MULTI OBJECT 6D POSE ESTIMATION FOR AR-ASSISTANCE TASKS

SVCP 2021, 26.07.2021 – 27.07.2021

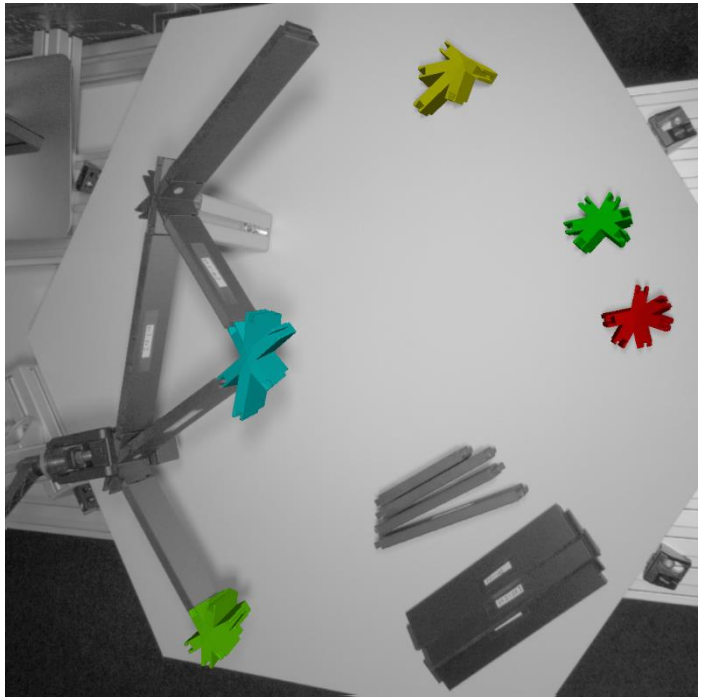
Niklas Gard

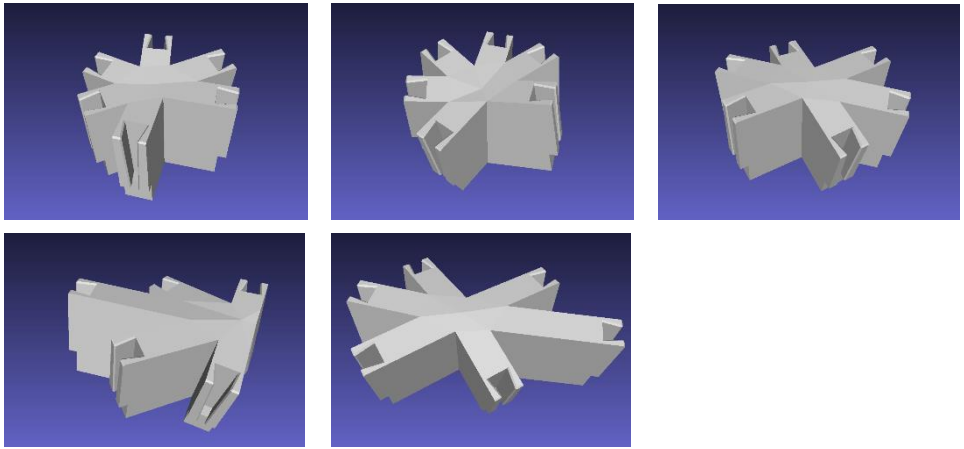
Agenda

- Motivation
- Deep 6D Object Detection and Registration
- Local Object Tracking
- AR-Applications

