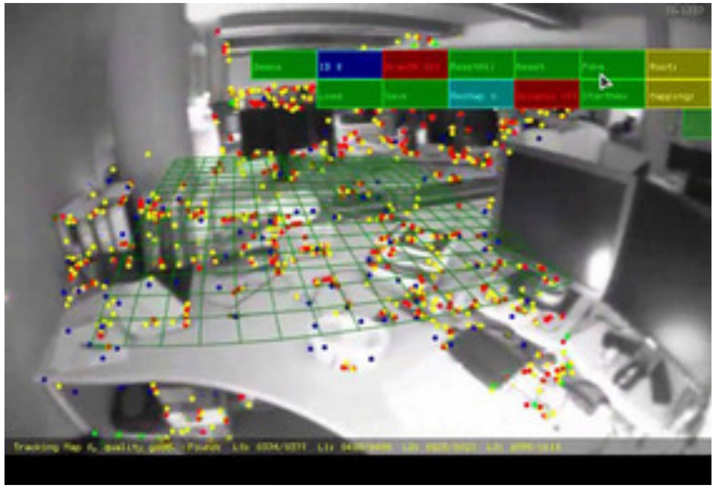
Marker vs. Natural Feature Tracking

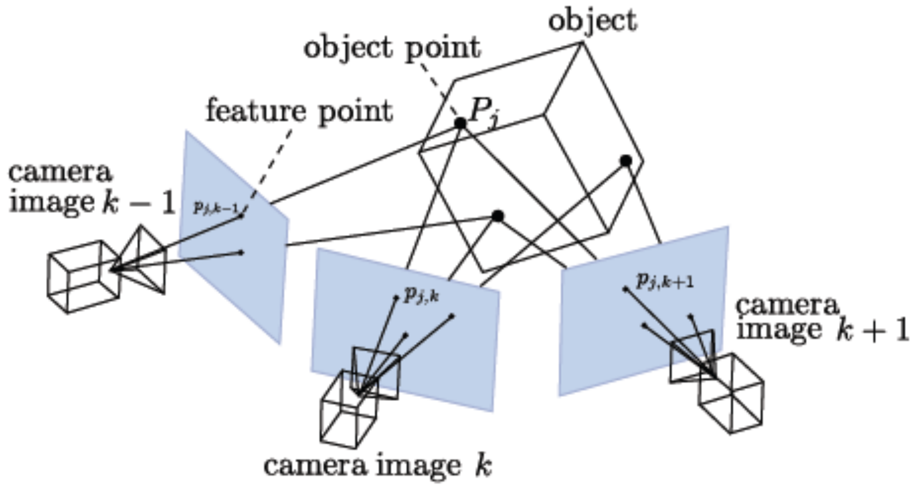
- Marker tracking
  - Usually requires no database to be stored
  - Markers can be an eye-catcher
  - Tracking is less demanding
  - The environment must be instrumented
  - Markers usually work only when fully in view
- Natural feature tracking
  - A database of keypoints must be stored/downloaded
  - Natural feature targets might catch the attention less
  - Natural feature targets are potentially everywhere
  - Natural feature targets work also if partially in view

Tracking from an Unknown Environment

- What to do when you don't know any features?
  - Very important problem in mobile robotics - Where am I?
- SLAM
  - Simultaneously Localize And Map the environment
  - Goal: to recover both camera pose and map structure while initially knowing neither. □
  - Mapping:
    - Building a map of the environment which the robot is in
  - Localisation:
    - Navigating this environment using the map while keeping track of the robot's relative position and orientation

Visual SLAM



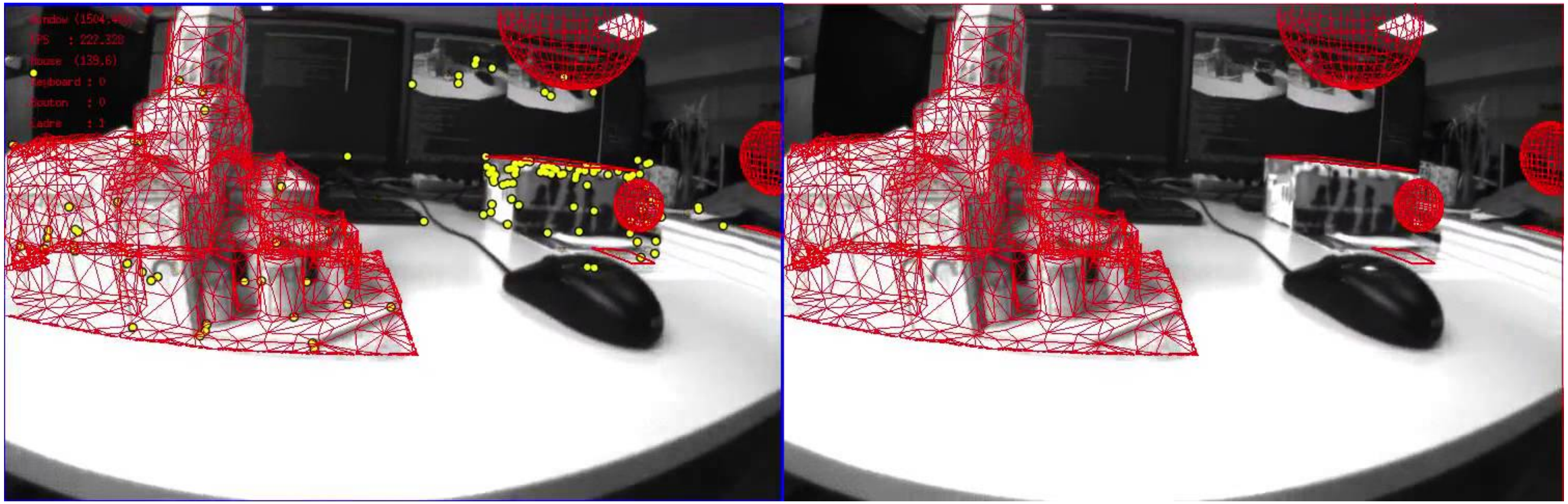


- Early SLAM systems (1986 -)
  - Computer visions and sensors (e.g. IMU, laser, etc.)
  - One of the most important algorithms in Robotics
- Visual SLAM
  - Using cameras only, such as stereo view
  - MonoSLAM (single camera) developed in 2007 (Davidson)

Example: Kudan MonoSLAM

https://www.youtube.com/watch?v=g2SFJGDz9cQ

How SLAM Works



- Three main steps
  1. Tracking a set of points through successive camera frames
  2. Using these tracks to triangulate their 3D position
  3. Simultaneously use the estimated point locations to calculate the camera pose which could have observed them
- By observing a sufficient number of points can solve for for both structure and motion (camera path and scene structure).

