Extended Reality: Towards Spatial Intelligence

IAGO SUÁREZ

QUALCOMM XR LABS

About Me

2015 — Software Engineer (UDC)

2016 — Master in Artificial Intelligence (UPM)
Starting PhD --> 🎓

2017 — XOIA Foundation

2018 — Line segment detection (IROS)

2019-21 — Feature Description - OpenCV

2021 — ETH Zürich Research Internship

2022 — New Horizon Technologies: AI Team Leader

2023 — Qualcomm XR Lab Europe

Index

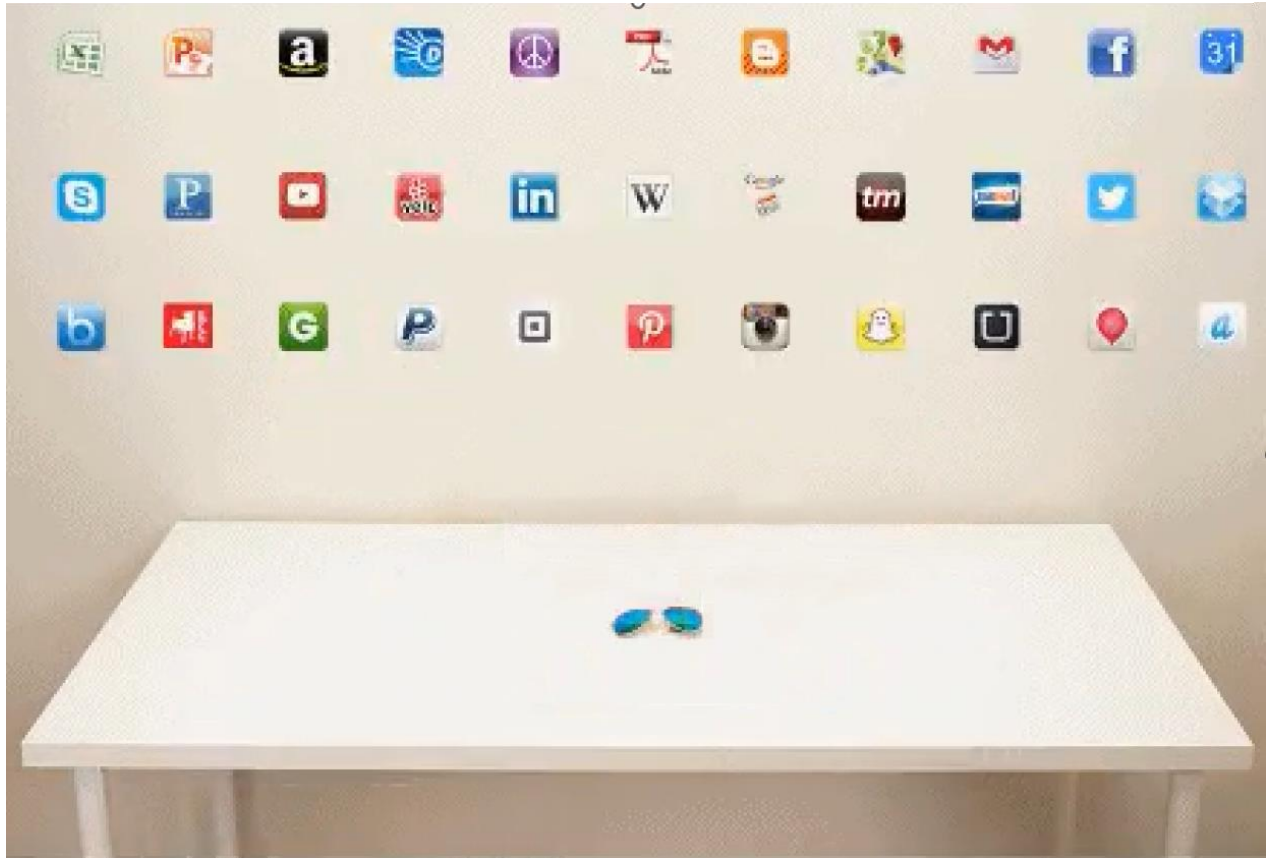
1. Intro: What is Extended Reality?
2. How does it work?
3. Computer Vision
4. Computer Graphics
5. Examples & Applications

What is Extended Reality?

https://www.youtube.com/watch?v=m_o99lJb-_4

Motivation

Motivation: Extended Reality



Motivation

Entertainment

Operator Training

Retail

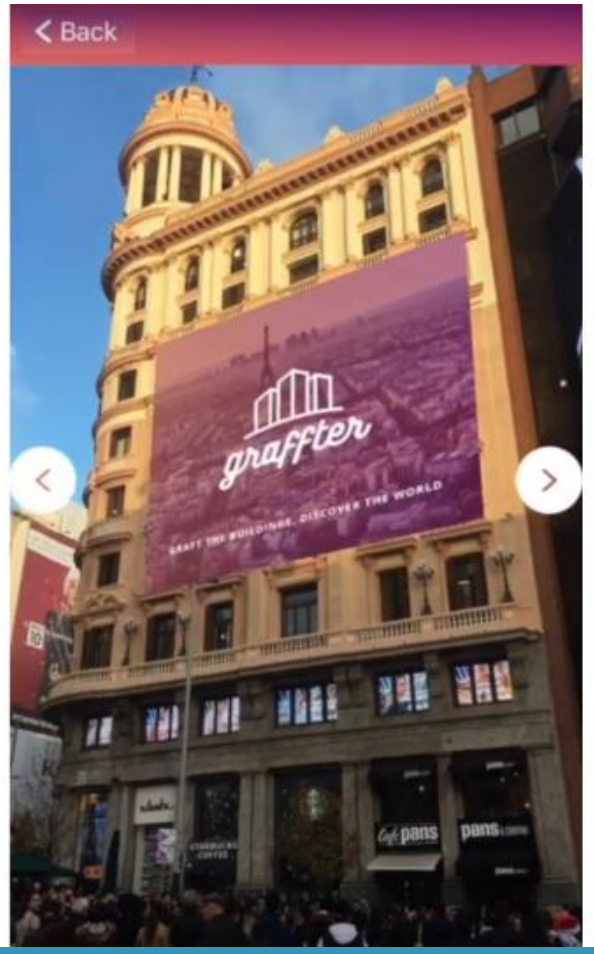
Virtual Try-on

History

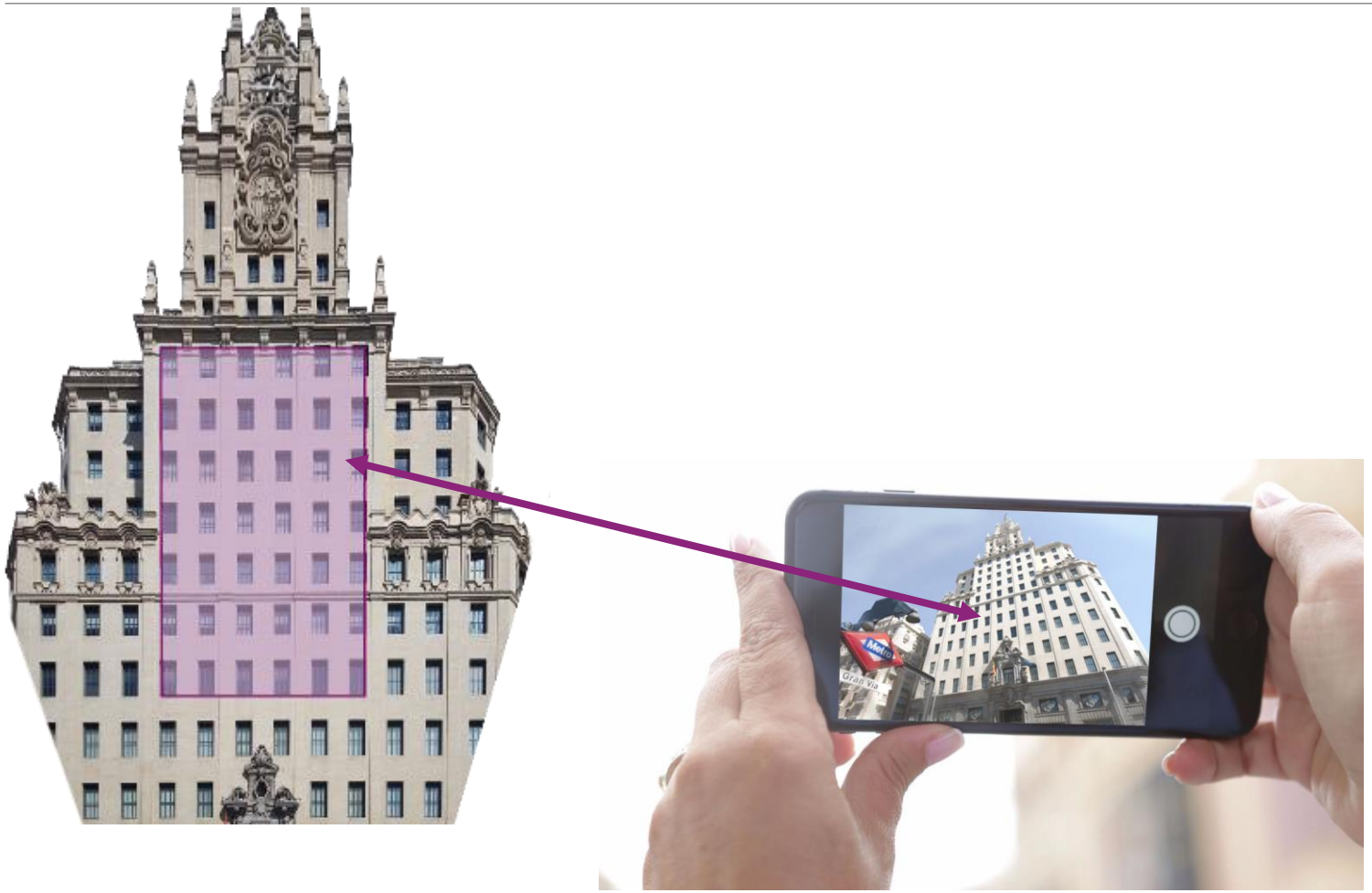
Real state & Architecture

How does it work?

Example: The Graffter - Urban Mixed Reality

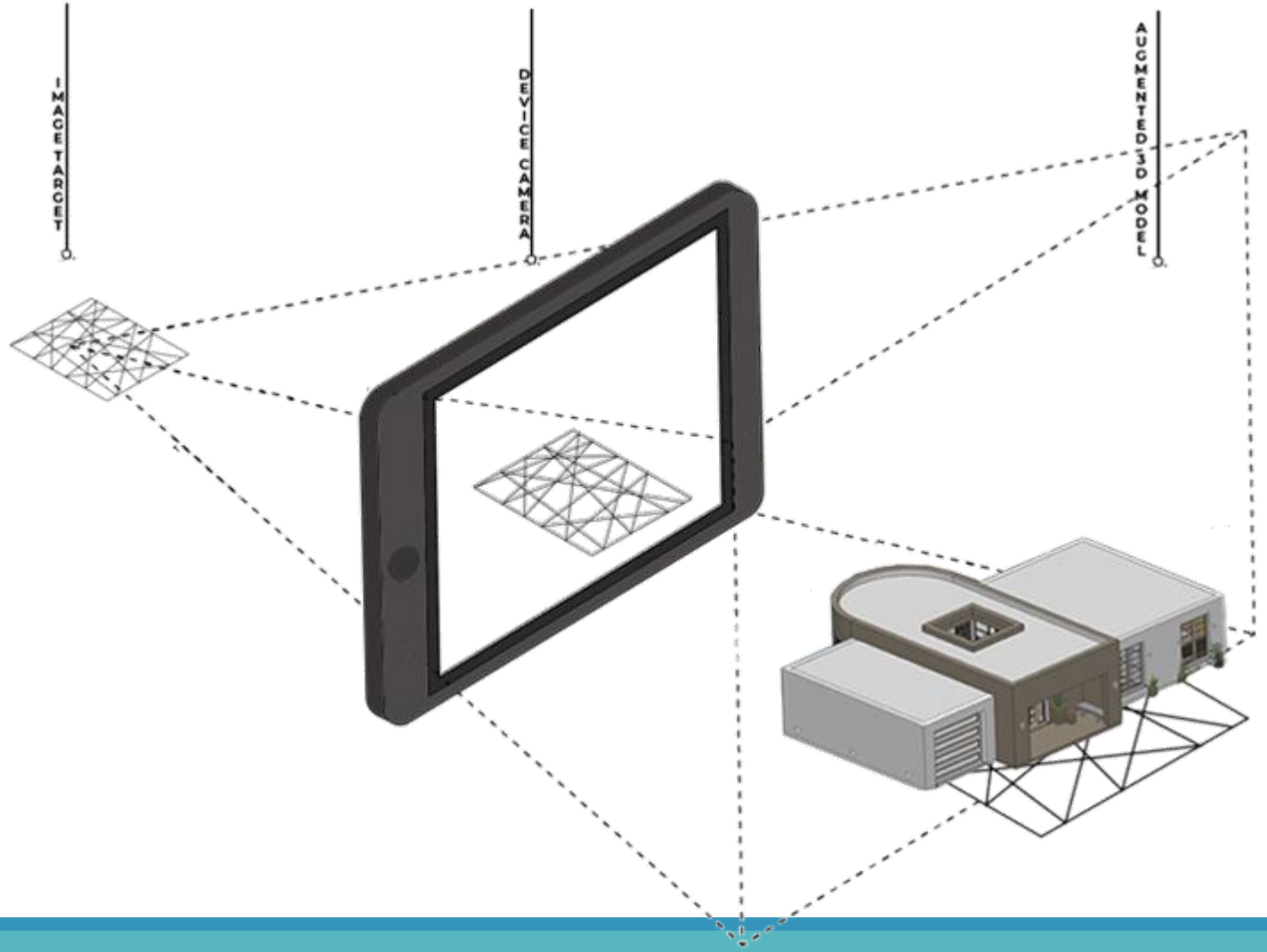


How does it work?

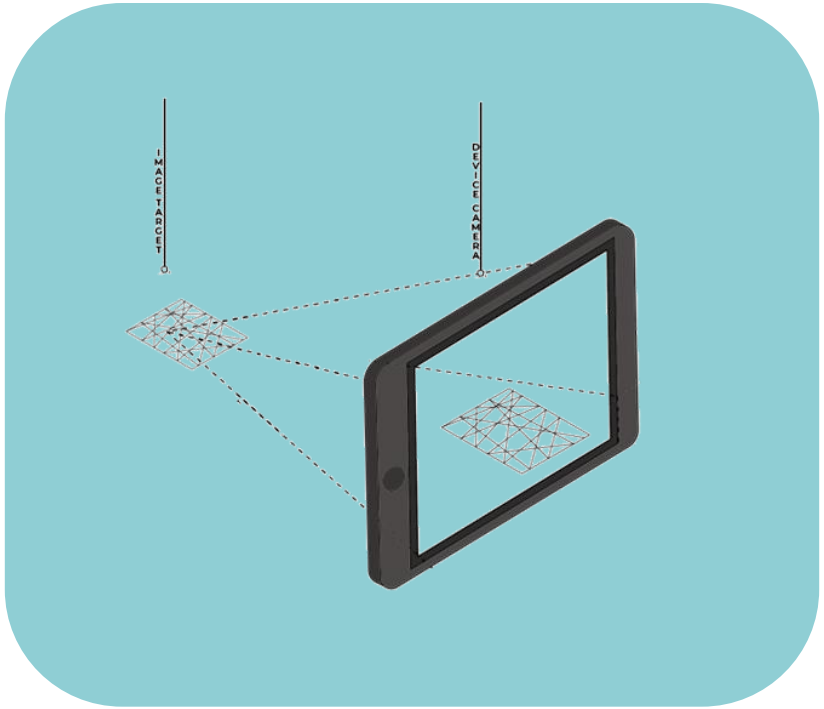


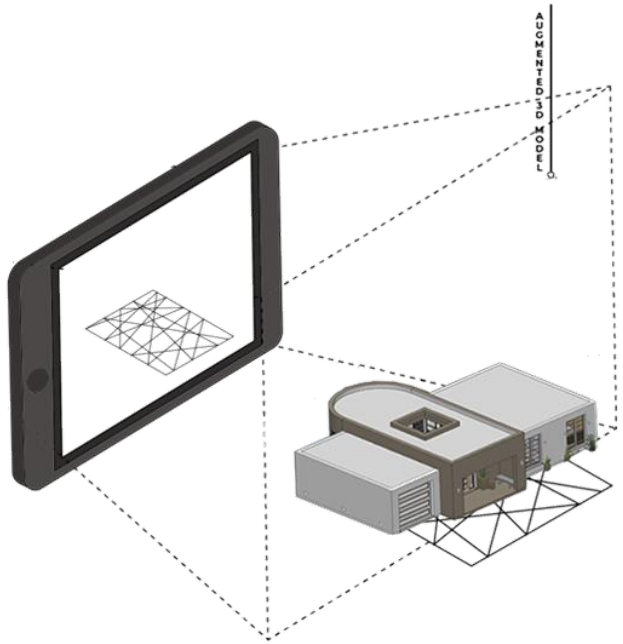
We need to know the geometric transformation between the image and the facade

How does it work?