Motivation

Deep 6D object detection and registration

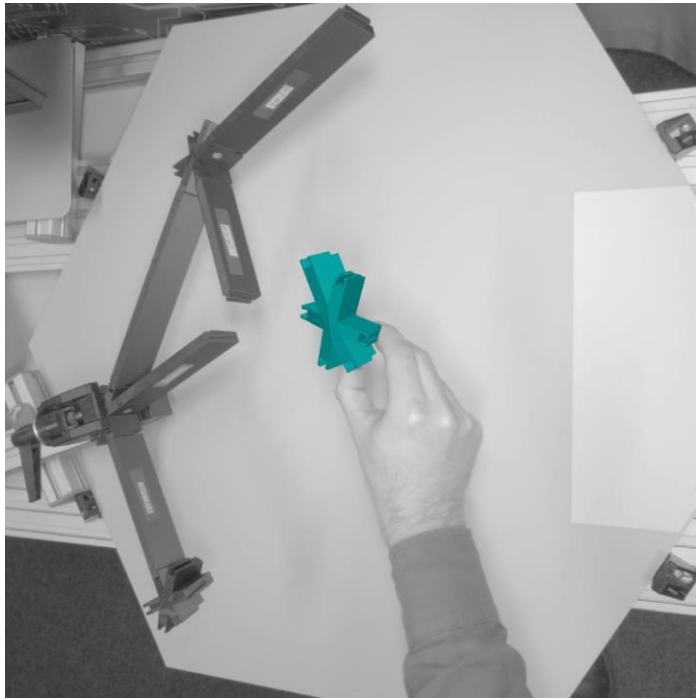


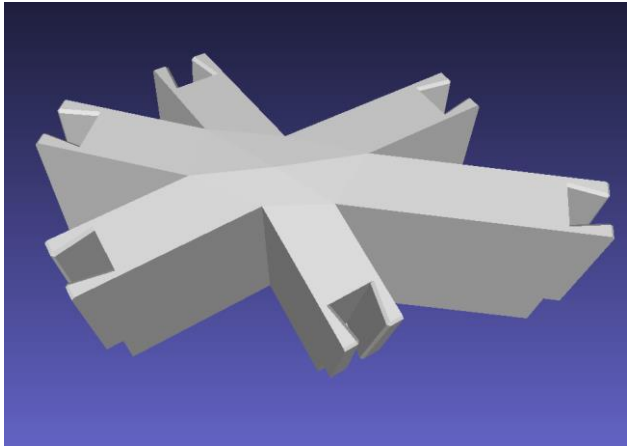


- Find pose of known 3D shapes with respect to the camera
- Pose: rotation and translation in 3D space

Motivation

Object tracking

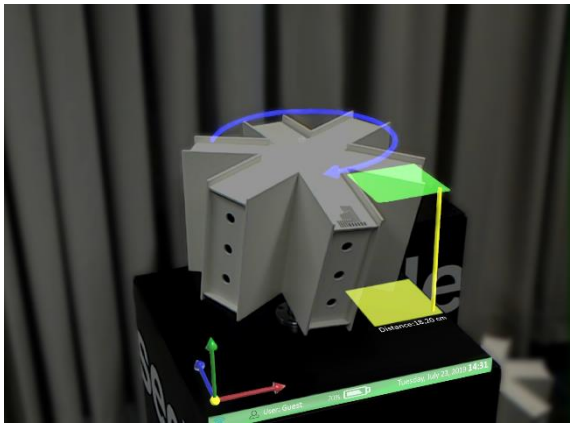




- Find pose of known 3D shapes with respect to the known pose from the last frame

Augmented assistance

AR assistance in facade construction



- Measurement of deviations from target 3D geometry
- Display **3D assistance information**

Fraunhofer HHI

Challenges

Detection

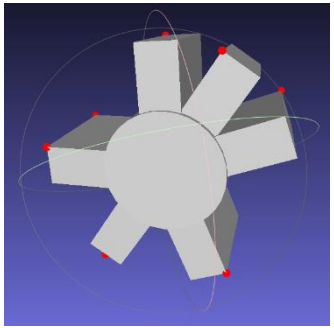
- geometrically similar untextured objects
- occlusions
- learning with synthetic data

Tracking

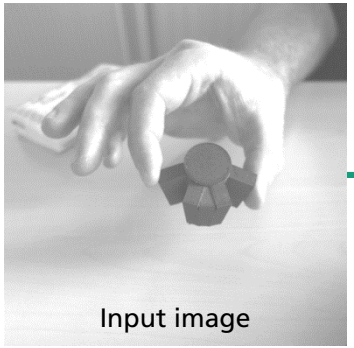
- real-time
- detect mismatches
- occlusions

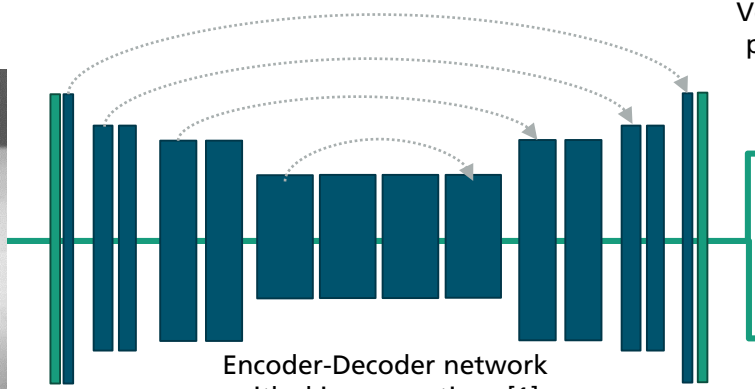
Deep 6D Object Detection and Registration