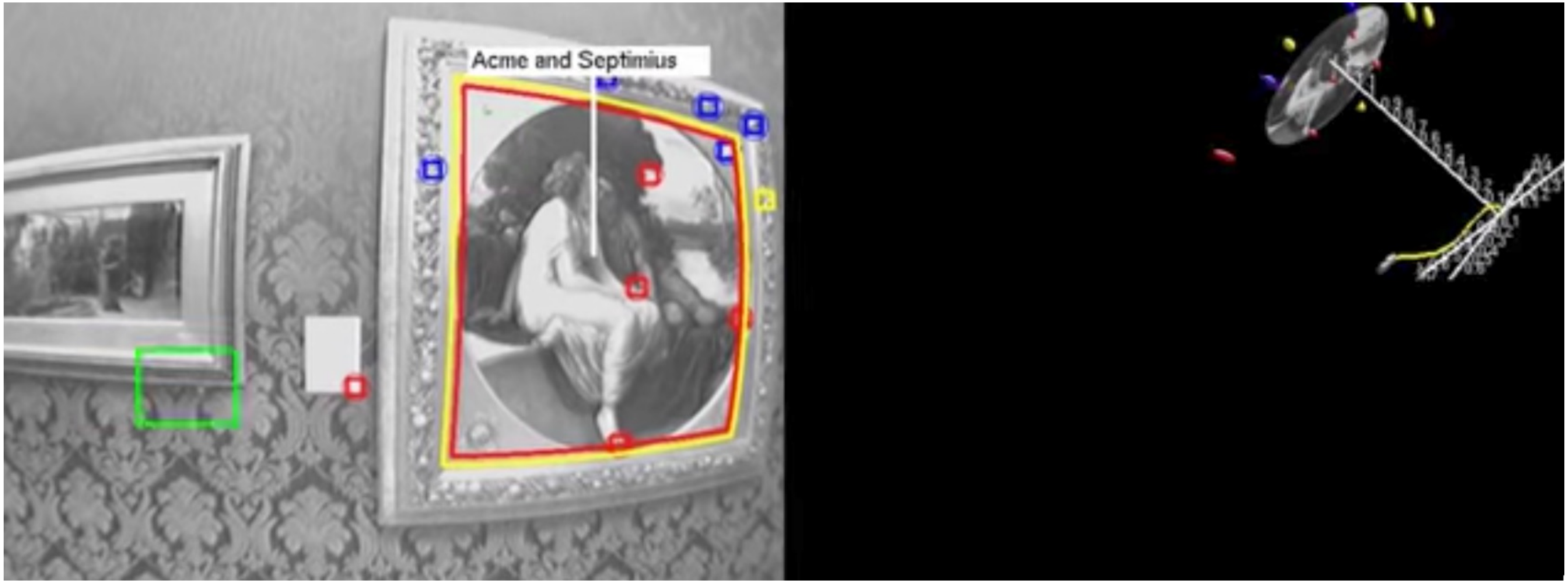
SLAM Optimization

- SLAM is an optimisation problem
  - compute the best configuration of camera poses and point positions in order to minimise reprojection error
    - difference between a point's tracked location and where it is expected to be
- Can be solved using bundle adjustment
  - a nonlinear least squares algorithm that finds minimum error
  - But – time taken grows as size of map increases
    - Multi-core machines can do localization and mapping on different threads
- Relocalisation
  - Allows tracking to be restarted when it fails

Evolution of SLAM Systems

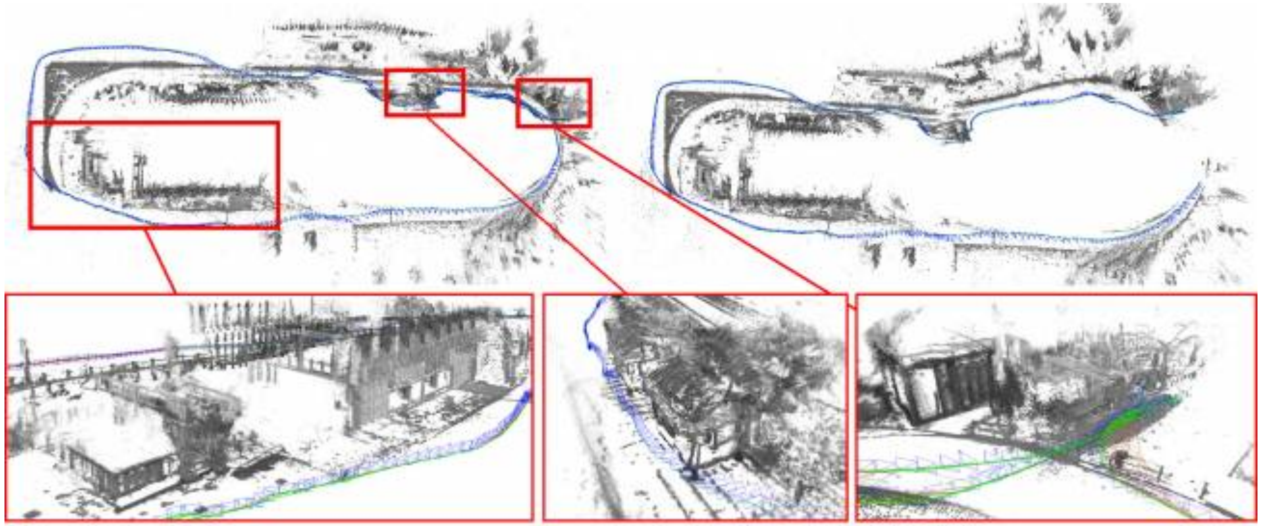
- MonoSLAM (Davidson, 2007)
  - Real time SLAM from single camera
- PTAM (Klein, 2009)
  - First SLAM implementation on mobile phone
- FAB-MAP (Cummins, 2008)
  - Probabilistic Localization and Mapping
- DTAM (Newcombe, 2011)
  - 3D surface reconstruction from every pixel in image
- KinectFusion (Izadi, 2011)
  - Realtime dense surface mapping and tracking using RGB-D

Demo: MonoSLAM



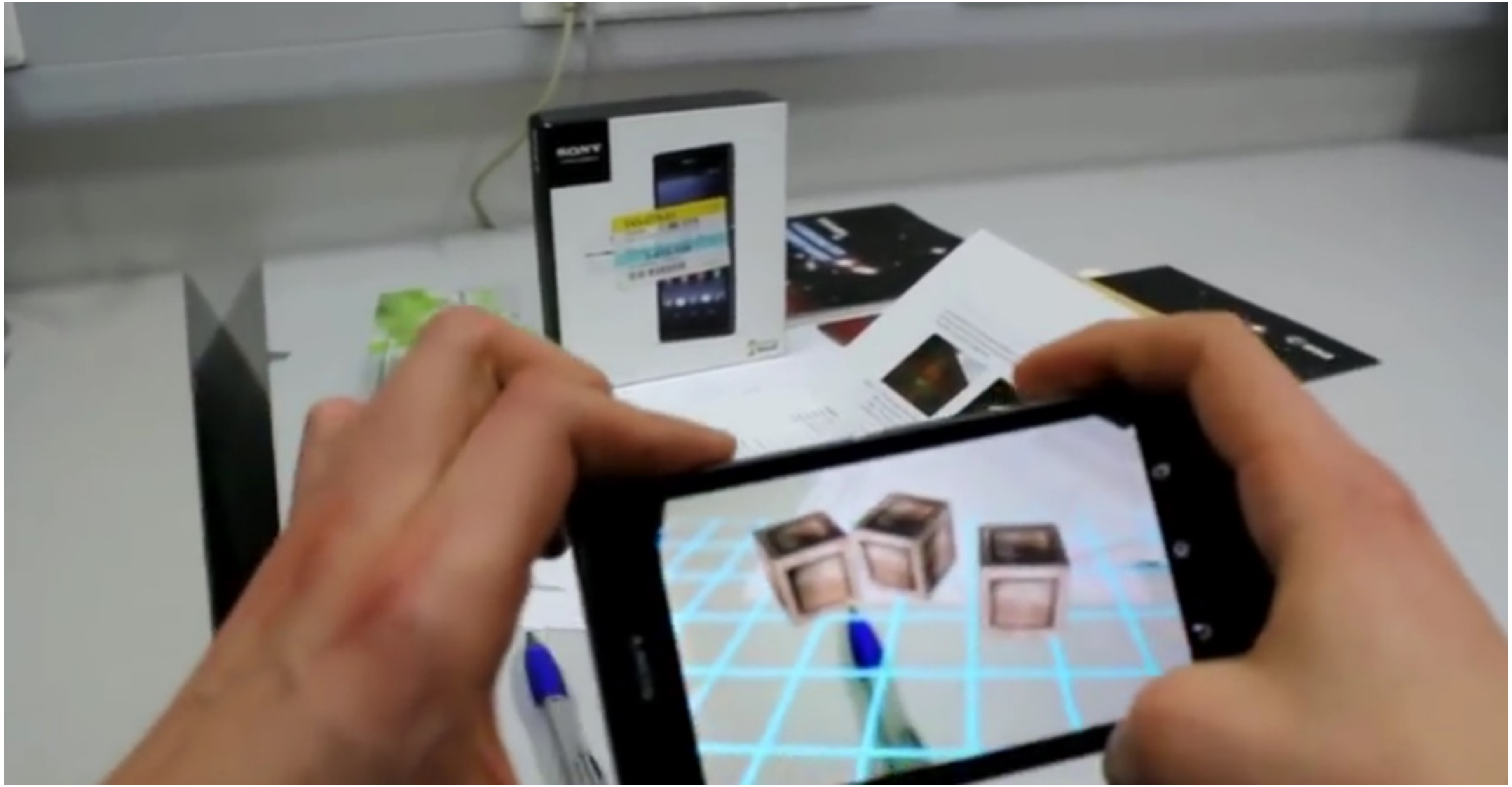
https://www.youtube.com/watch?v=saUE7JHU3P0