How does it work?





1. Computer Vision

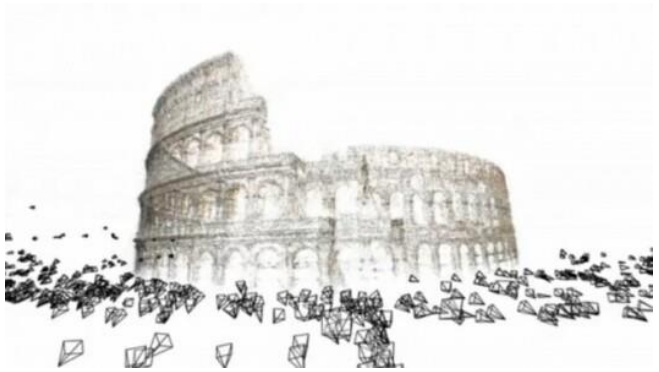
2. Computer Graphics

Computer Vision?

Self-driving cars

Augmented & Virtual Reality

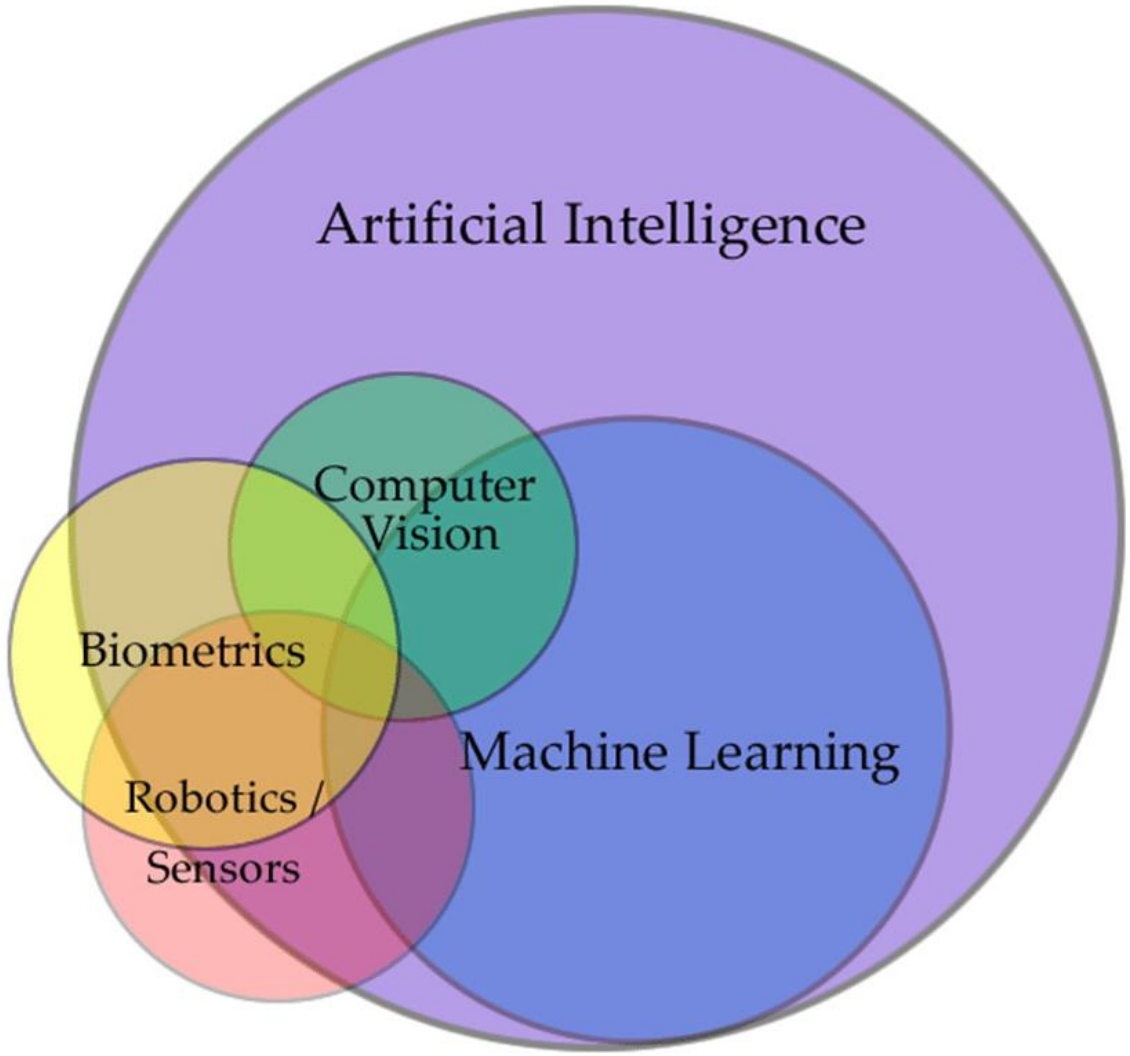
Robotics



3D Reconstruction

Face Apps

Computer Vision (CV) inside AI



Let’s review three different examples

QR / ARUCO

Marker-less Target

Real life (SLAM)

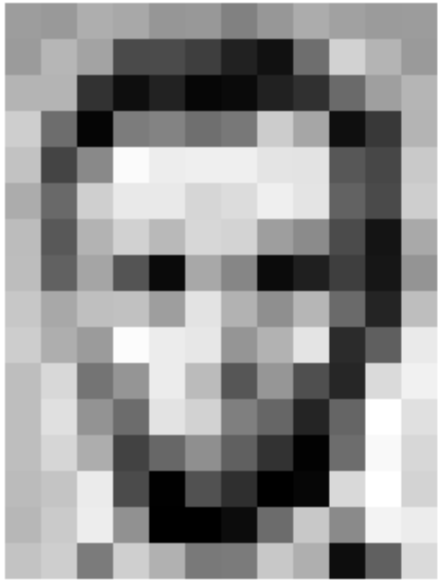
Let’s review three different examples

QR / ARUCO

Marker-less Target

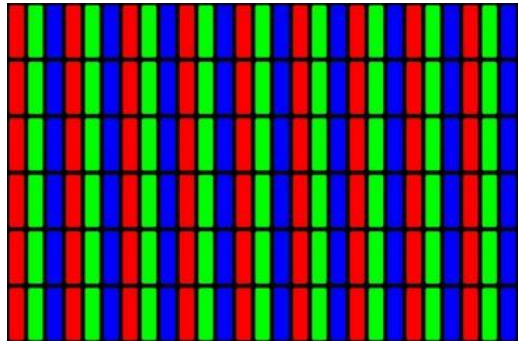
Real life (SLAM)

What is an image?

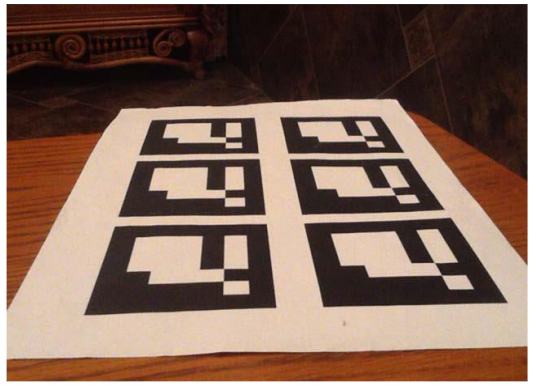


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218



QR / ARUCO Detection



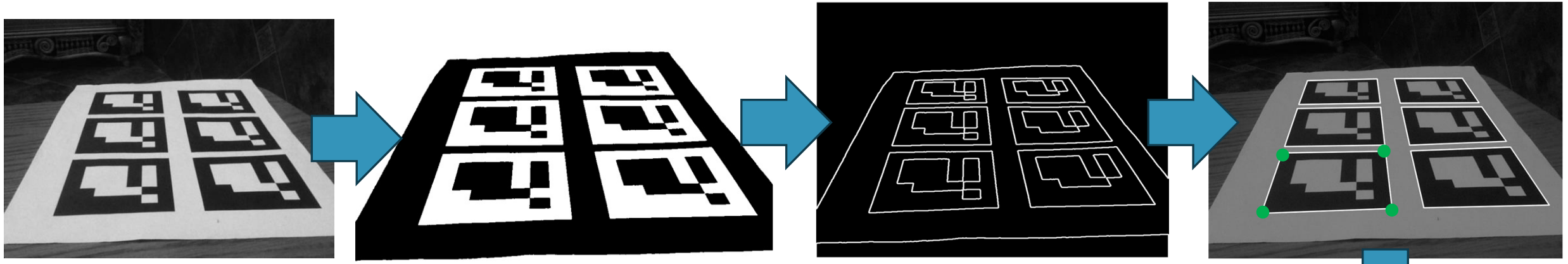
Mastering OpenCV with Practical Computer Vision Projects

QR / ARUCO Detection

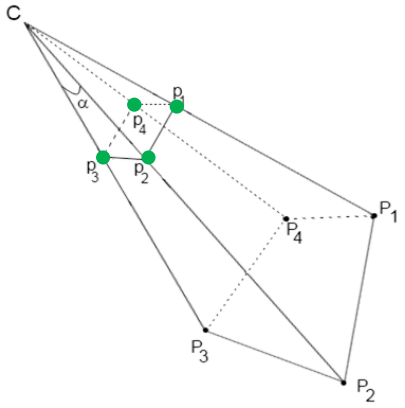
Image Binarization

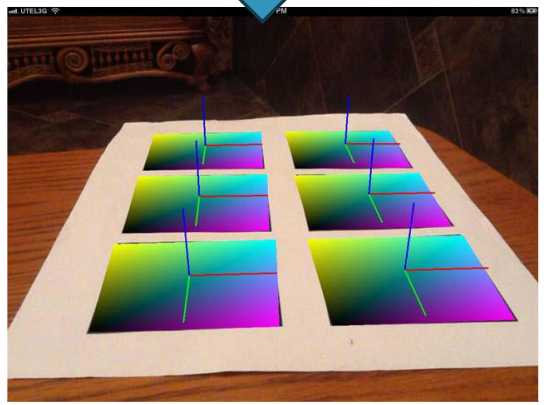
Contour Detection

Contour Selection



Pose Estimation





Mastering OpenCV with Practical Computer Vision Projects

QR / ARUCO Detection

Image Binarization

Contour Detection

Contour Selection

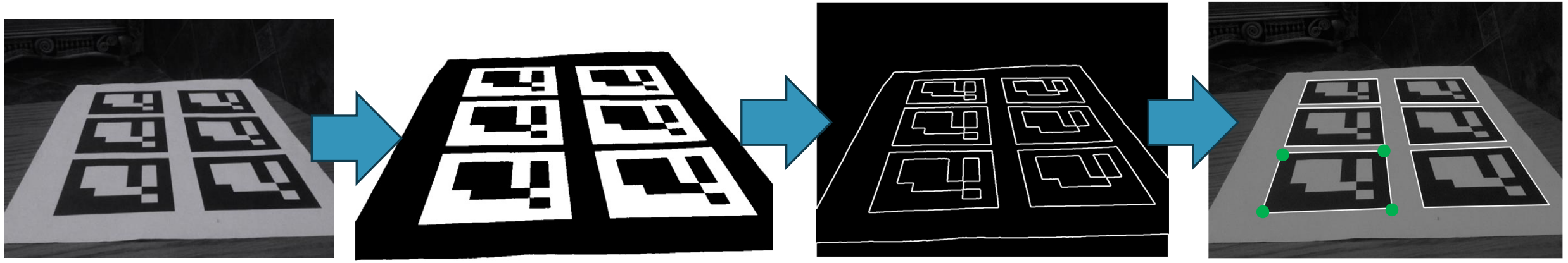
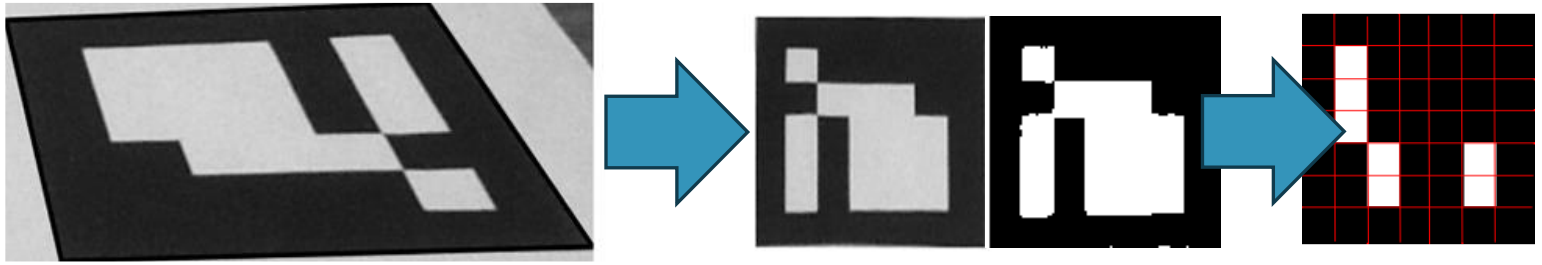