Neural network for pose estimation

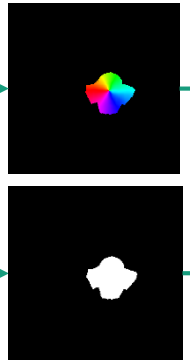


3D keypoints



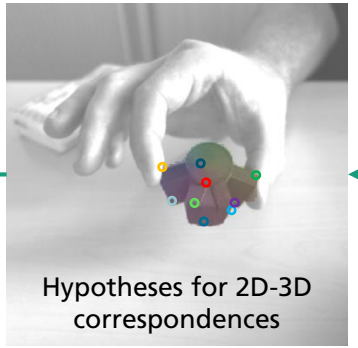


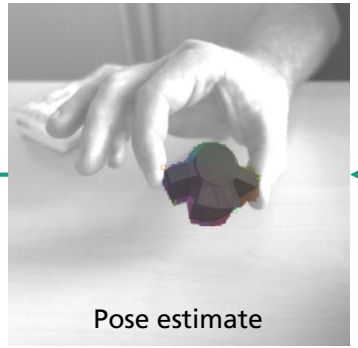
Vector-field
prediction

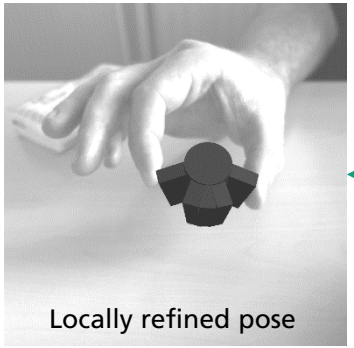


Semantic segmentation

RANSAC
Voting





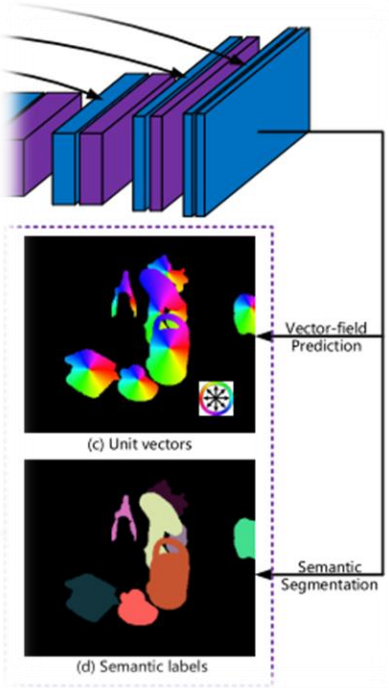


[1] PVNet: Pixel-wise Voting Network for 6DoF Pose Estimation Sida, Peng et al., 2019 CVPR

How to deal with multiple object classes?

How to deal with multiple object classes?

Problems of naive solution



Last layer splits the number of inputs to a large number of outputs

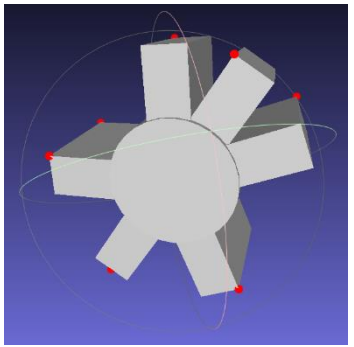
Outputs:

#objects × #keypoints × 2

+

#objects + 1

e.g. **248** output maps for 13 Objects with 9 Keypoints

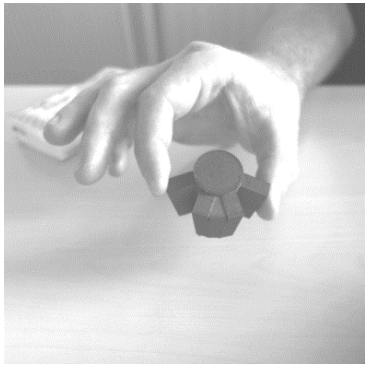


3D keypoints

- Difficult to train
  - Needs a lot of GPU memory in training
- **Decreasing** accuracys with every object

One model per object?

- Good accuracy and commonly used



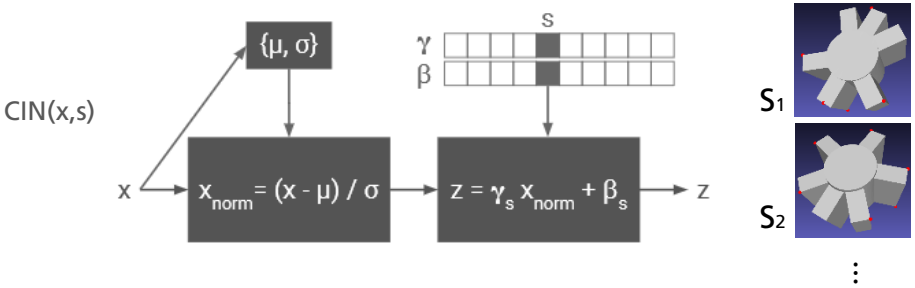
How to know which object is shown if I have 20 trained models?