LSD-SLAM (Engel 2014)



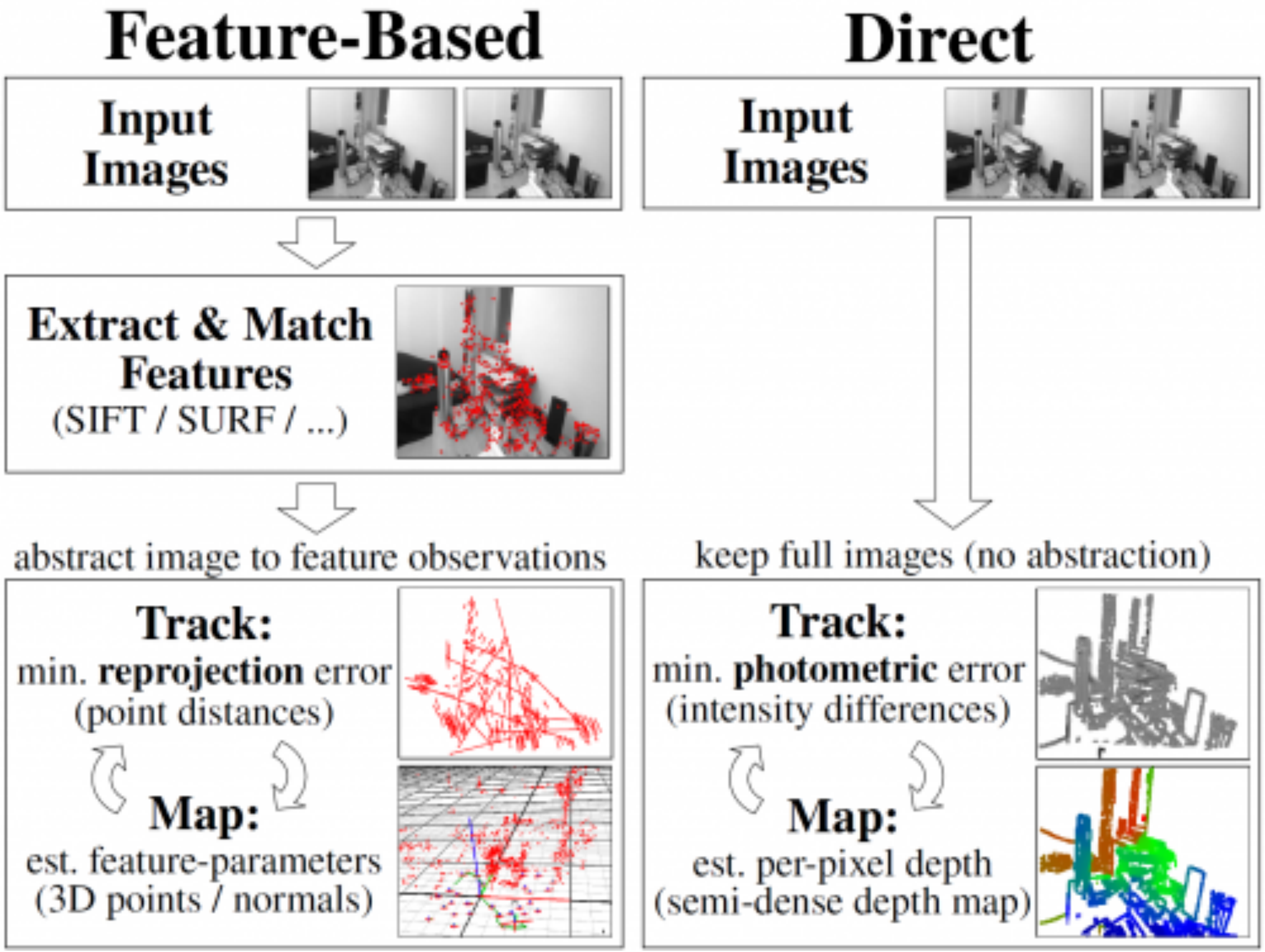
- A novel, direct monocular SLAM technique
- Uses image intensities both for tracking and mapping.
  - The camera is tracked using direct image alignment, while
  - Geometry is estimated as semi-dense depth maps
- Supports very large scale tracking
- Runs in real time on CPU and smartphone

Demo: LSD-SLAM



https://www.youtube.com/watch?v=GnuQzP3gty4

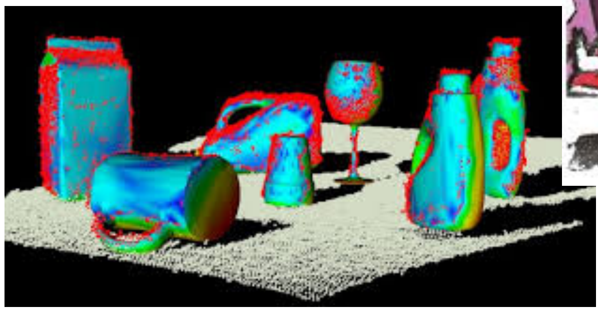
Direct Method vs. Feature Based



- Direct uses all information in image, cf feature based approach that only use small patches around corners and edges

Applications of SLAM Systems

- Many possible applications
  - Augmented Reality camera tracking
  - Mobile robot localisation
  - Real world navigation aid
  - 3D scene reconstruction
  - 3D Object reconstruction
  - Etc..



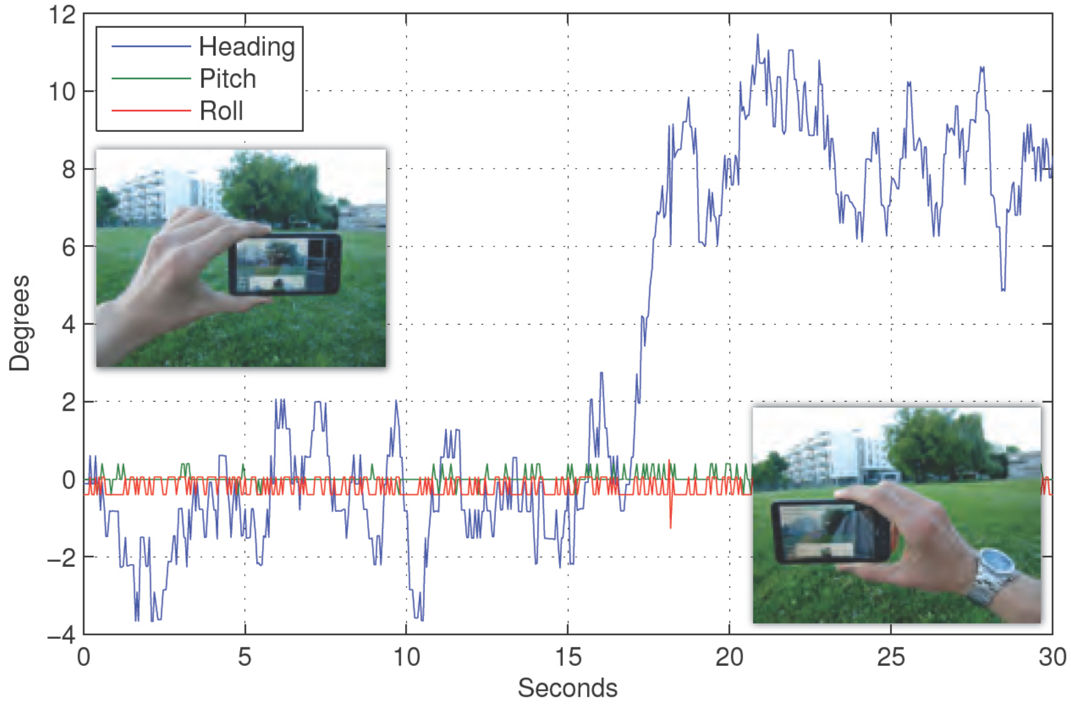
- Assumptions
  - Camera moves through an unchanging scene
  - So not suitable for person tracking, gesture recognition
    - Both involve non-rigidly deforming objects and a non-static map