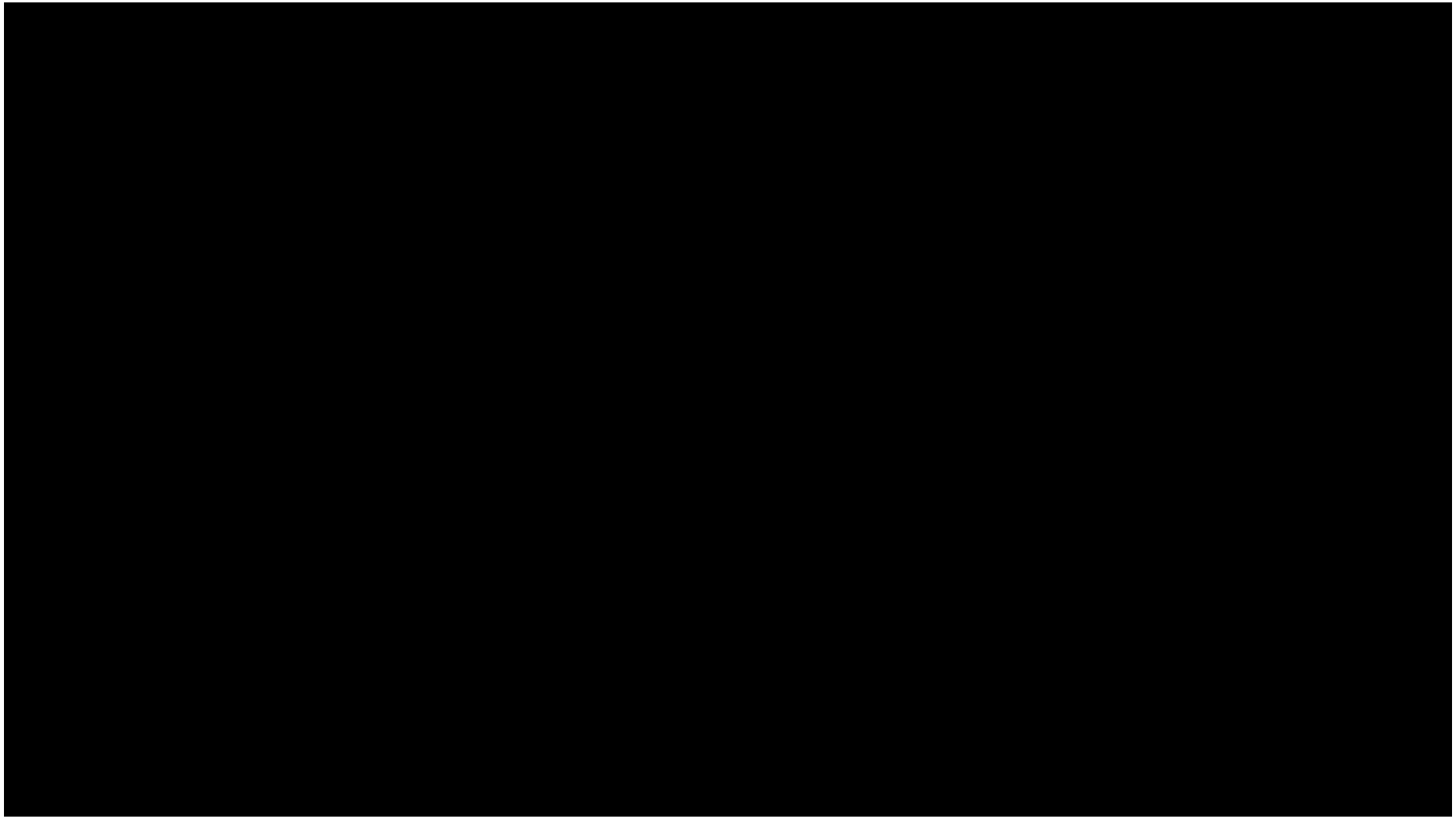
Image warping

Identification



Mastering OpenCV with Practical Computer Vision Projects

AR With QR-like Pattern



Let’s review three different examples

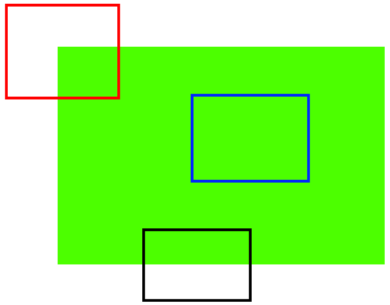
QR / ARUCO

Marker-less Target

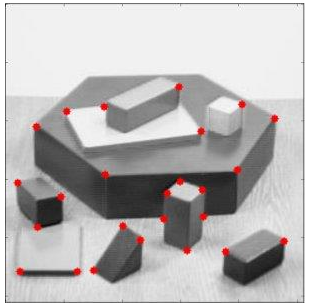
Real life (SLAM)

Marker-less Target

What points do we want to use?



cv.goodFeaturesToTrack()



Feature Detection and Description - OpenCV

Identifying Corners/Blobs: SIFT

- SIFT (Lowe, 1999) is the most widely used descriptor:

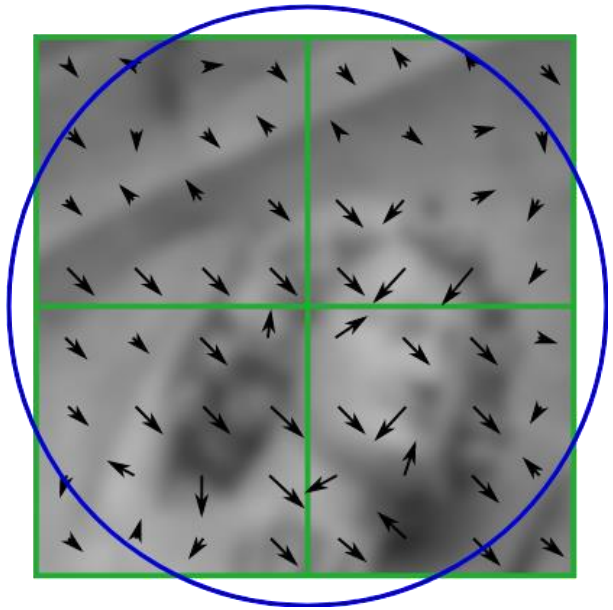


Image gradients

Identifying Corners/Blobs: SIFT

- Uses the histograms of gradients in a fixed grid:

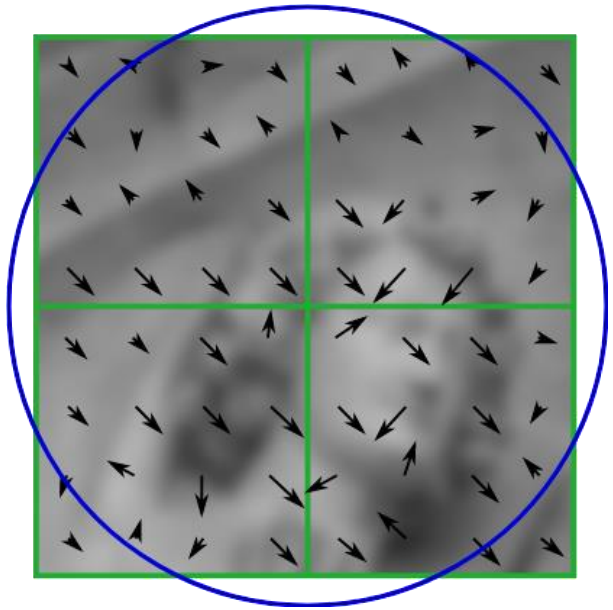
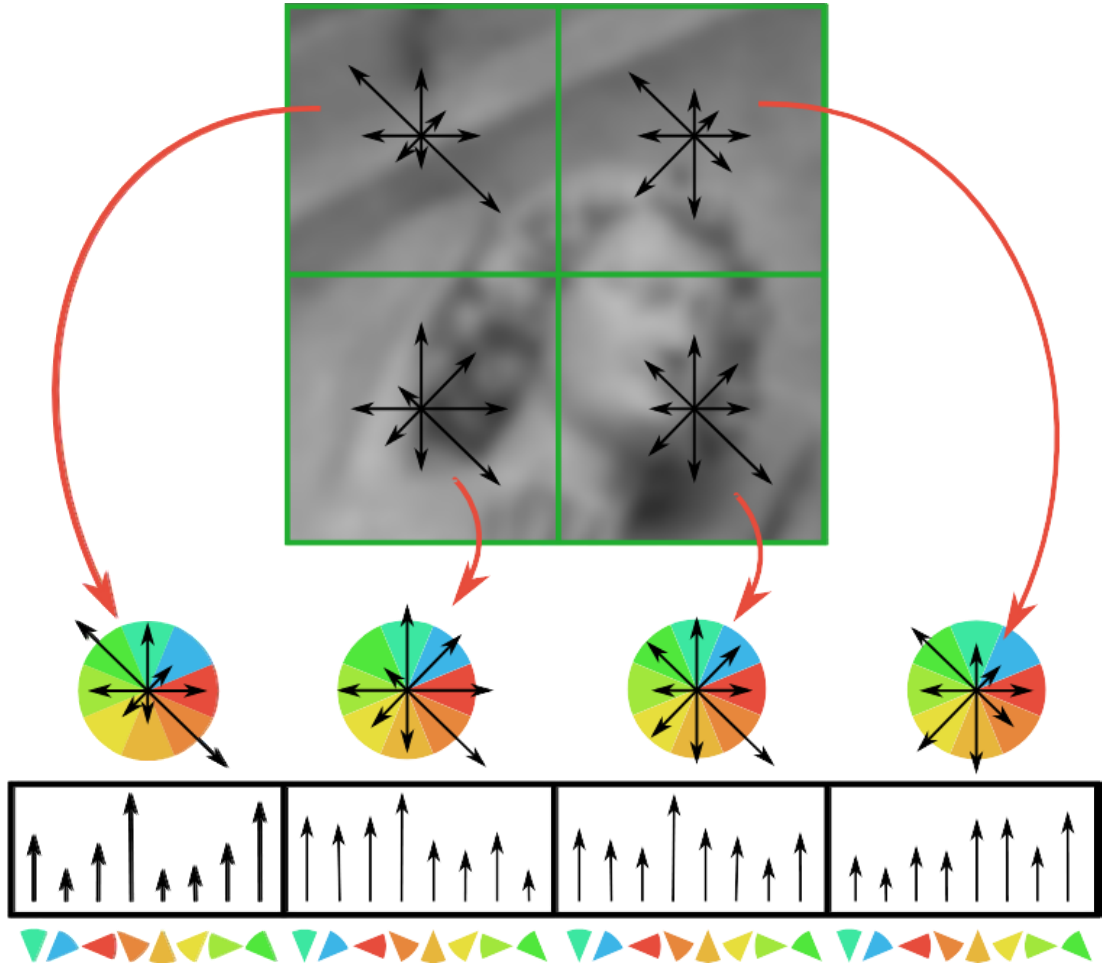


Image gradients

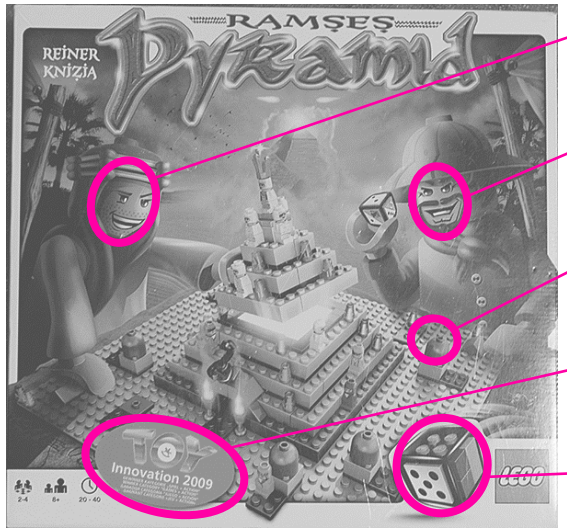


Feature Matching

Reference Image

Feature Matching

Reference Image

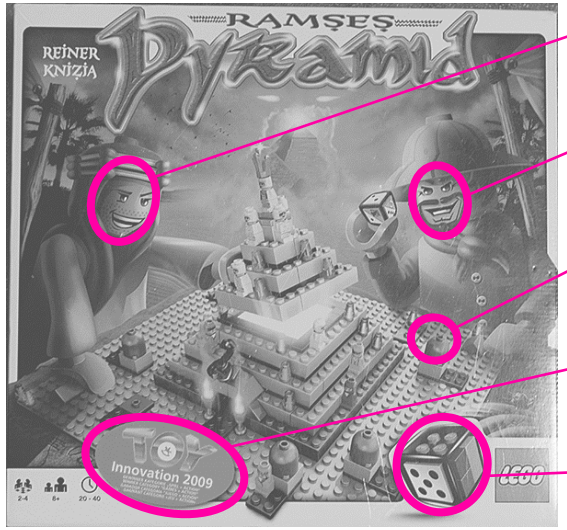


Descriptor

.2	.1	.8	.2	.3
.4	.1	.1	.2	.3
.2	.3	.8	.0	.9
.4	.1	.4	.2	.1
.1	.7	.8	.8	.6

Feature Matching

Reference Image

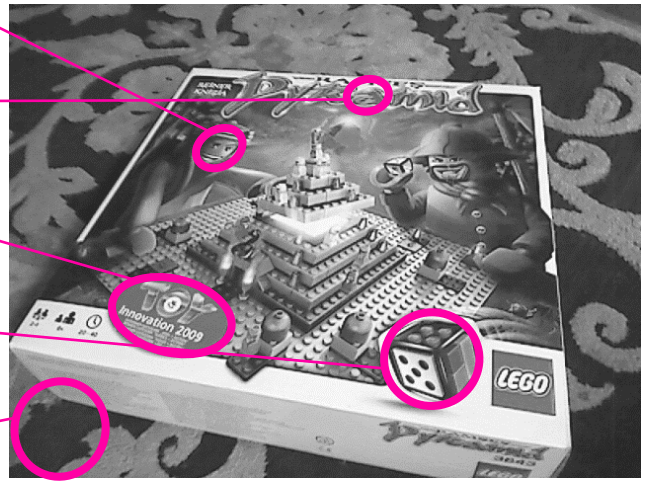


.2	.1	.8	.2	.3
.4	.1	.1	.2	.3
.2	.3	.8	.0	.9
.4	.1	.4	.2	.1
.1	.7	.8	.8	.6

Nearest Neighbor Matching

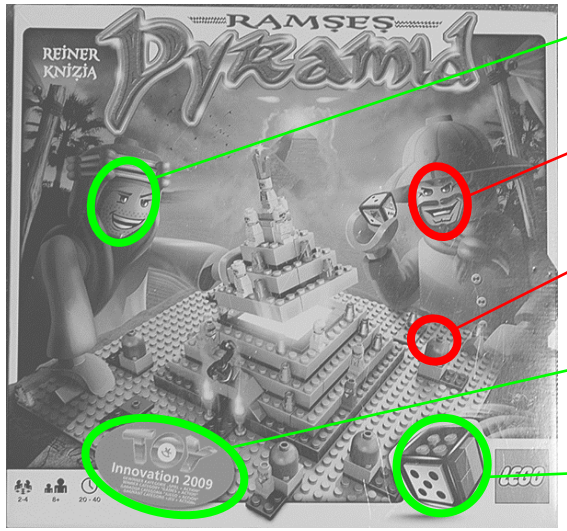
.2	.1	.8	.2	.3
.4	.1	.1	.2	.3
.4	.1	.4	.2	.1
.1	.7	.8	.8	.6
.2	.3	.8	.0	.9

Query Image



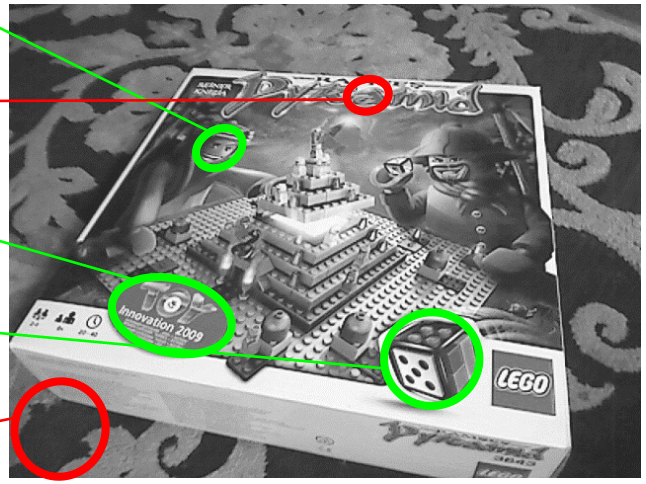
Feature Matching

Reference Image



.2	.1	.8	.2	.3	Correct Match	.2	.1	.8	.2	.3
.4	.1	.1	.2	.3	Incorrect Match	.4	.1	.1	.2	.3
.2	.3	.8	.0	.9		.4	.1	.4	.2	.1
.4	.1	.4	.2	.1		.1	.7	.8	.8	.6
.1	.7	.8	.8	.6		.2	.3	.8	.0	.9