- Extra 2D detector?
  - More training effort.
- Multiple inferences per image
- Similar objects?
  - All objects must be shown in every training of every model.
- Needs a lot of memory

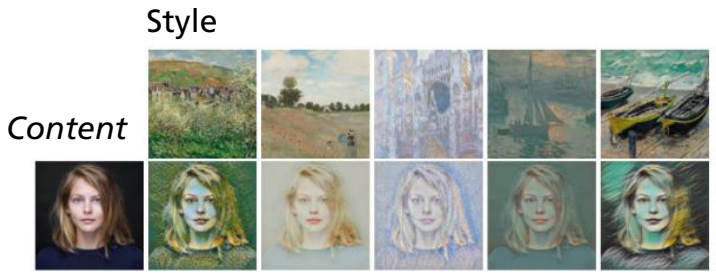
Object specific parametrization

Object as style

- Additional object specific parameters
  - (De)normalization of convolutional layer output depending of predefined output class



Style-transfer with **n** predefined styles



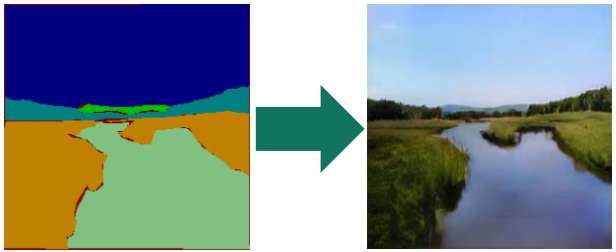
Conditional Instance Normalization (CIN) [1]

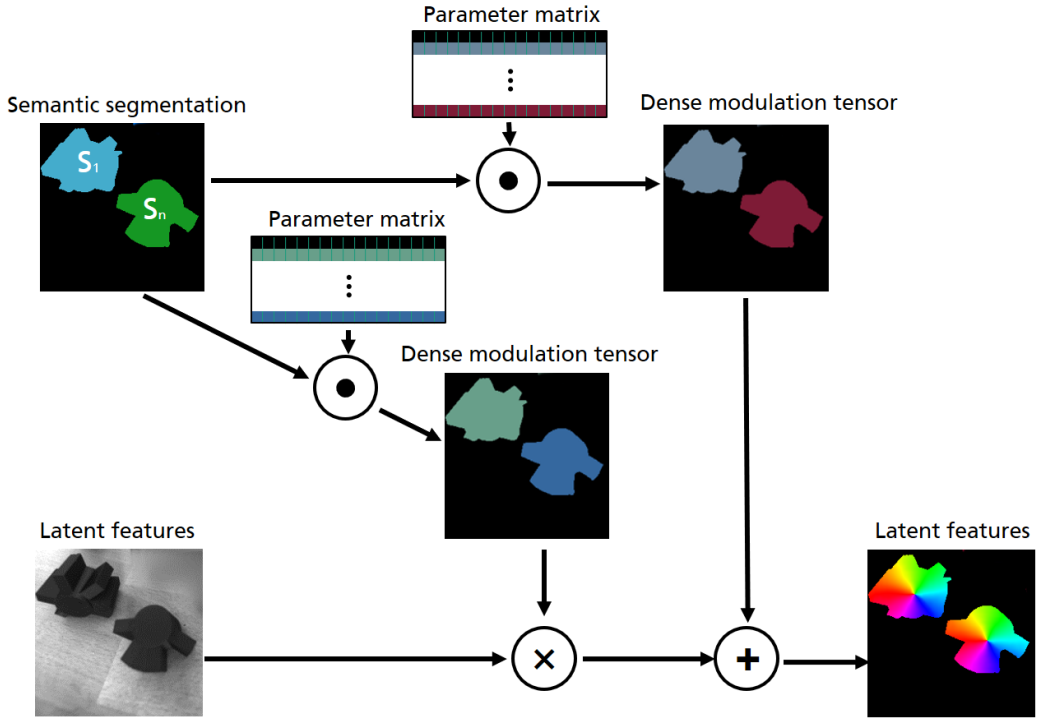
[1] Dumoulin, Vincent, Jonathon Shlens, and Manjunath Kudlur. "A learned representation for artistic style. 2016

Object specific parametrization

Class-adaptive (de)normalization

- Learnable affine transformation parameters dependent of pixel class
- Selection of row in parameter matrix dependent of pixel
- Idea from GAN based semantic image synthesis [1]