Hybrid Tracking

Combining several tracking modalities together

Active-Active	vision-magnetic
Active-Passive	magnetic-vision
Active-Inertial	vision-inertial, acoustic-inertial
Passive-Inertial	
Passive-Inertial	compass-inertial, vision-inertial
Inertial-Inertial	

Sensor Tracking



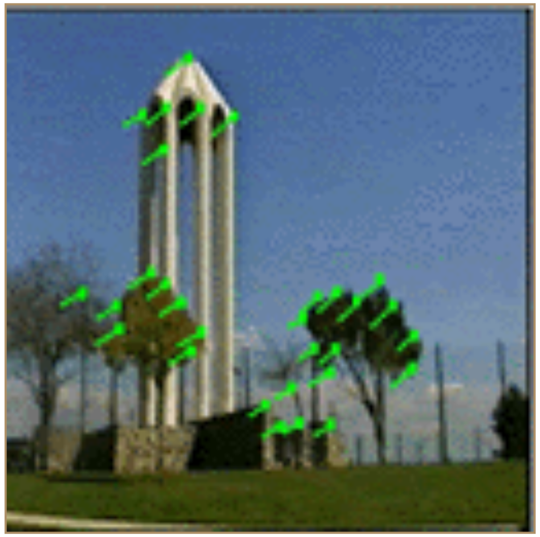
Inertial Compass Drifting Over Time

- Used by many "AR browsers"
- GPS, compass, accelerometer, gyroscope
- Not sufficient alone (drift, interference)

Combining Sensors and Vision

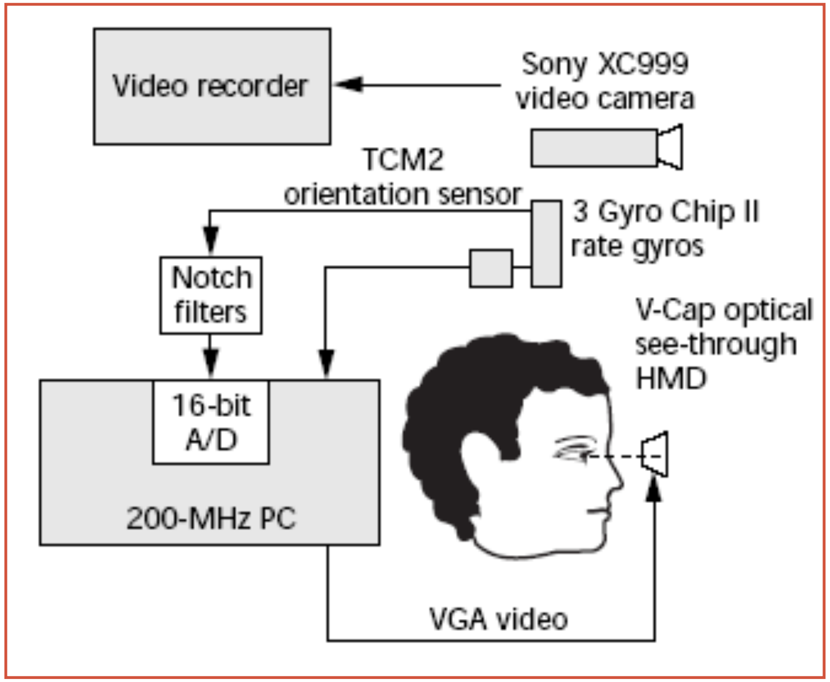
- Sensors
  - Produces noisy output (= jittering augmentations)
  - Are not sufficiently accurate (= wrongly placed augmentations)
  - Gives us first information on where we are in the world, and what we are looking at
- Vision
  - Is more accurate (= stable and correct augmentations)
  - Requires choosing the correct keypoint database to track from
  - Requires registering our local coordinate frame (online-generated model) to the global one (world)

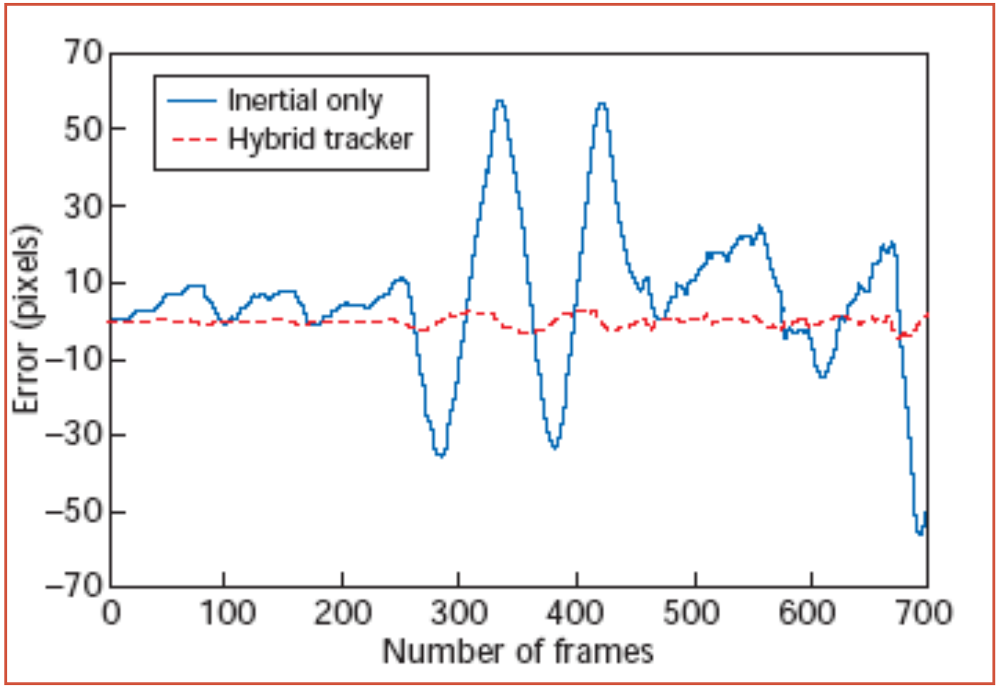
Example: Outdoor Hybrid Tracking



- Combines
  - computer vision
  - inertial gyroscope sensors
- Both correct for each other
  - Inertial gyro
    - provides frame to frame prediction of camera orientation, fast sensing
    - drifts over time
  - Computer vision
    - Natural feature tracking, corrects for gyro drift
    - Slower, less accurate