Query Image



Robust Estimation: RANSAC

Robust Estimation: RANSAC

Robust Estimation: RANSAC

Mastering OpenCV with Practical Computer Vision Projects

Example: Time Trip Toralla

https://xoia.es/

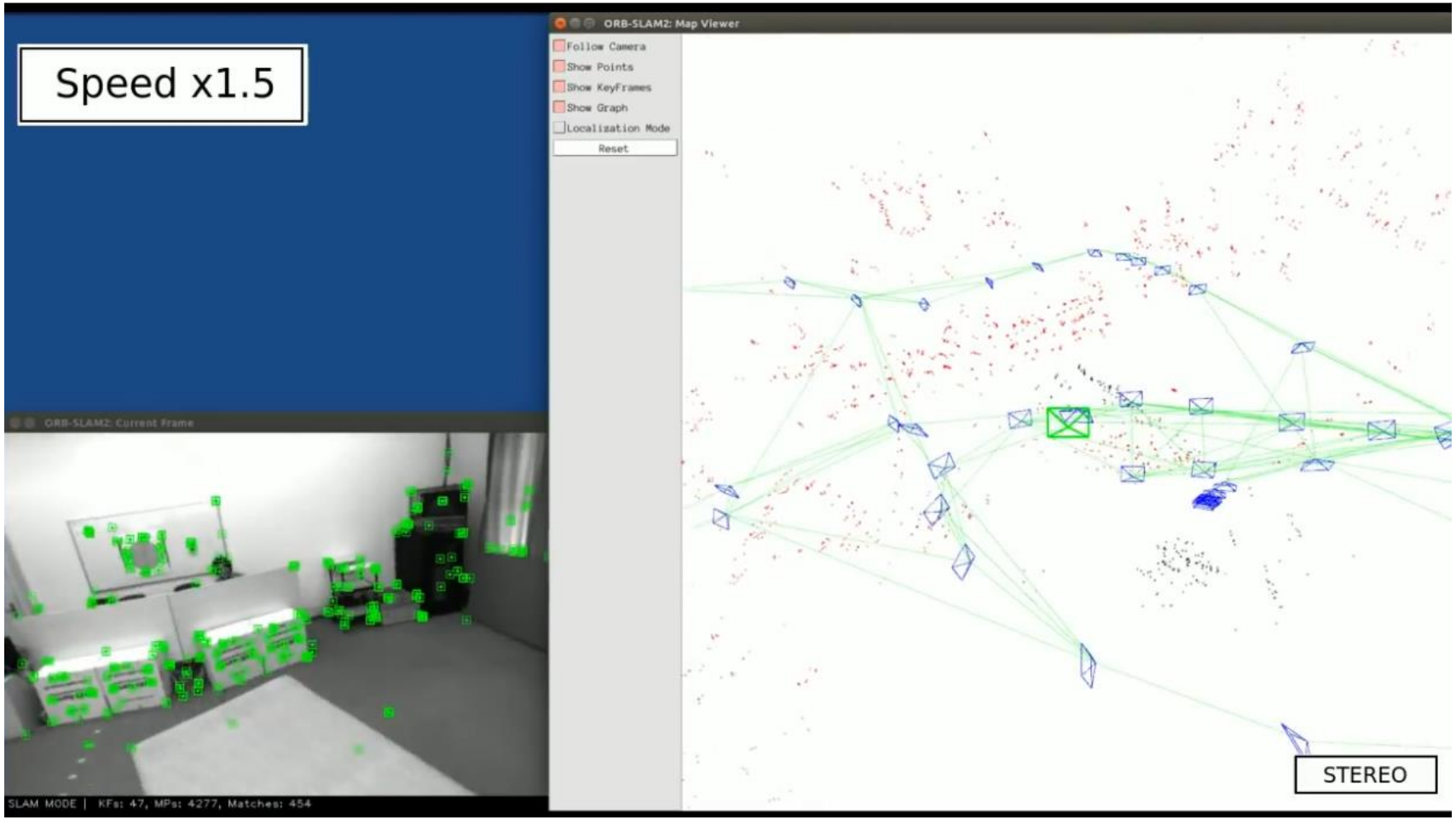
Let’s review three different examples

QR / ARUCO

Marker-less Target

Real life (SLAM)

SLAM: Simultaneous Localization And Mapping

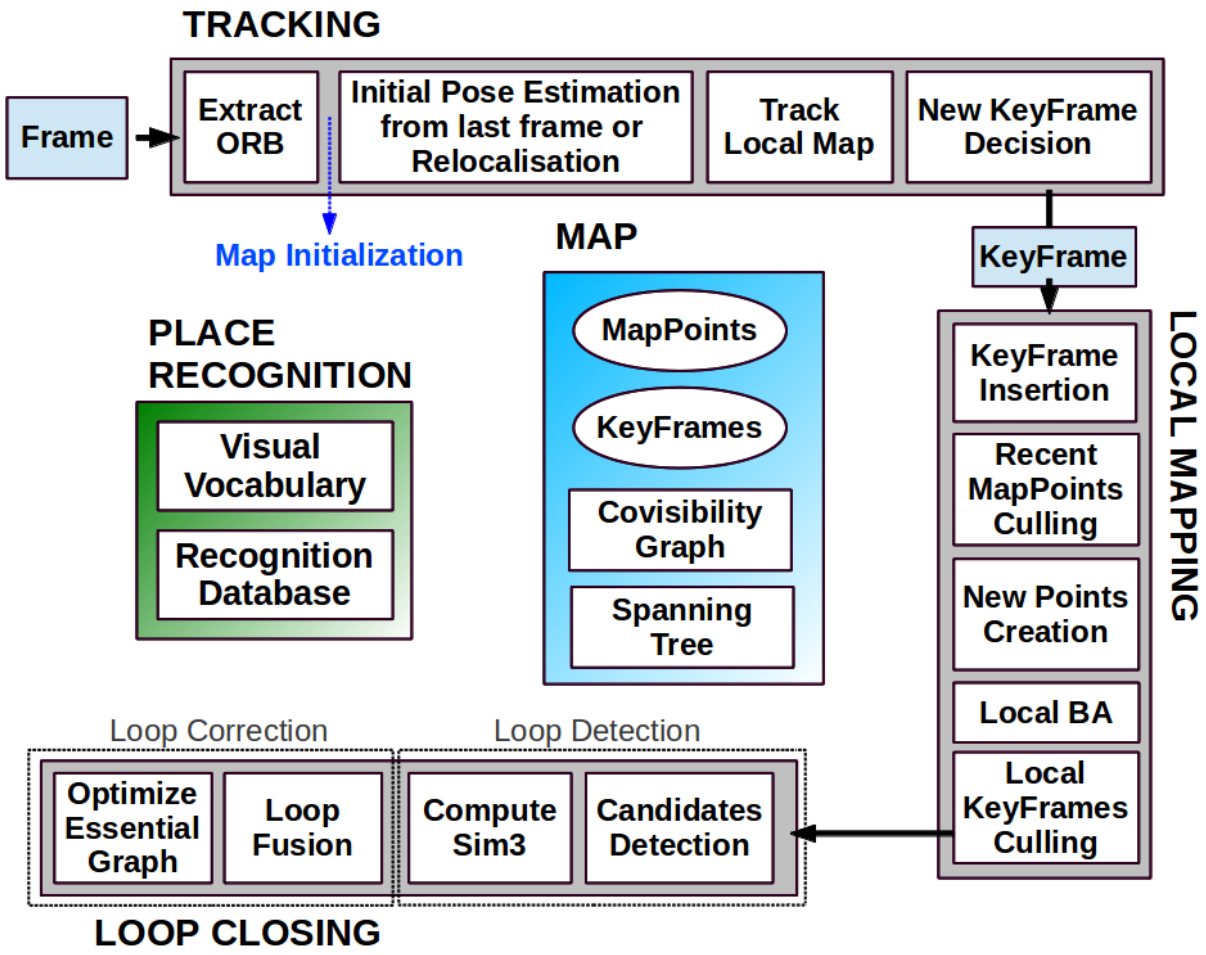


https://github.com/UZ-SLAMLab/ORB_SLAM3

SLAM: Simultaneous Localization And Mapping

There is three main components:

- Tracking
- Mapping
- Loop Closing



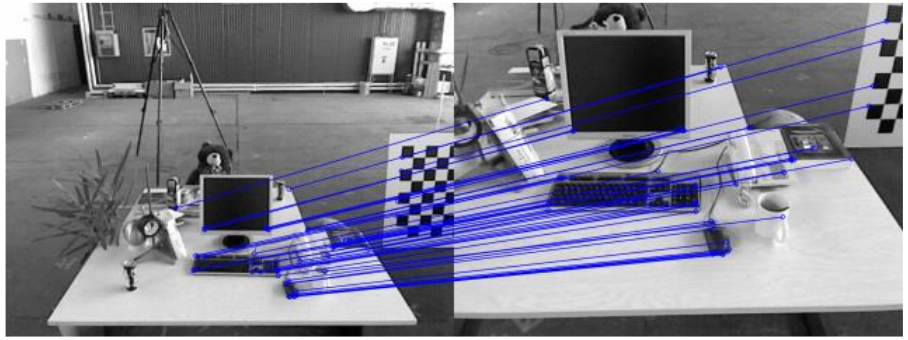
https://github.com/UZ-SLAMLab/ORB_SLAM3

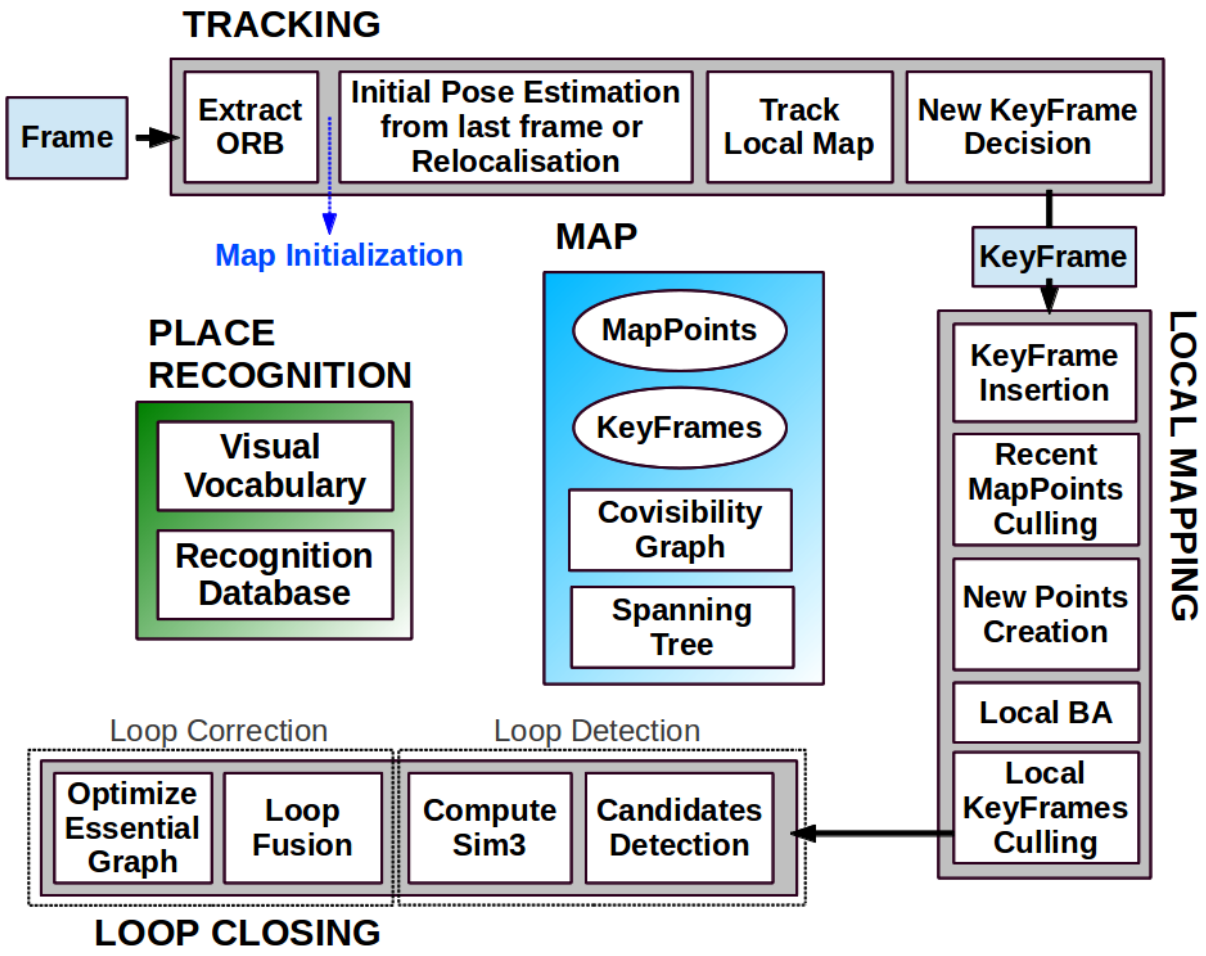
SLAM: Simultaneous Localization And Mapping

There is three main components:

- Tracking
- Mapping
- Loop Closing

In all steps we can use local descriptors from previous section!