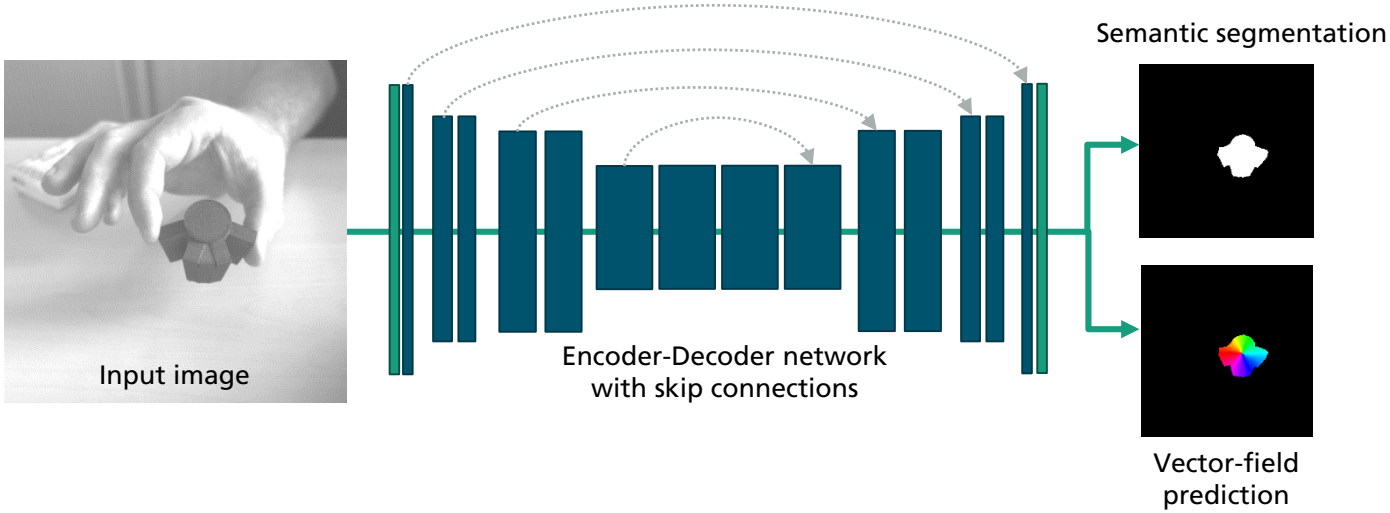


Niklas Gard

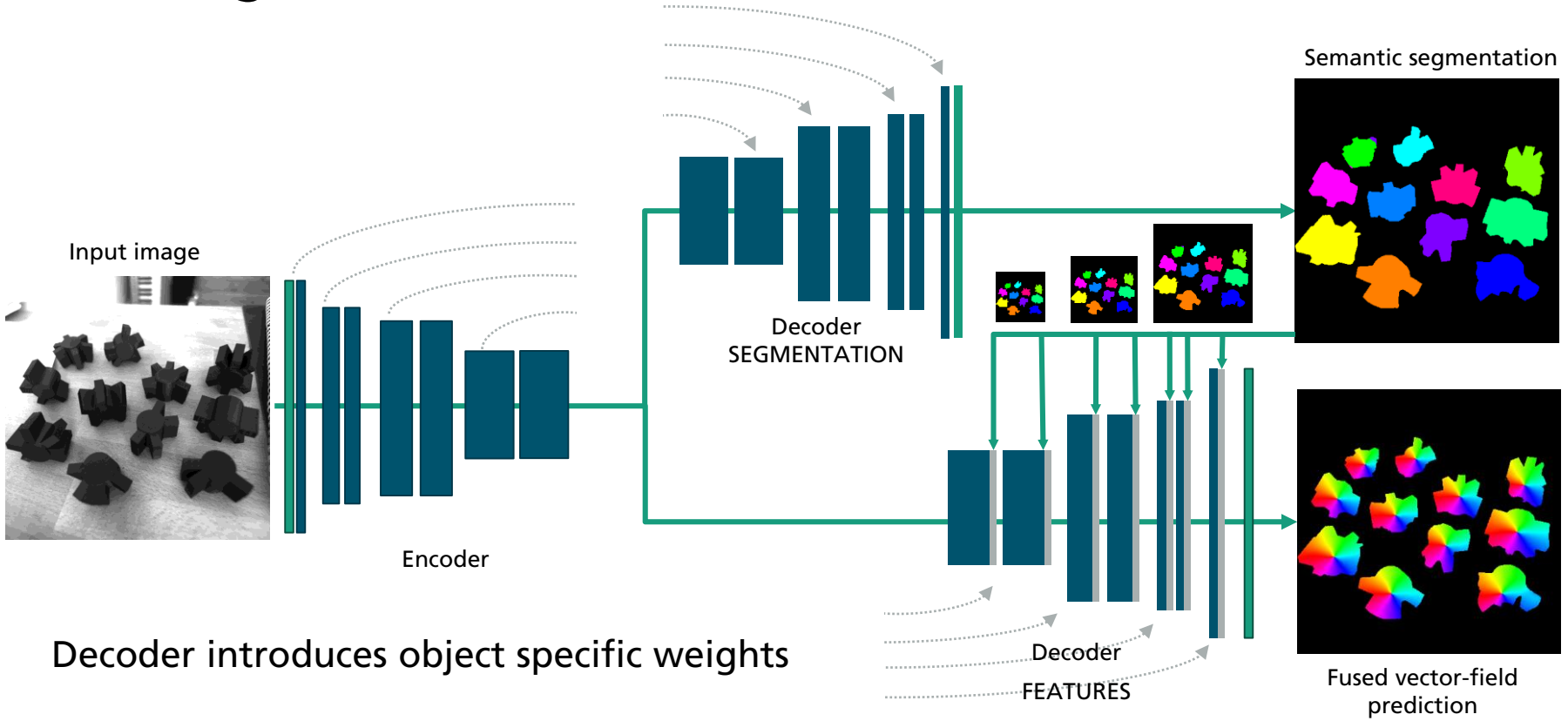
[1] Tan, Zhentao, et al. "Efficient Semantic Image Synthesis via Class-Adaptive Normalization." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