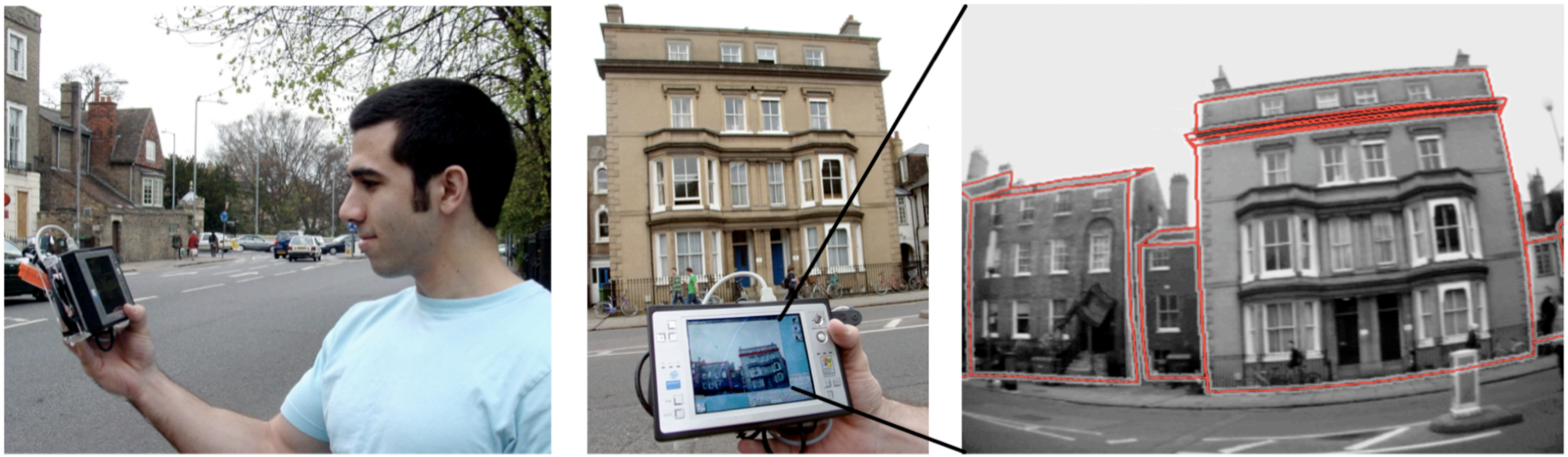
Outdoor AR Tracking System





You, Neumann, Azuma outdoor AR system (1999)

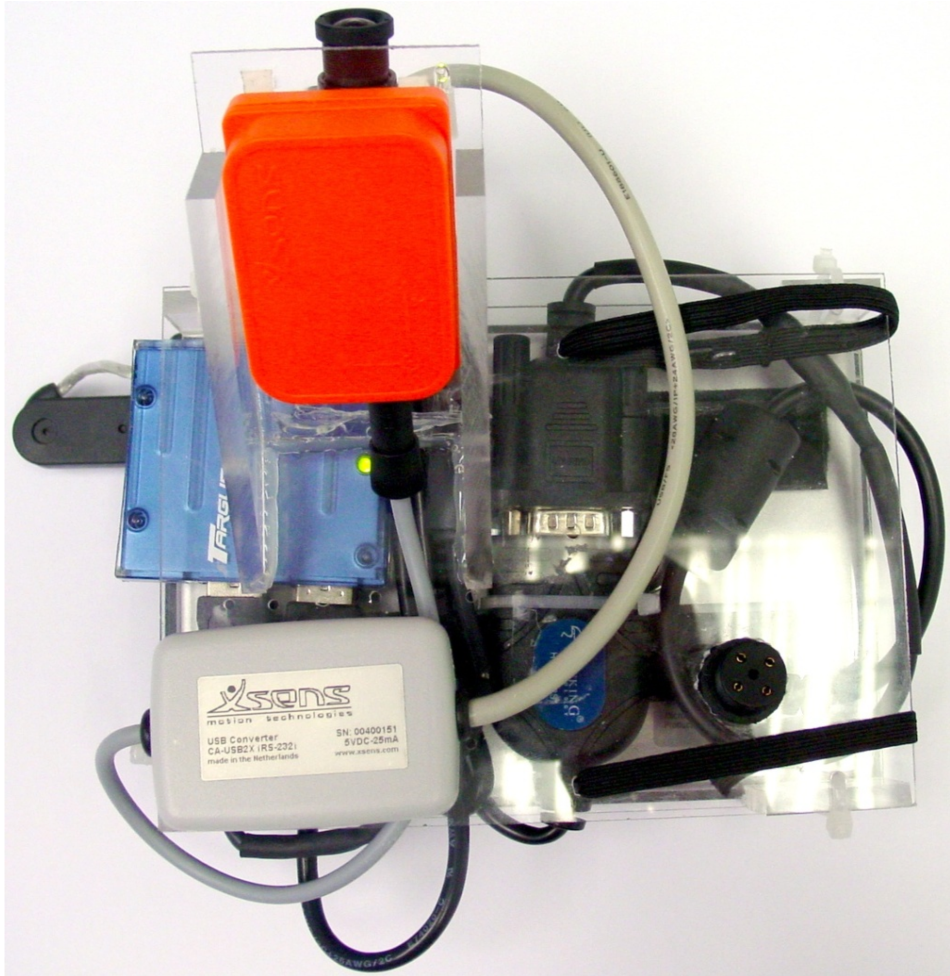
Robust Outdoor Tracking



- Hybrid Tracking
  - Computer Vision, GPS, inertial
- Going Out
  - Reitmayr & Drummond (Univ. Cambridge)

Reitmayr, G., & Drummond, T. W. (2006). Going out: robust model-based tracking for outdoor augmented reaity. In *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on* (pp. 109-118). IEEE.

Handheld Display



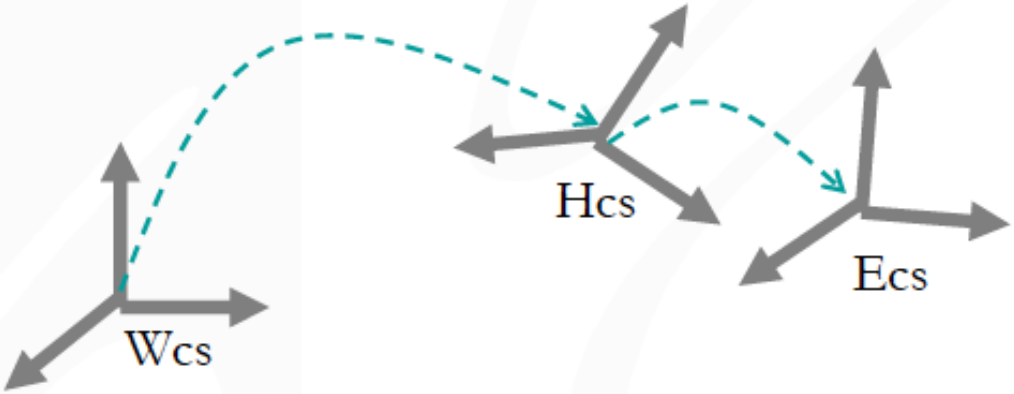
Example: Apple's ARKit

https://www.youtube.com/watch?v=6xDyVBsBtX8

REGISTRATION

Spatial Registration

- Defining Relative Position of Each Elements of a Scene



- Elements are: User, User's eye, Environment (Table, Room, Building), Objects, etc.
- Coordinate Systems (Euclidian System)
- Initially: Calibration
- Temporally: 3D/6D **Tracking**

The Registration Problem

- Virtual and Real content must stay properly aligned
- If not:
  - Breaks the illusion that the two coexist
  - Prevents acceptance of many serious applications

t = 0 seconds

t = 1 second

Sources of Registration Errors

- Static errors
  - Optical distortions (in HMD)
  - Mechanical misalignments
  - Tracker errors
  - Incorrect viewing parameters
- Dynamic errors
  - System delays (largest source of error)
    - 1 ms delay = 1/3 mm registration error