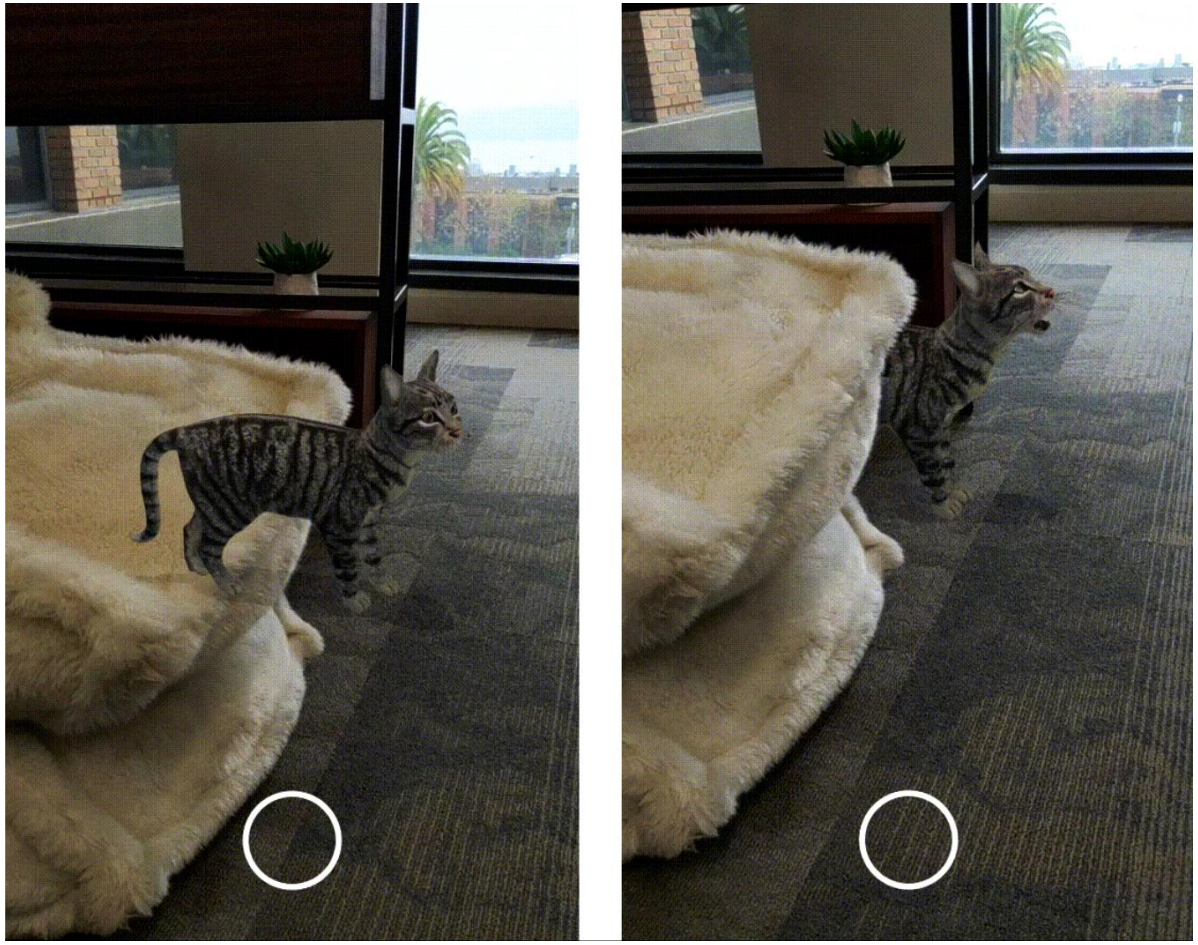
https://github.com/UZ-SLAMLab/ORB_SLAM3

Modern SLAM: Example NICE-SLAM

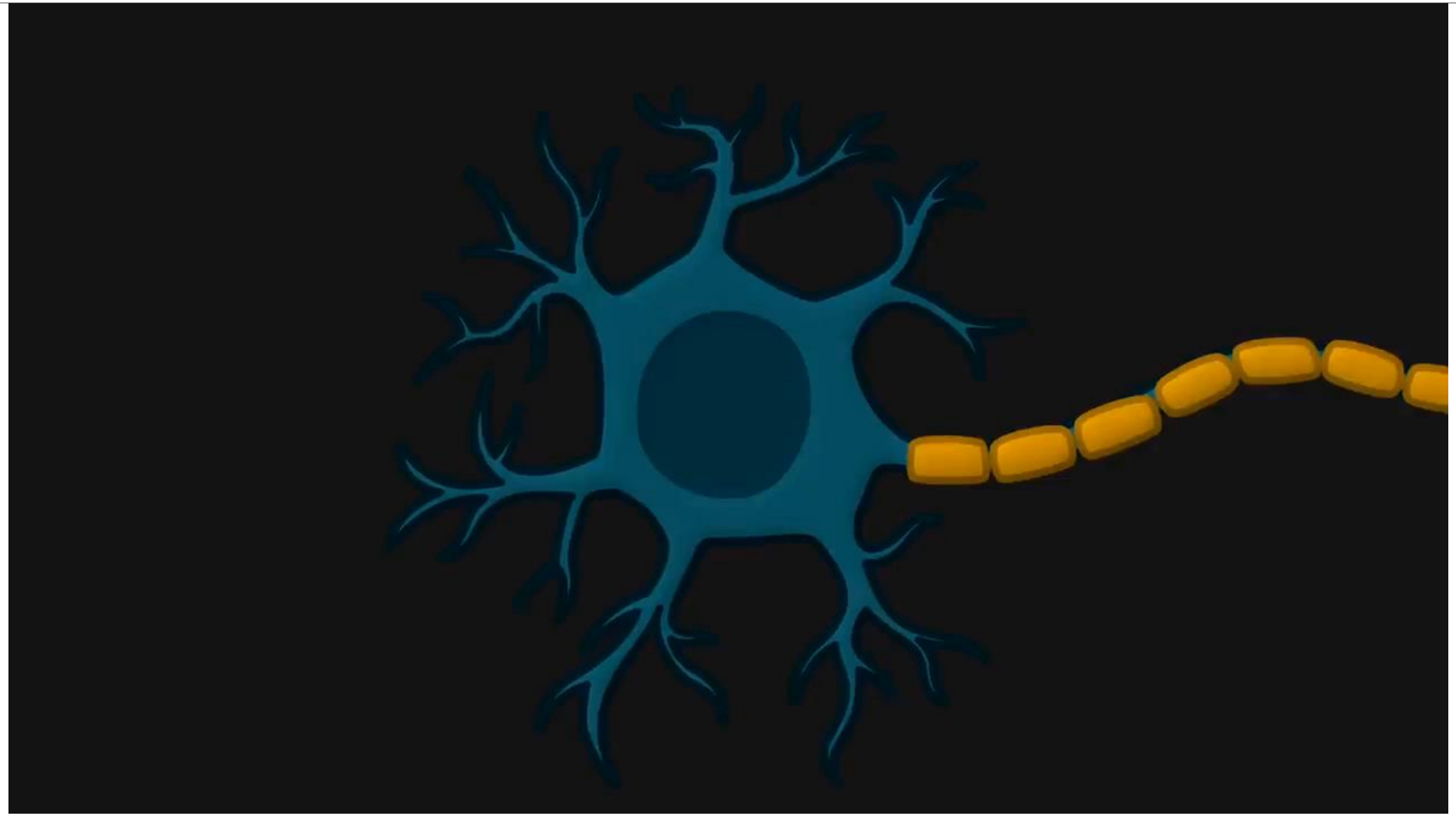
https://github.com/cvg/nice-slam

NICE-SLAM: Neural Implicit Scalable Encoding for SLAM

The Importance of Good Depths

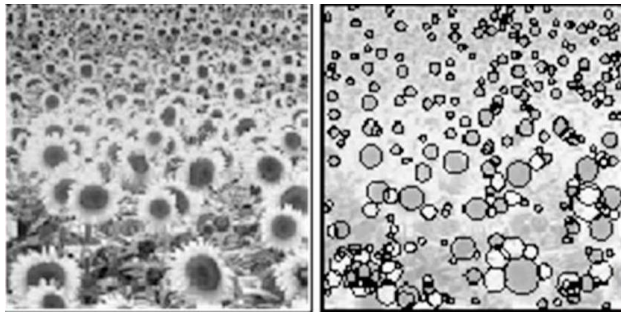


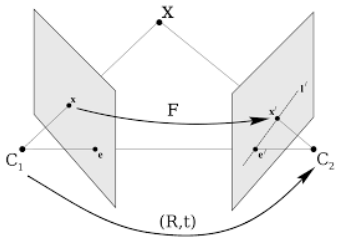
Neural Networks in Computer Vision



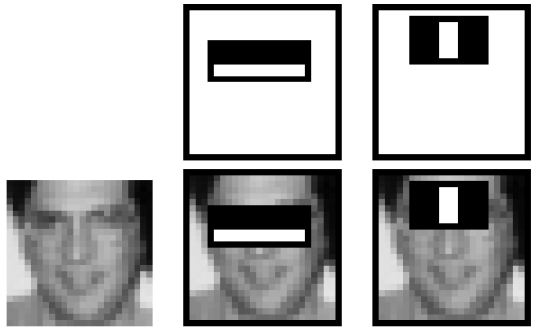
Computer Vision Evolution

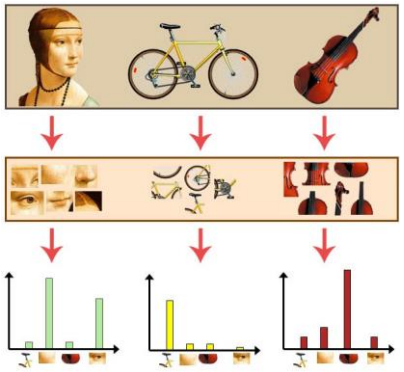
- **90's**: Geometry and low-level vision. Signal processing



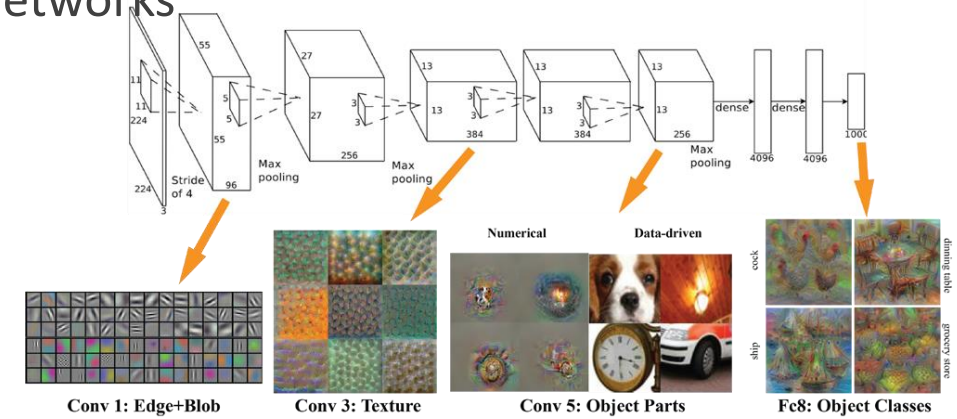


- **00's**: Machine Learning success: Boosting, Bag of Words

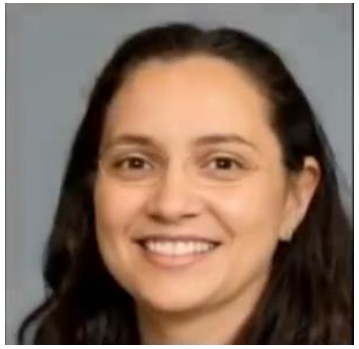




- **10's**: Deep Learning and Convolutional Neural Networks

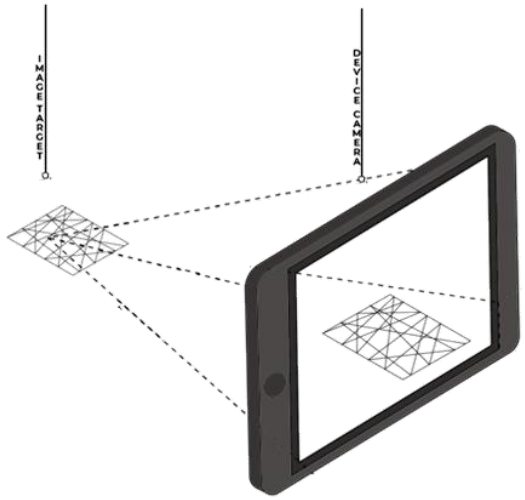


- **20's**: Neural Rendering, GANs, Graph Nets

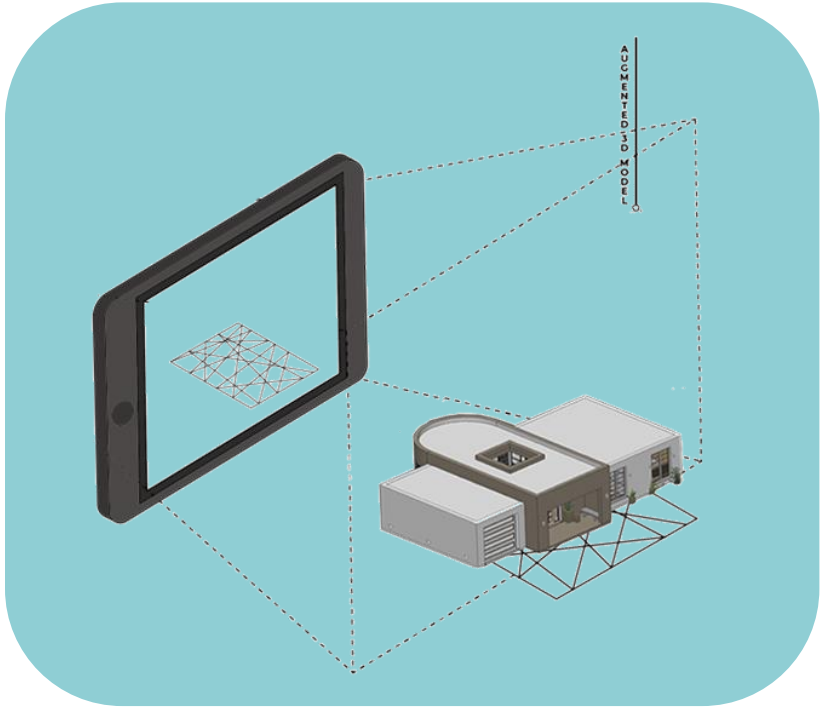


Example: Snapdragon Spaces

How does it work?



1. Computer Vision



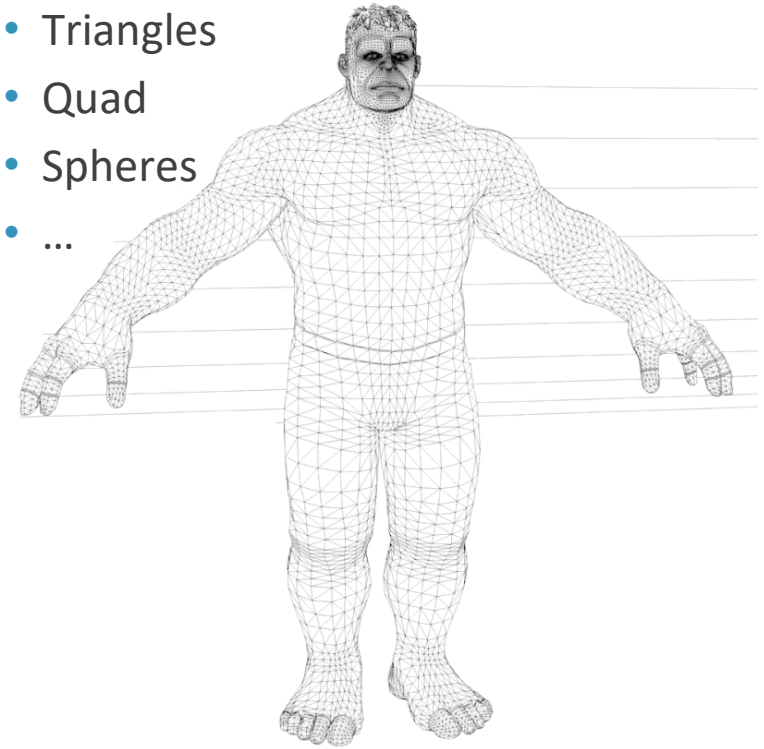
2. Computer Graphics

Computer Graphics

https://www.youtube.com/embed/MnQLiZSYZxM

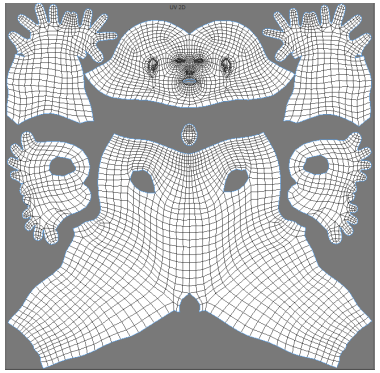
How do we represent objects?

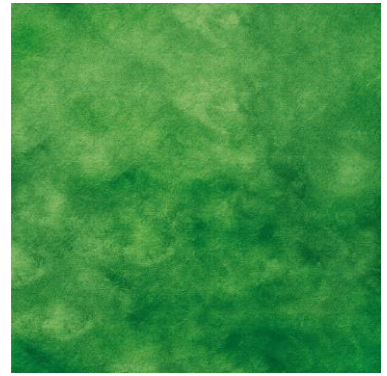
- **Geometry**
  - Triangles
  - Quad
  - Spheres
  - …



.obj, .stl, .step, …

- **Appearance**
  - UV-Map
  - Texture (color)



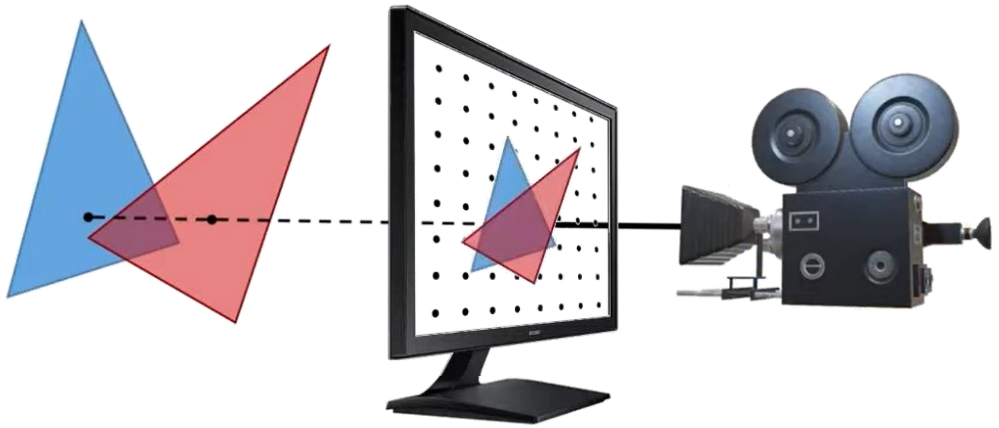


.mtl, .png, .jpg, .tiff, …

Computer Graphics: Geometry

Ray Tracing

For each pixel sample find the closest primitive



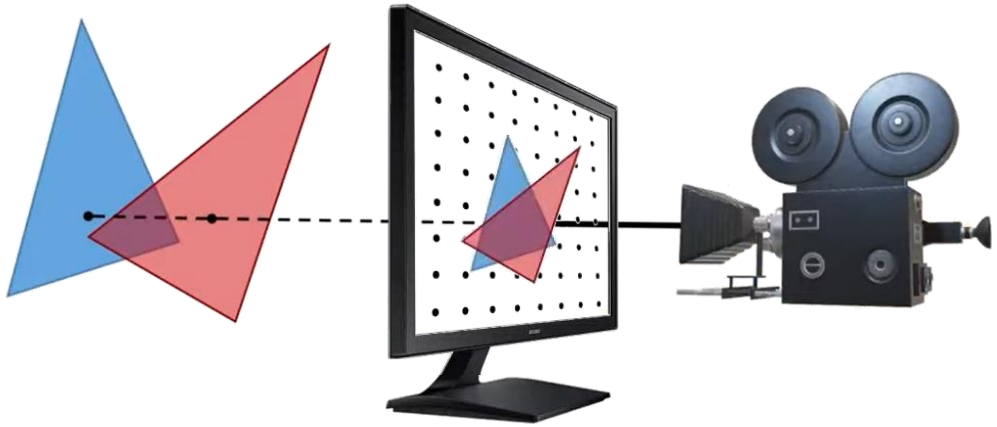
Very simple ☺

Very slow ☹

Computer Graphics: Geometry

Ray Tracing

For each pixel sample find the closest primitive

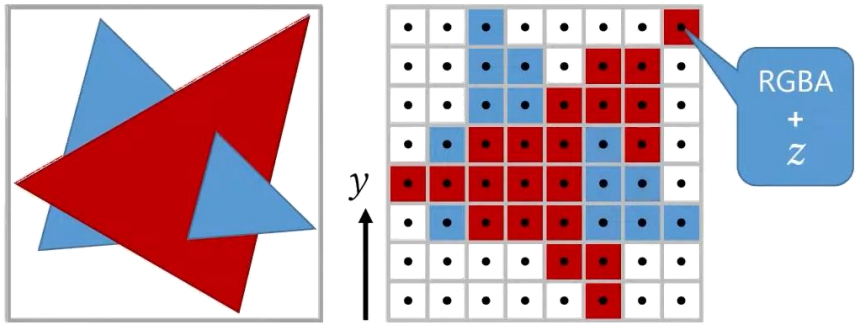


Very simple ☺

Very slow ☹

Rasterization (Rendering Pipeline)