Fraunhofer HHI

Extending the network structure

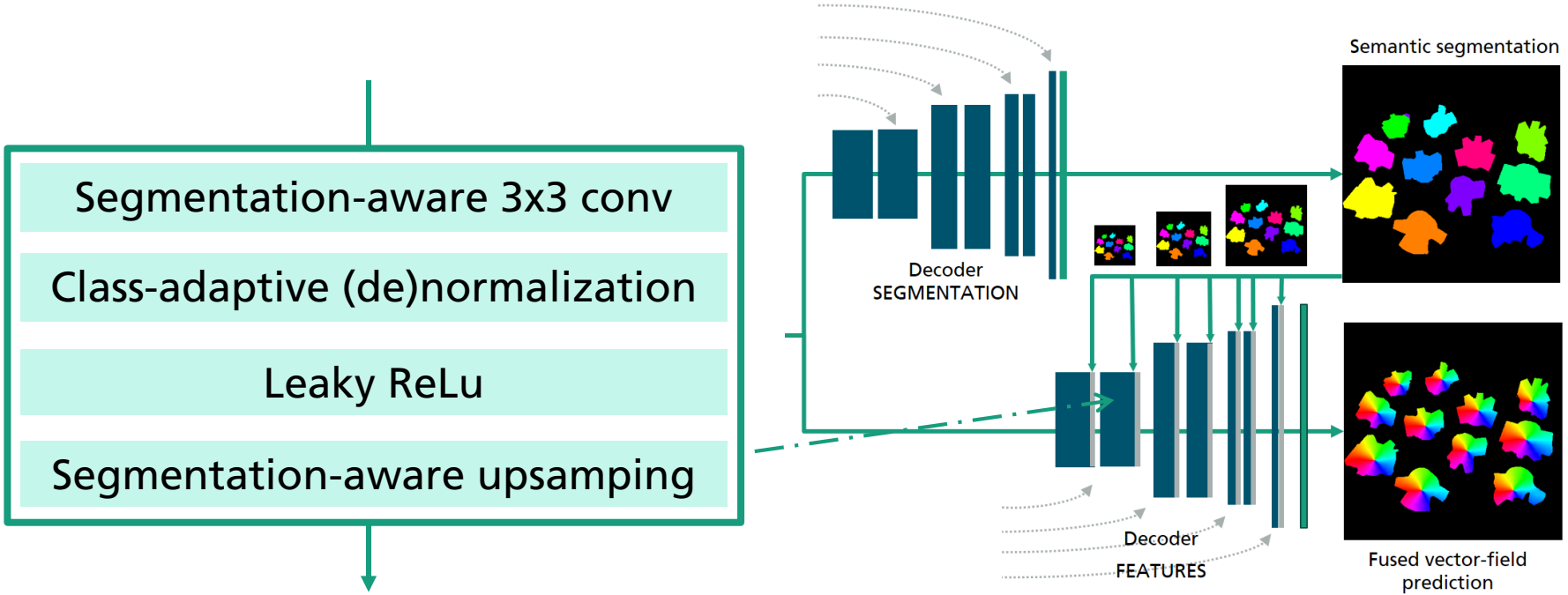


Extending the network structure



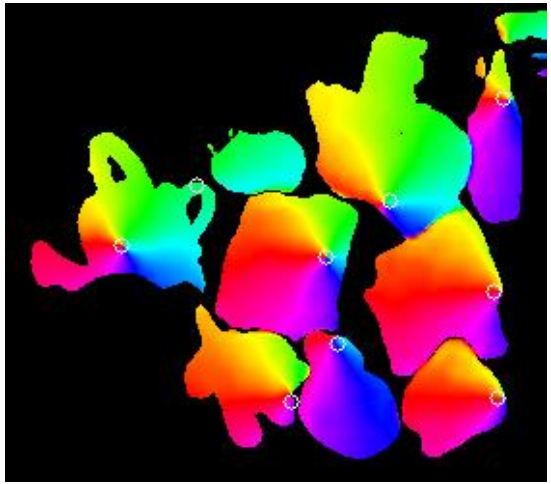
Decoder introduces object specific weights