Reducing Static Errors

- Distortion compensation
  - For lens or display distortions
- Manual adjustments
  - Have user manually alighn AR and VR content
- View-based or direct measurements
  - Have user measure eye position
- Camera calibration (video AR)
  - Measuring camera properties

View Based Calibration (Azuma 94)

Figure 1: Wooden frame for calibration and registration

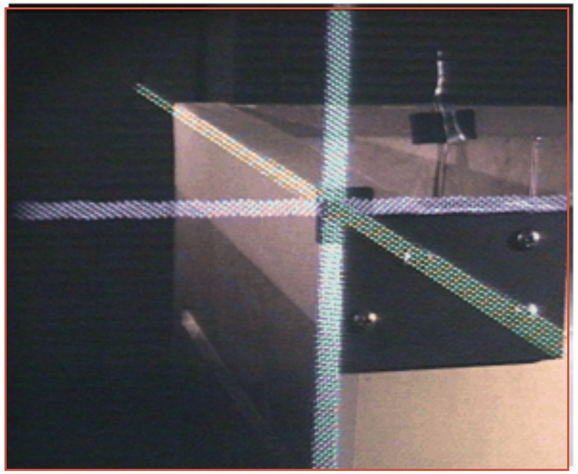


Figure 2: View seen in HMD, virtual axes on real frame

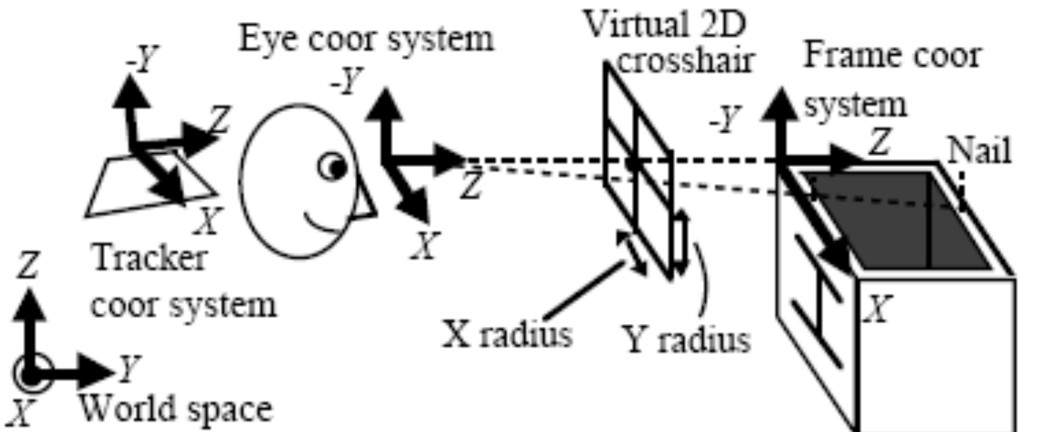
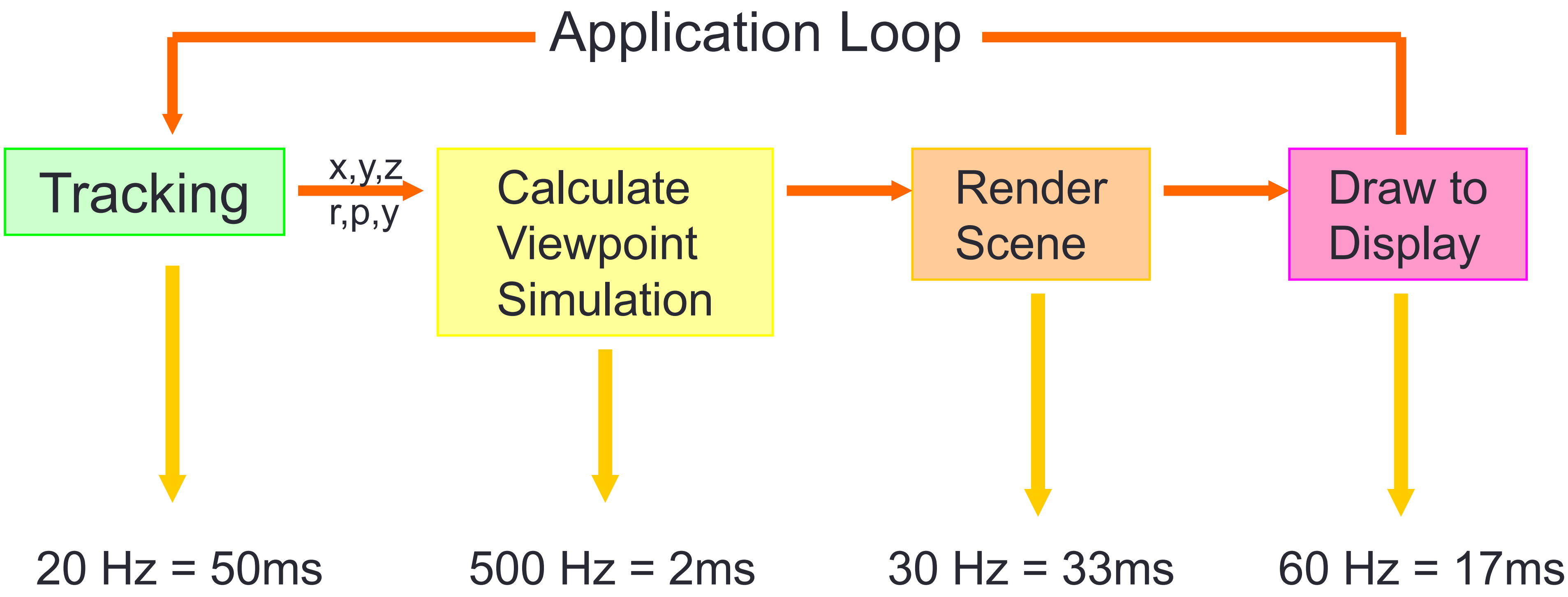