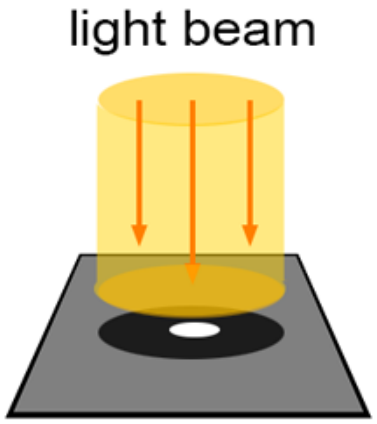
For each primitive find pixel samples

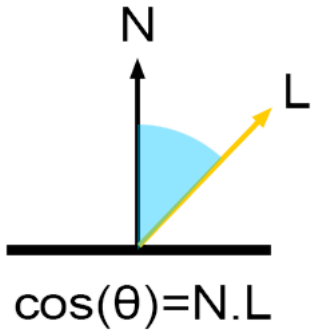


Very fast ☺

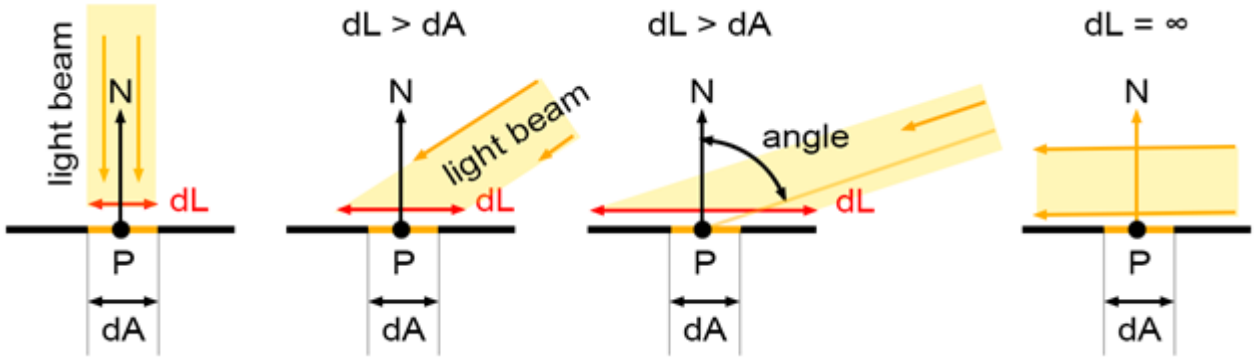
Complex

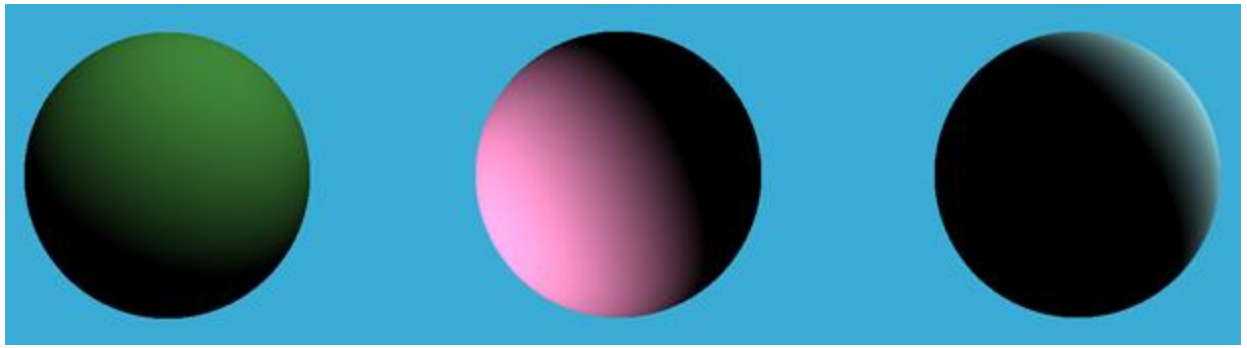
Lambertian Model





Lambert's Cosine Law



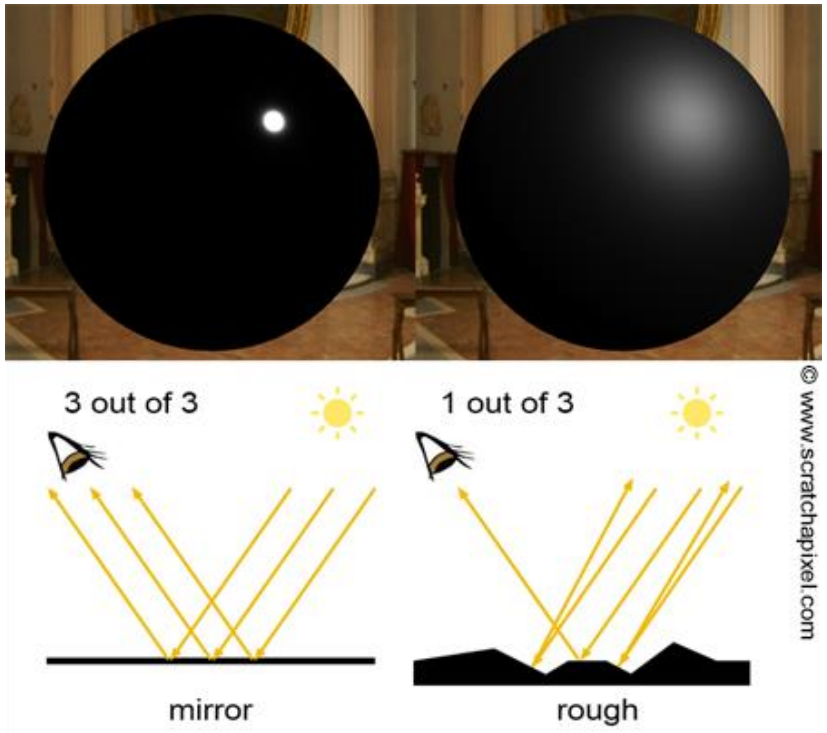


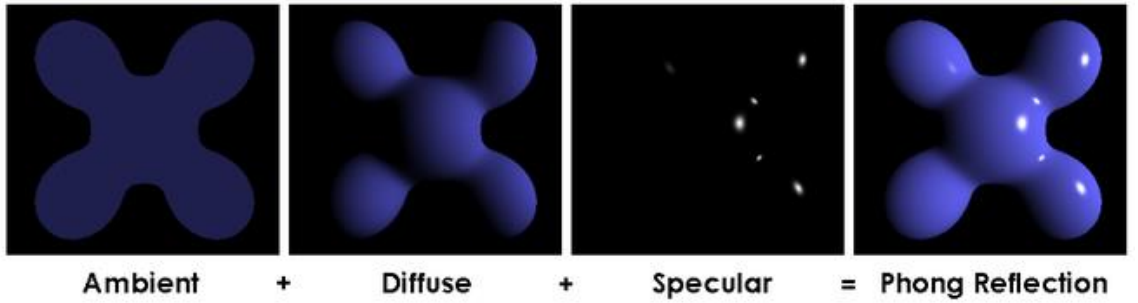
Some examples…

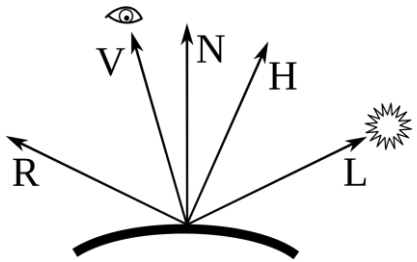
https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/diffuse-lambertian-shading.html

Phong Model

Specular Material Diffuse Material







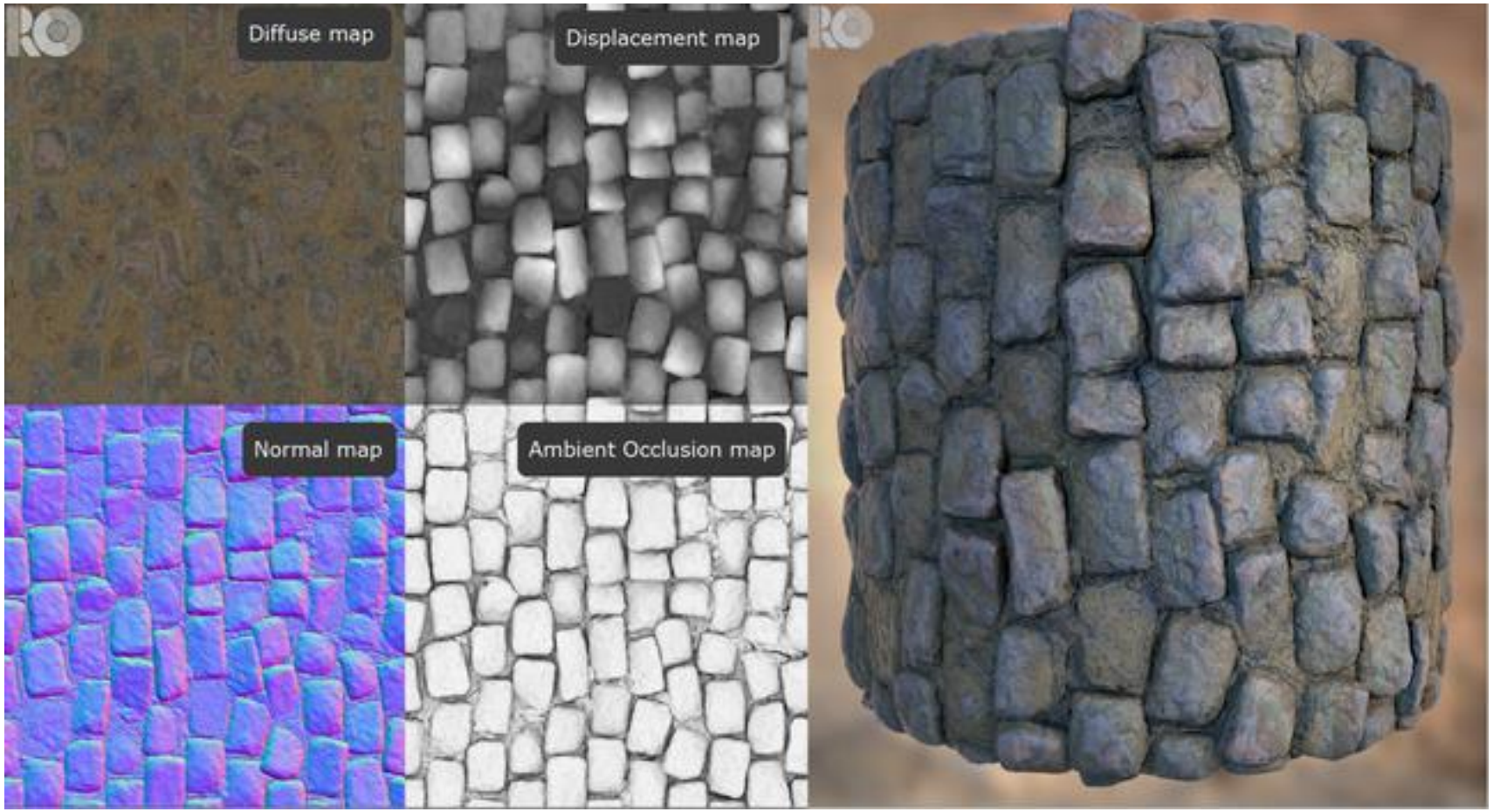
$$I_{\mathrm{p}} = k_{\mathrm{a}} i_{\mathrm{a}} + \sum_{m \in \text{lights}} \left(k_{\mathrm{d}} (\hat{L}_m \cdot \hat{N}) i_{m,\mathrm{d}} + k_{\mathrm{s}} (\hat{R}_m \cdot \hat{V})^{\alpha} i_{m,\mathrm{s}} \right).$$

https://www.scratchapixel.com/lessons/3d-basic-rendering/phong-shader-BRDF/phong-illumination-models-brdf.html

https://www.artstation.com/channels/textures_materials
Tutorial Displacement and Normal Maps

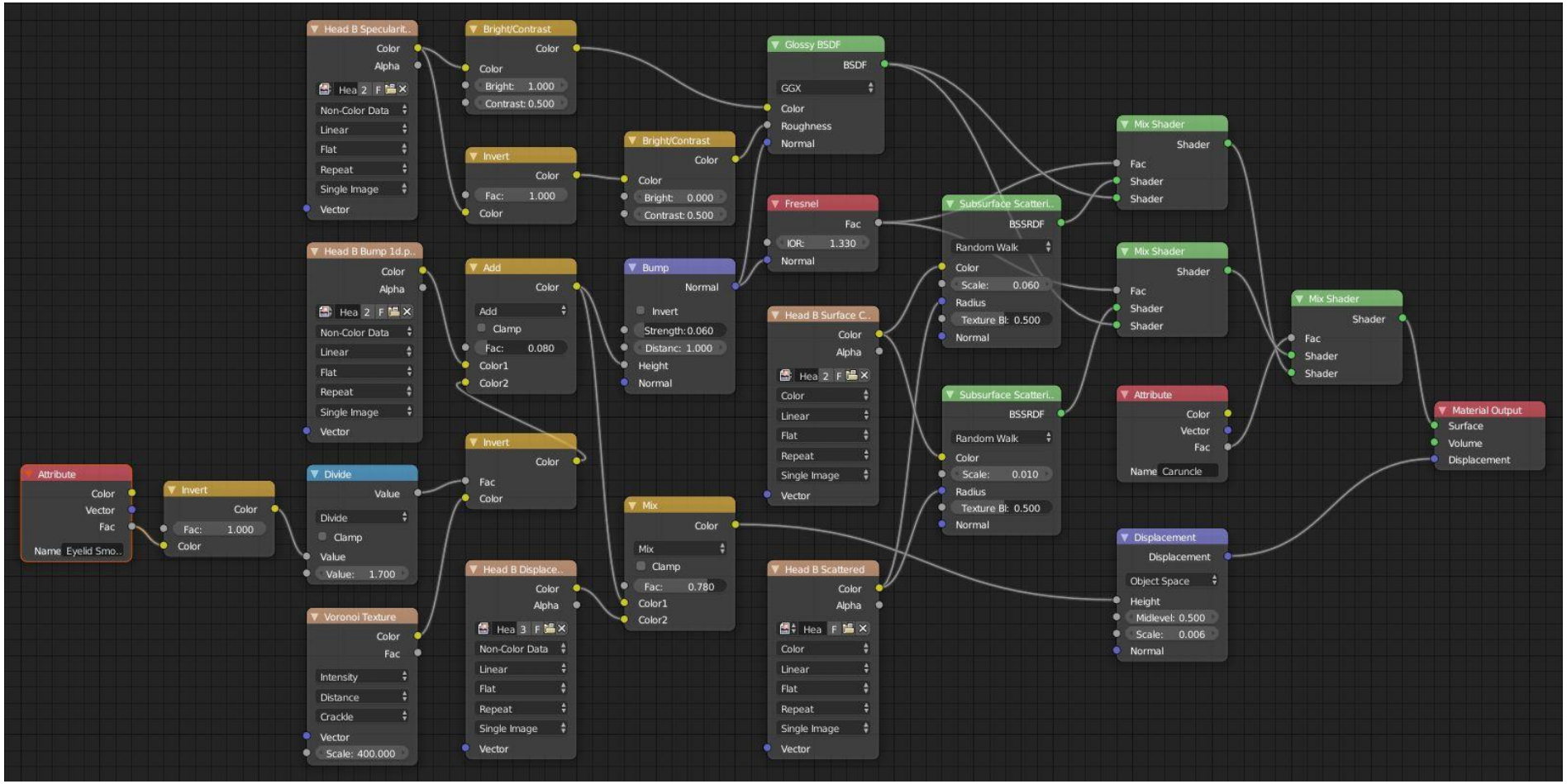
Computer Graphics: Materials

There are many ways to represent materials. Let's keep simple:

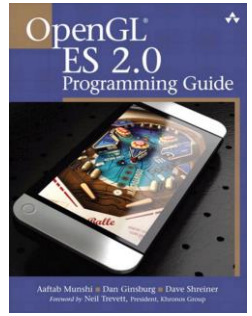


Computer Graphics: Materials

Materials can also be very complicated:



Shaders



Small programs that run in the GPU. OpenGL Shading Language (GLSL)

- **Vertex Shader** -> Executed once per (triangle) vertex

```
attribute vec4 vertexPosition;  // Describes the position of the vertex in the object reference sys.

uniform mat4 modelMatrix;       // Describes the position of the object
uniform mat4 viewMatrix;        // Describes the position of the camera (extrinsics)
uniform mat4 projectionMatrix;  // Describes the intrinsic camera properties

varying vec4 v_Color;           // This will be passed into the fragment shader.

void main()
{
    gl_Position = projectionMatrix * viewMatrix * modelMatrix * vertexPosition;
}
```

- **Fragment Shader** -> Executed once per pixel

```
varying vec4 v_Color;          // This is the color from the vertex shader interpolated across the
                               // triangle per fragment.

// The entry point for our fragment shader.
void main()
{
    gl_FragColor = v_Color;    // Pass the color directly through the pipeline.
}
```

Computer Graphics

Interactive Computer Graphics
University of Utah

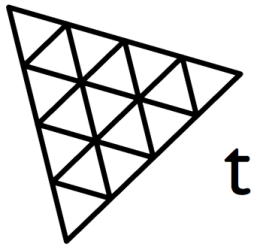
https://www.youtube.com/playlist?list=PLplnkTzzqsZS3R5DjmCQsqupu43oS9CFN

6.837: Introduction to Computer Graphics
MIT

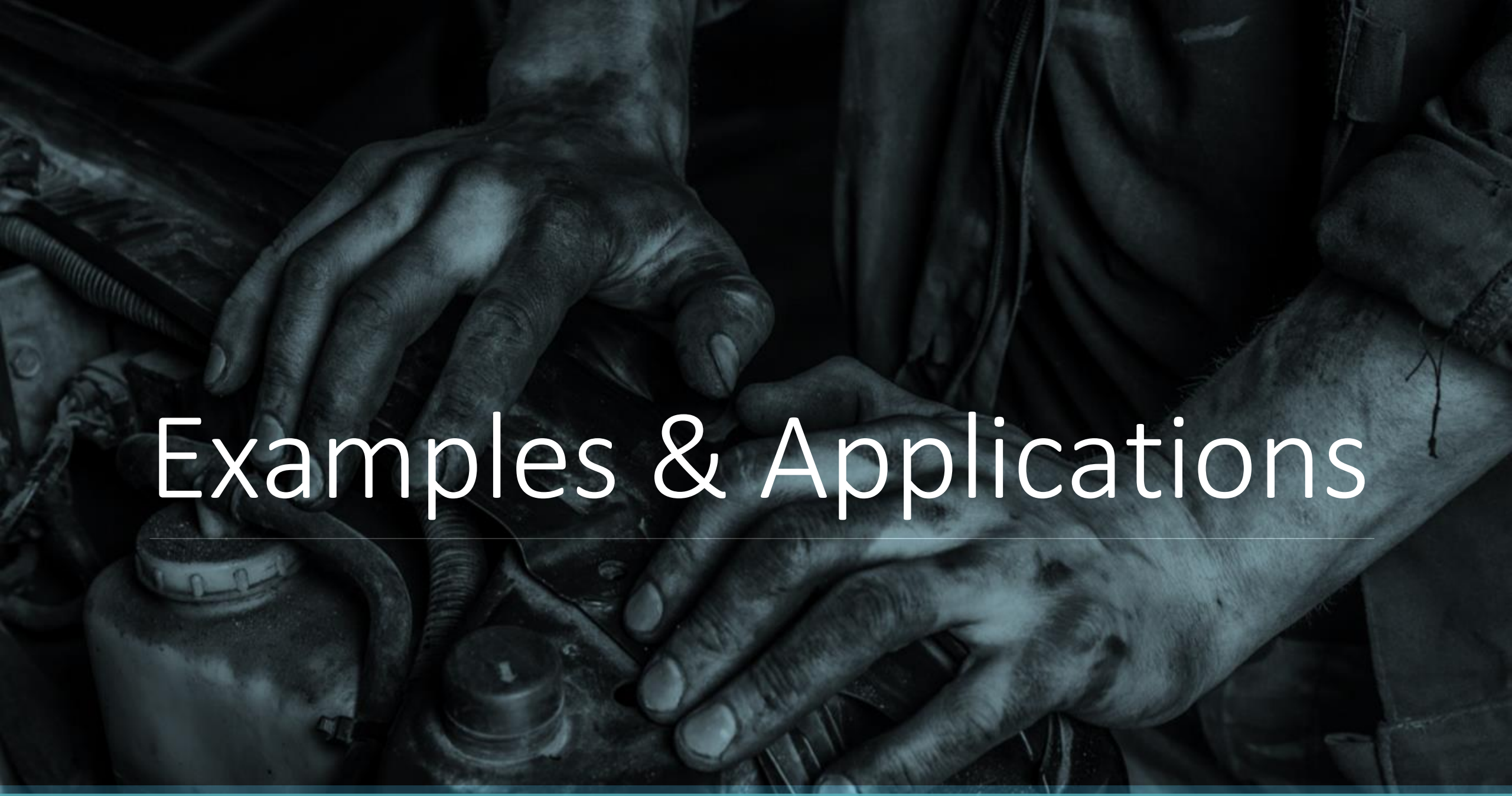
https://www.youtube.com/playlist?list=PLQ3UicqQtfNuBjzJ-KEWmG1yjiRMXYKhh

Useful Tools

UNREAL
ENGINE

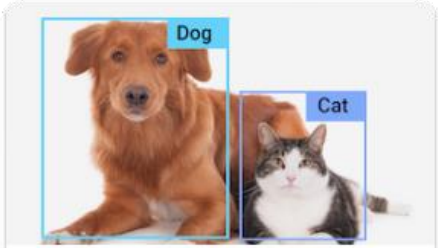


three.js



Examples & Applications

Google Mediapipe



Object Detection
Track and label objects in images.

See demo Customize

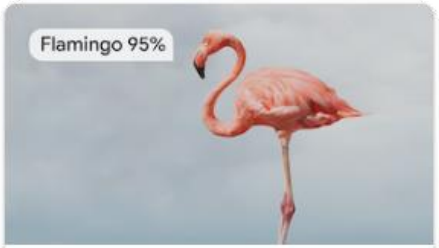


Image Classification
Identify content in images.

See demo Customize

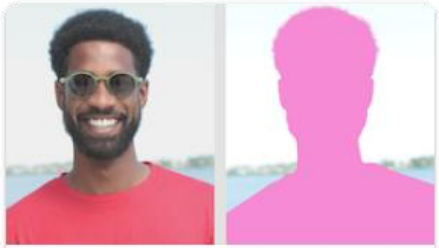
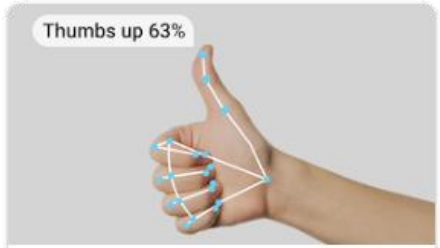


Image Segmentation
Locate objects and create image masks with labels.

See demo

Interactive Segmentation
Segment the object of interest in an image.

See demo



Gesture Recognition
Identify and recognize hand gestures.

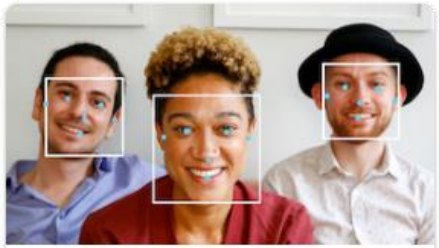
See demo Customize

Hand Landmark Detection
Detect hand landmarks.

See demo

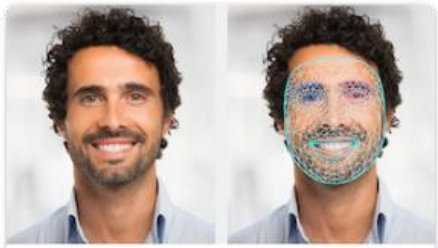
Image Embedding
Convert images into embedding vectors.

See demo



Face Detection
Detect faces in real time.

See demo



Face Landmark Detection
Detect face landmarks and blendshape scores in real time.

See demo

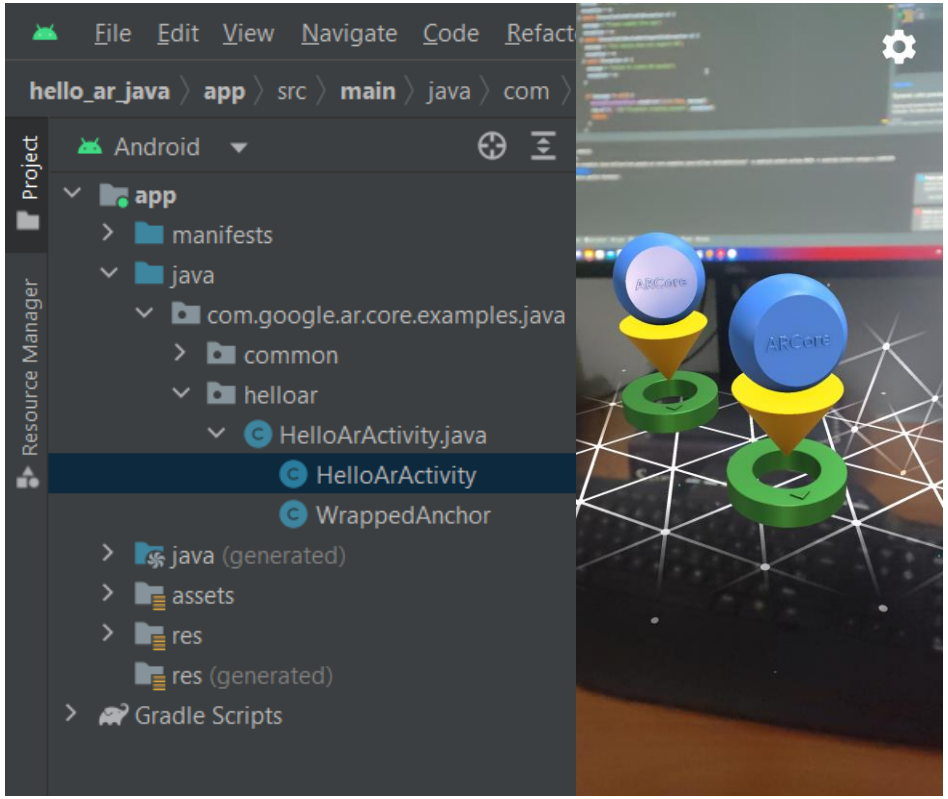
Pose Landmark Detection
Identify key points on the body in real time.