Modified vector-field decoder

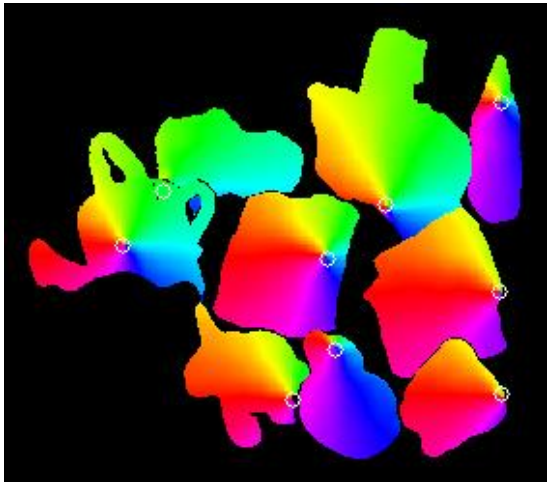


Segmentation-aware transformations

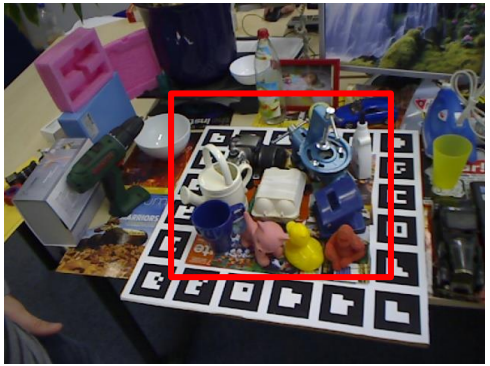
Improving accuracy at occluded areas



Vector-field without segmentation-awareness



Vector-field with segmentation-awareness

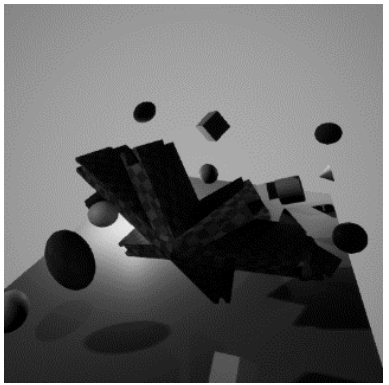


Input

- Features upsampling with respect to object boundaries
- Convolution respects object boundaries

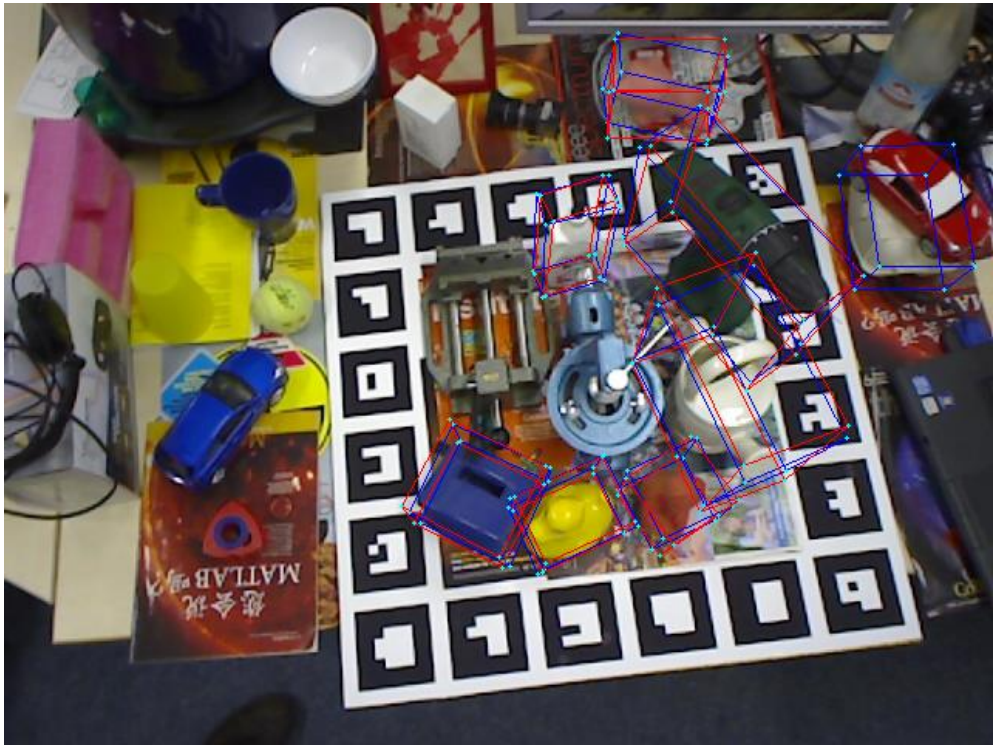
Training the network with synthethic data