Figure 4: Virtual crosshair and coordinate systems

Dynamic errors

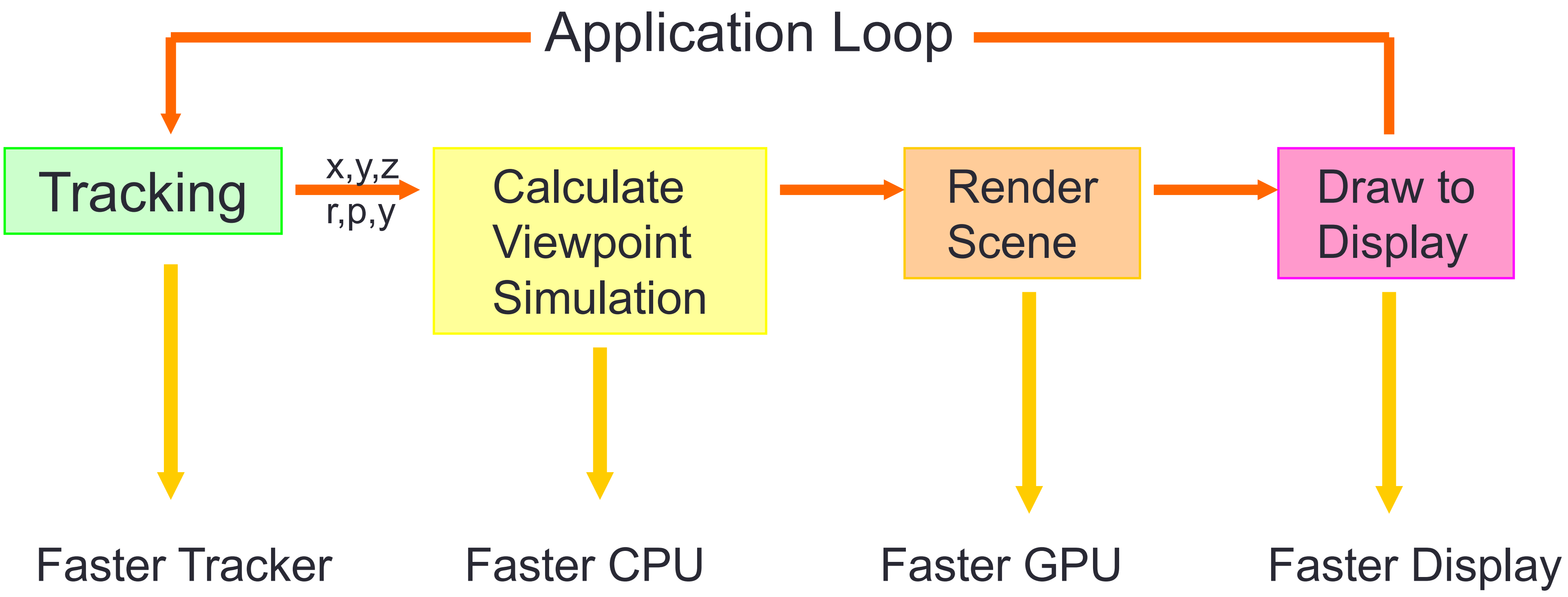


- Total Delay = 50 + 2 + 33 + 17 = 102 ms
  - 1 ms delay = 1/3 mm = 33mm error

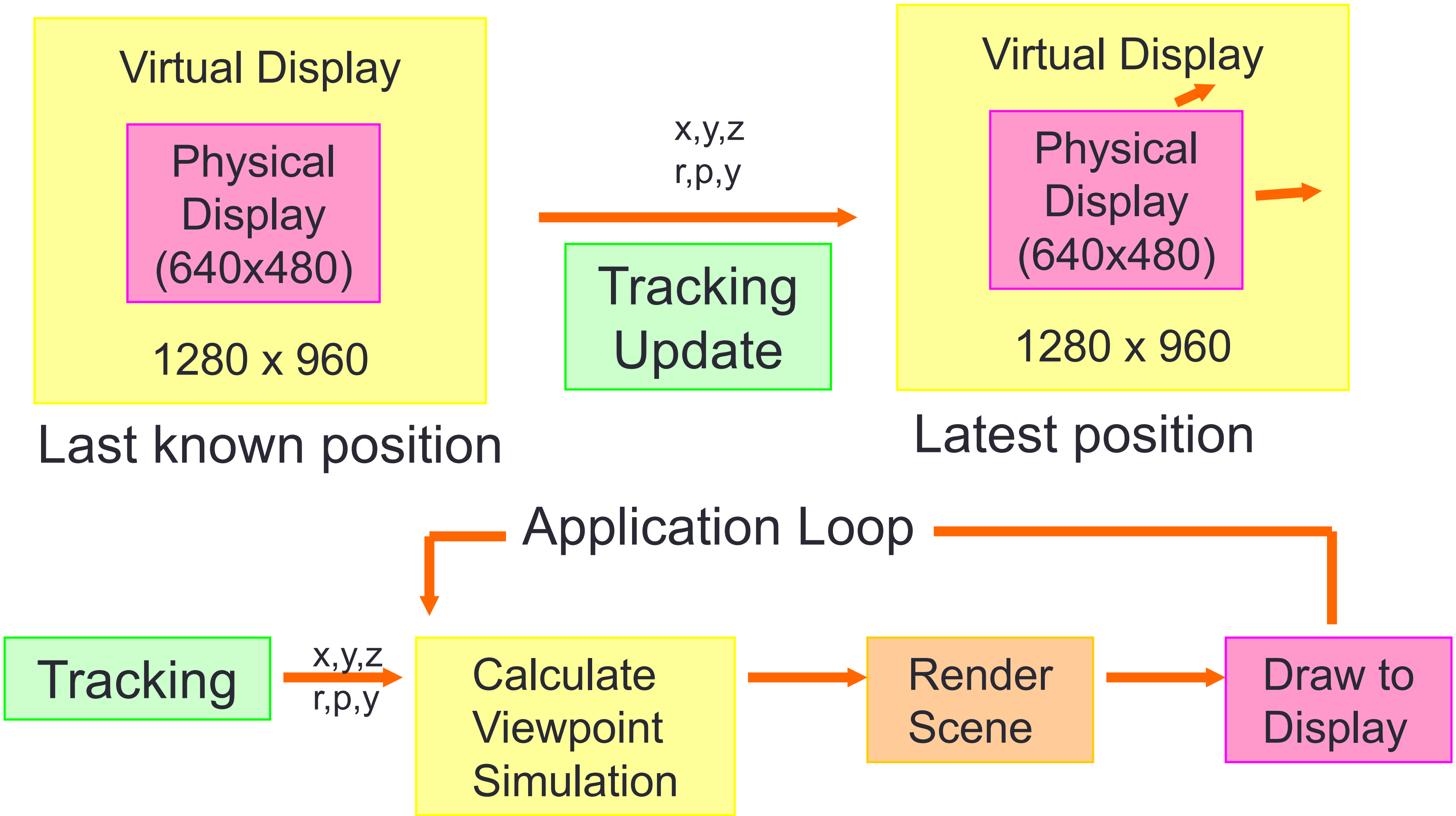
Reducing dynamic errors (I)

- Reduce system lag
  - Faster components/system modules
- Reduce apparent lag
  - Image deflection
  - Image warping