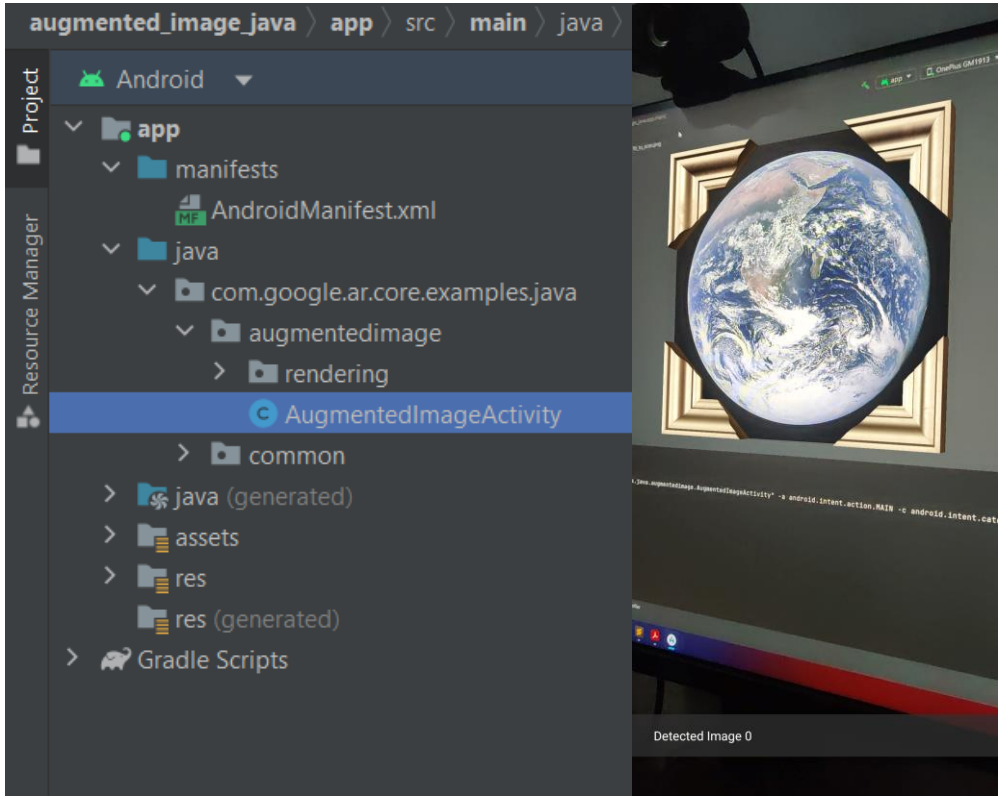
See demo

https://mediapipe-studio.webapps.google.com/demo/face_landmarker

https://codepen.io/Babich/pen/VwXrjvK

AR Core Examples



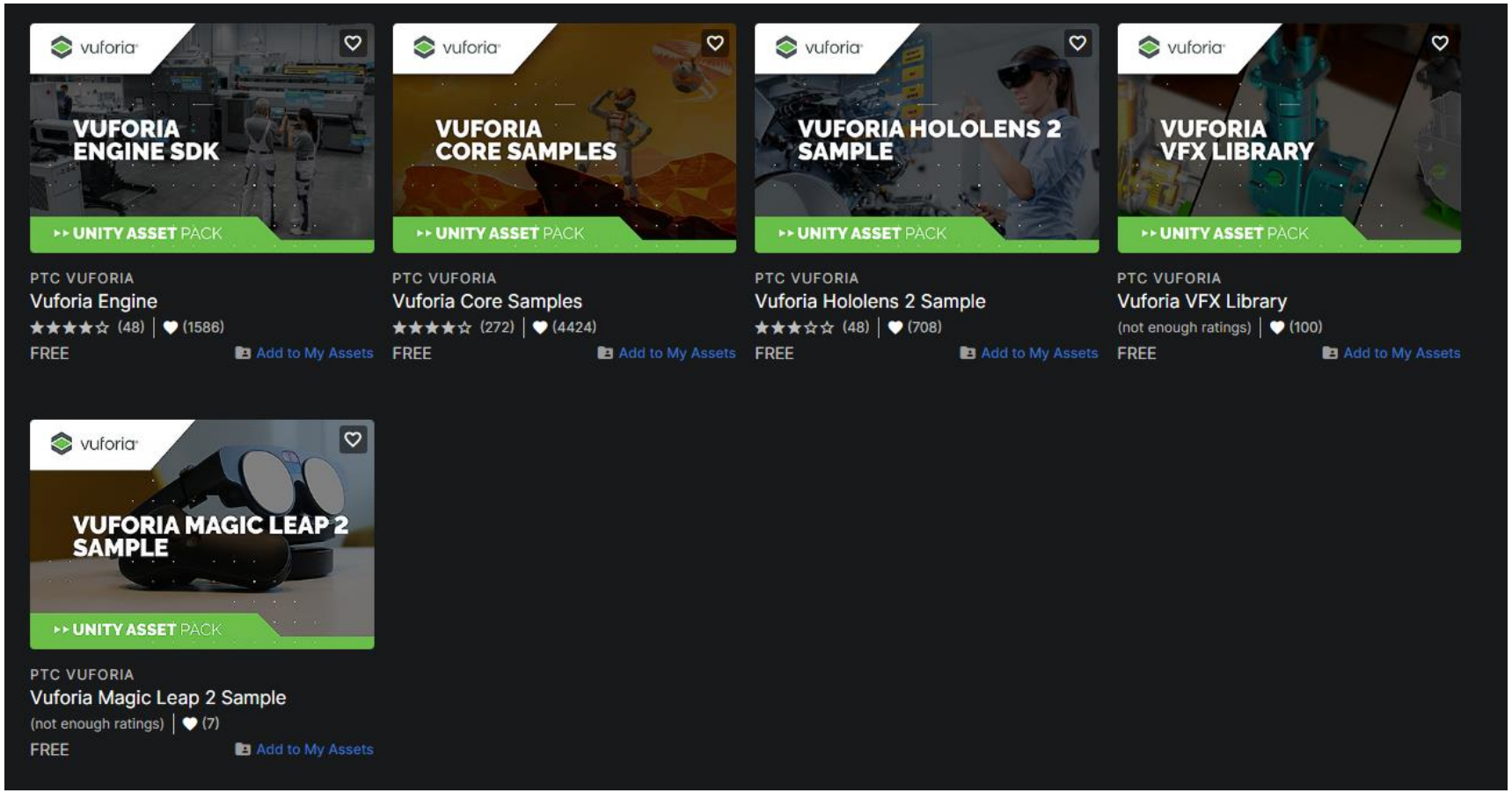


https://github.com/google-ar/arcore-android-sdk

https://developers.google.com/ar/develop/java/quickstart

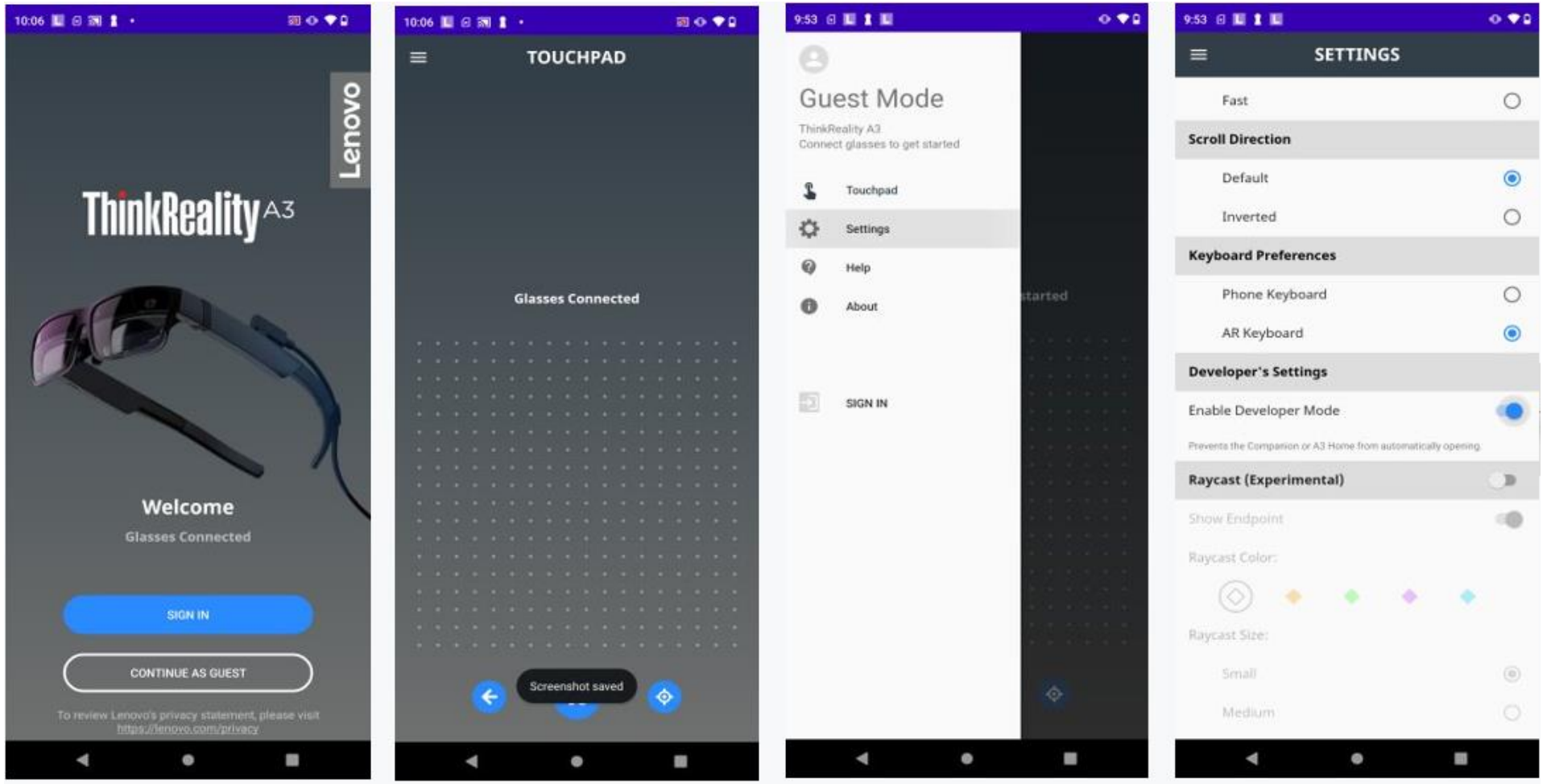
Vuforia Examples

- Model Targets
- Area Targets
- Ground Plane
- Image Targets
- VuMark
- Cylinder Targets
- Multi Targets
- Instant Image Targets
- Cloud Recognition
- Virtual Buttons



https://developer.vuforia.com/downloads/samples

Qualcomm Snapdragon Spaces



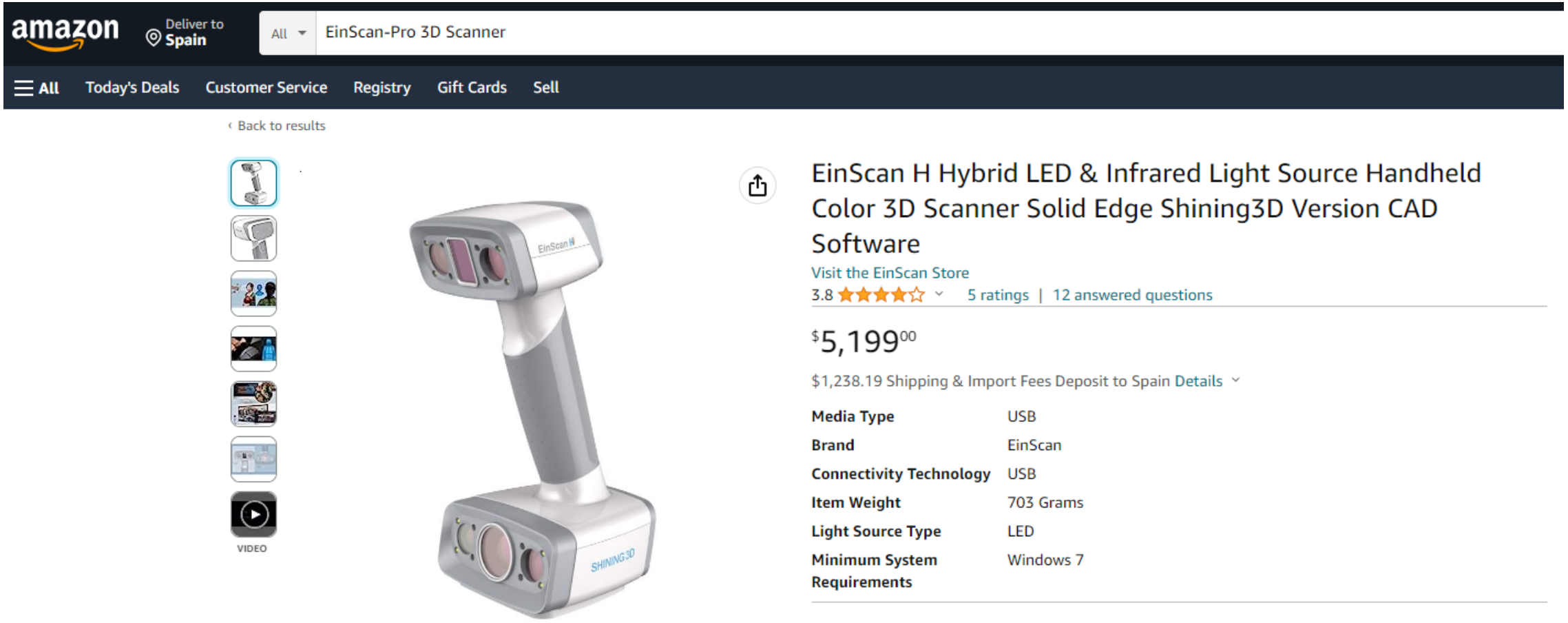
https://spaces.qualcomm.com/

Is XR a solved problem?

XR: Towards Spatial Intelligence

Embodiment provides new challenges. Example: **EPIC KITCHENS** https://epic-kitchens.github.io/2023

XR: Towards Spatial Intelligence



Simple and Realistic Object Capture

https://developer.apple.com/augmented-reality/object-capture/

XR: Towards Spatial Intelligence

vMAP: Vectorised Object Mapping for Neural Field SLAM

Xin Kong Shikun Liu Marwan Taher Andrew Davison

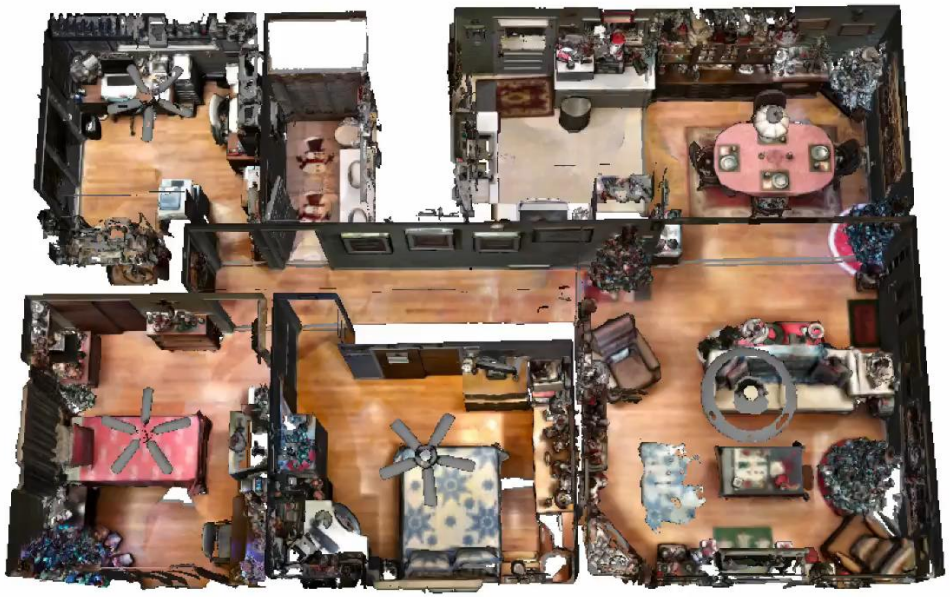
CVPR 2023

3D object detection in real-time SLAM

https://kxhit.github.io/vMAP

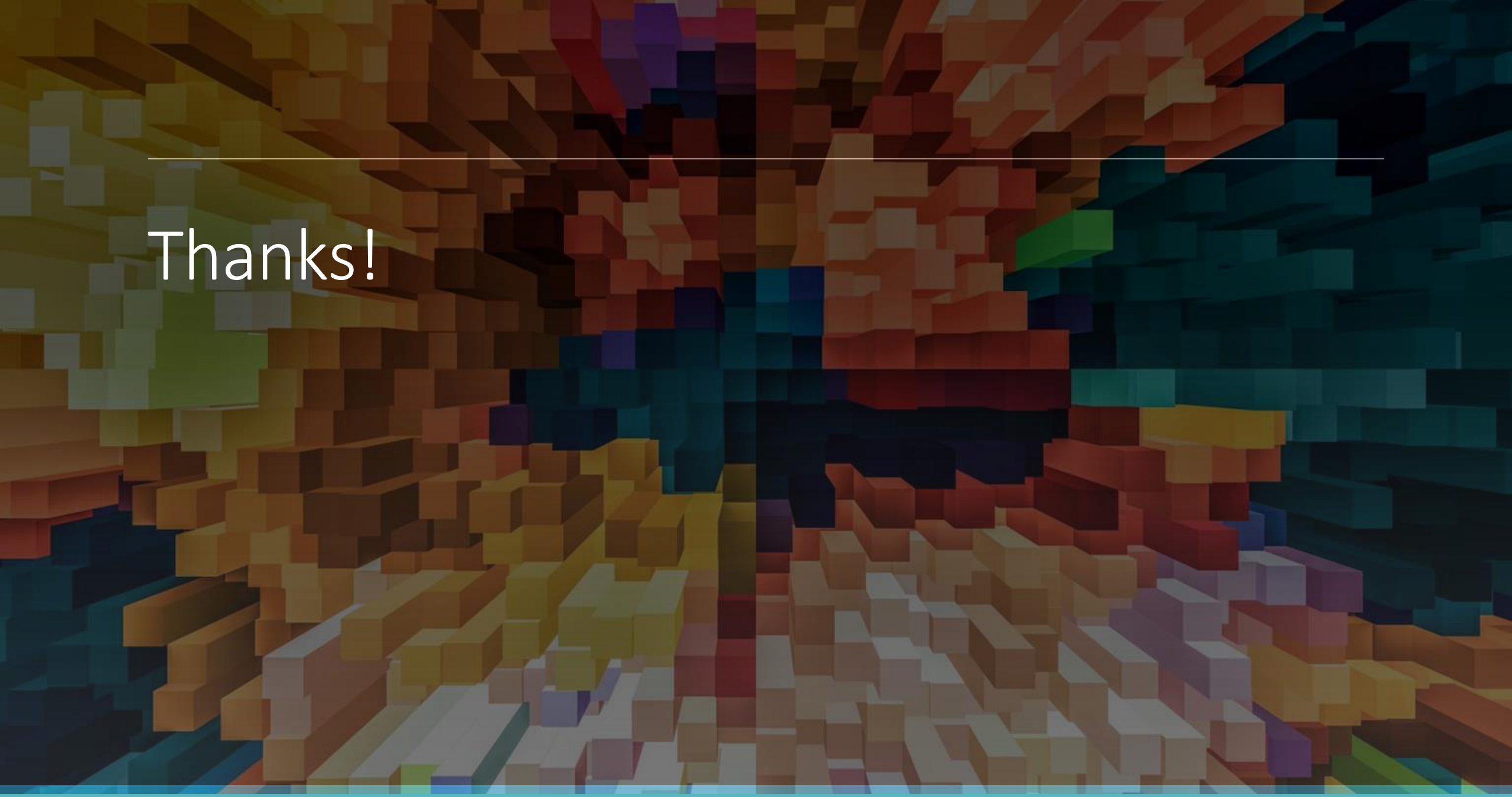
XR: Towards Spatial Intelligence

Text queries:



Generalizing to new concepts: OpenScene

https://pengsongyou.github.io/openscene


Thanks!