Reducing System Lag

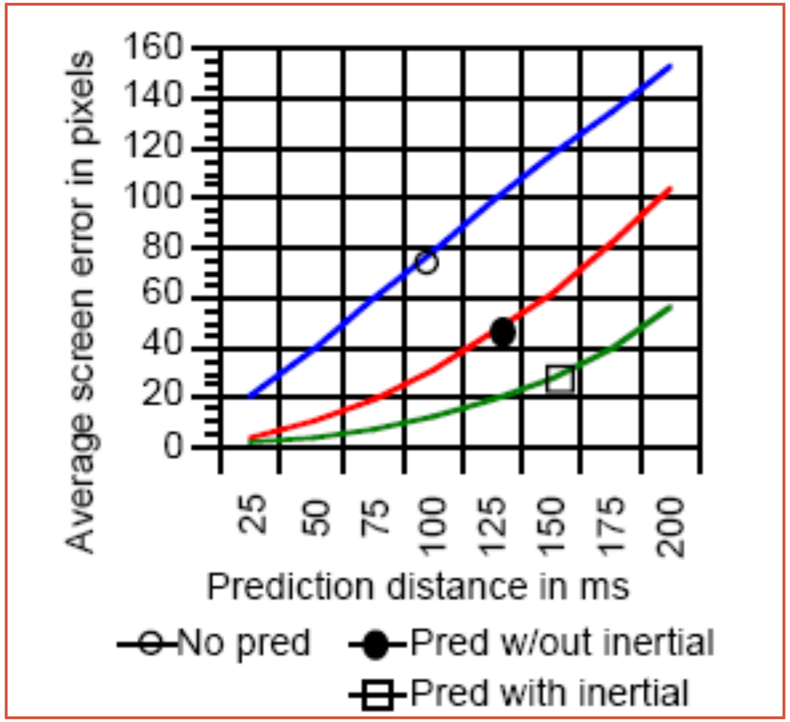


Reducing Apparent Lag



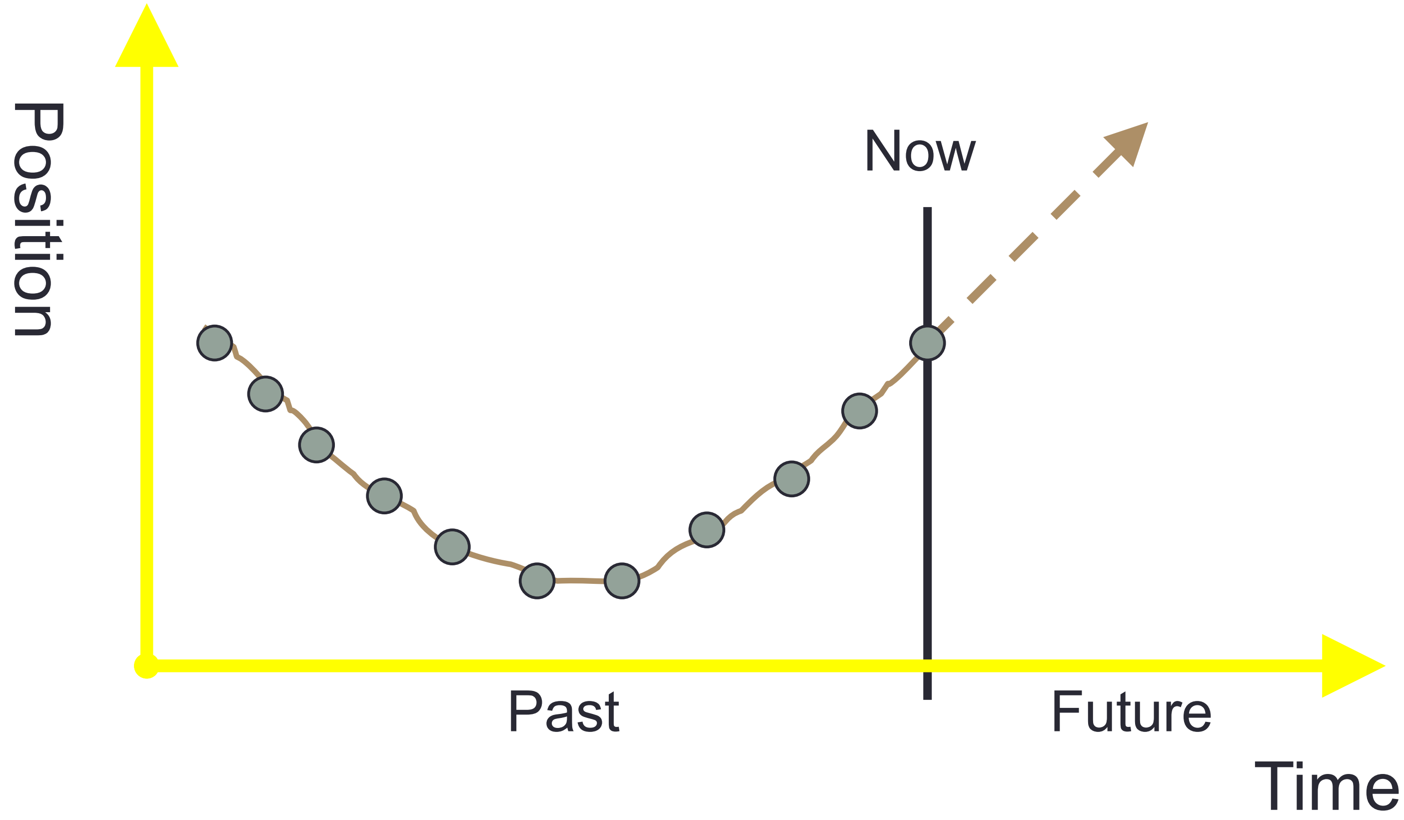
Reducing dynamic errors (2)

- Match video + graphics input streams (video AR)
  - Delay video of real world to match system lag
  - User doesn't notice
- Predictive Tracking
  - Inertial sensors helpful



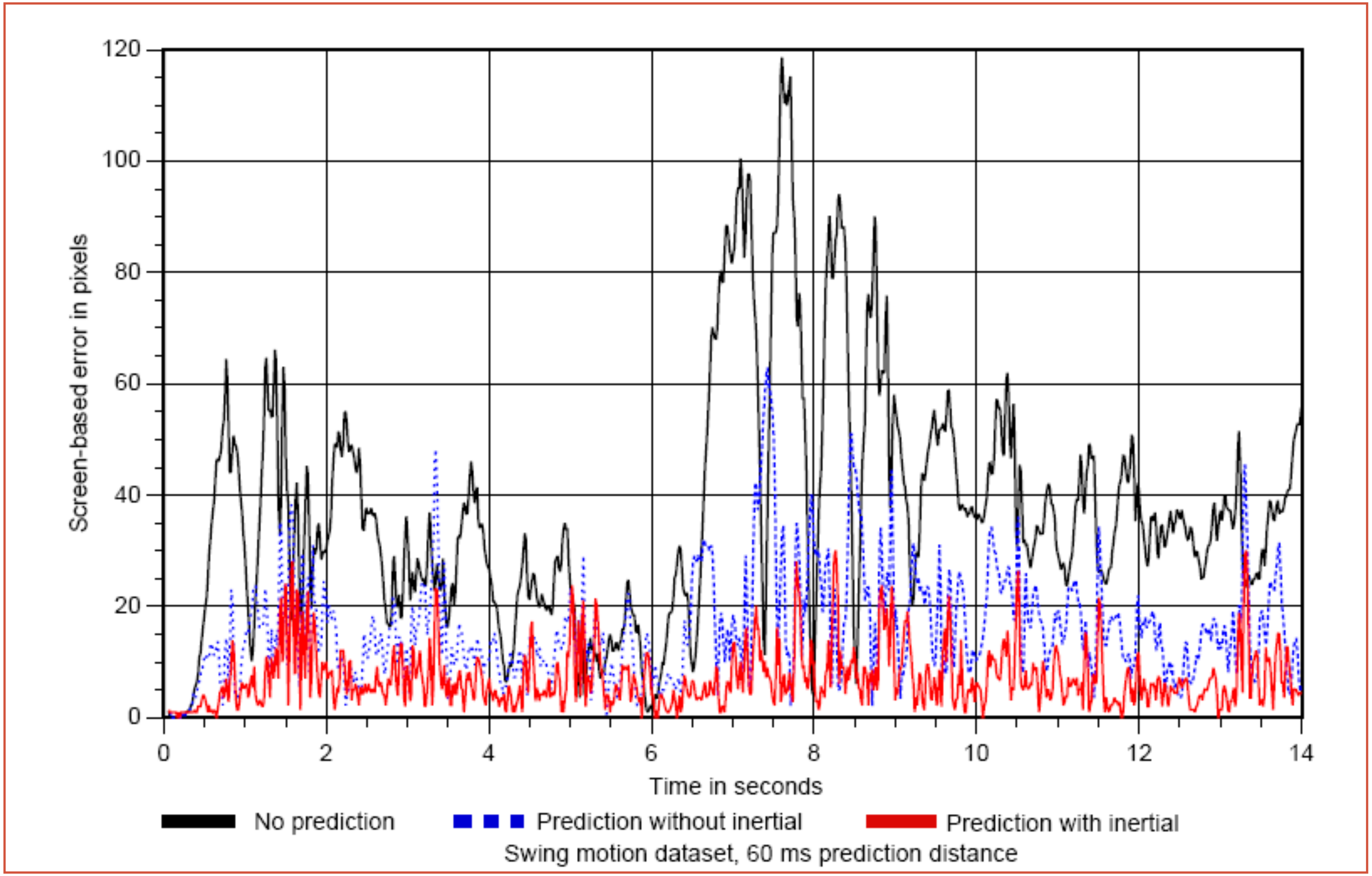
Azuma / Bishop 1994

Predictive Tracking



Can predict up to 80 ms in future (Holloway)

Predictive Tracking (Azuma 94)



Wrap-up

- Tracking and Registration are key problems
- Registration error
  - Measures against static error
  - Measures against dynamic error
- AR typically requires multiple tracking technologies
  - Computer vision most popular
- Research Areas:
  - SLAM systems, Deformable models, Mobile outdoor tracking

www.empathiccomputing.org

mark.billinghurst@unisa.edu.au

@marknb00