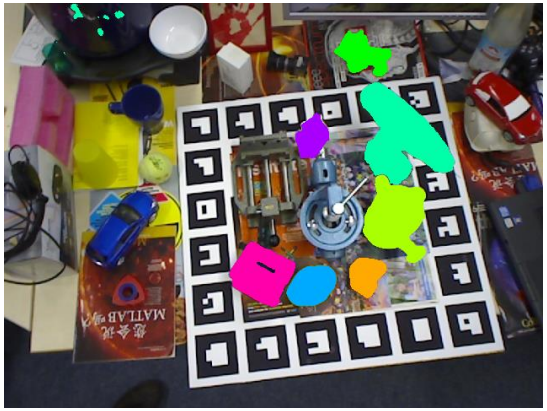
- Near-fotorealistic rendering
- Scene overview
- Rendered with blender



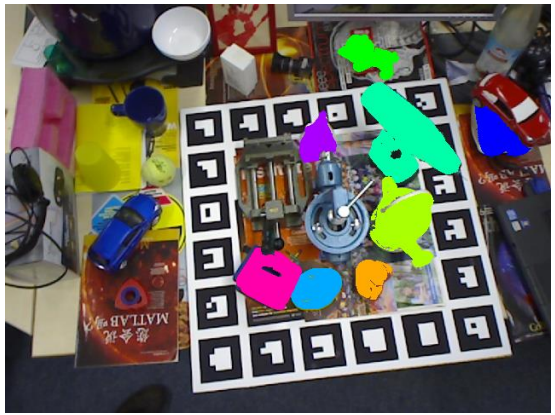
- Randomise scene parameters
- One object per image
- Rendered with Unreal Engine

Example result Linemod-Occlusion





Estimated segmentation

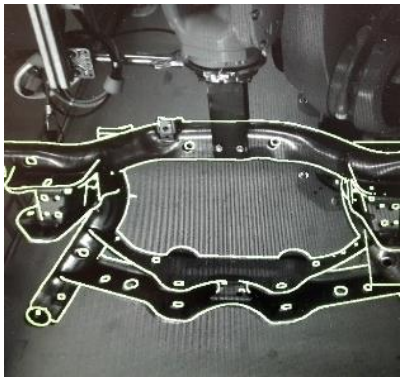


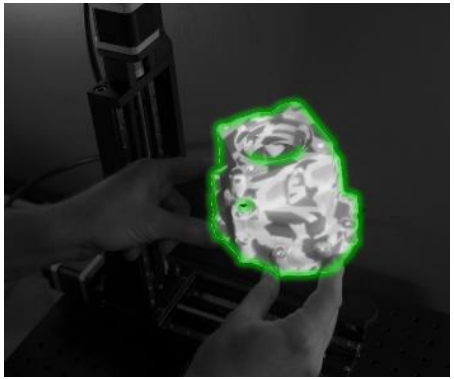
Groundtruth segmentation

Local Object Tracking

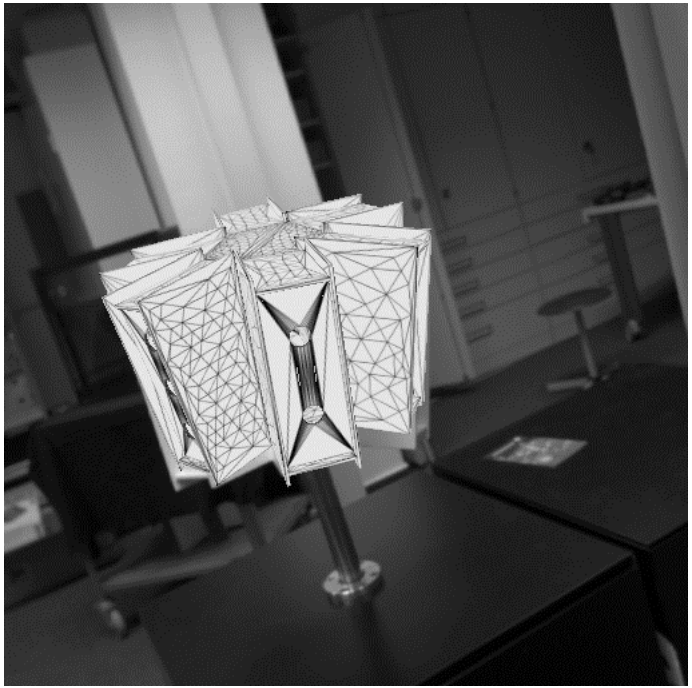
Local 6D tracking

Edge features for local pose estimation



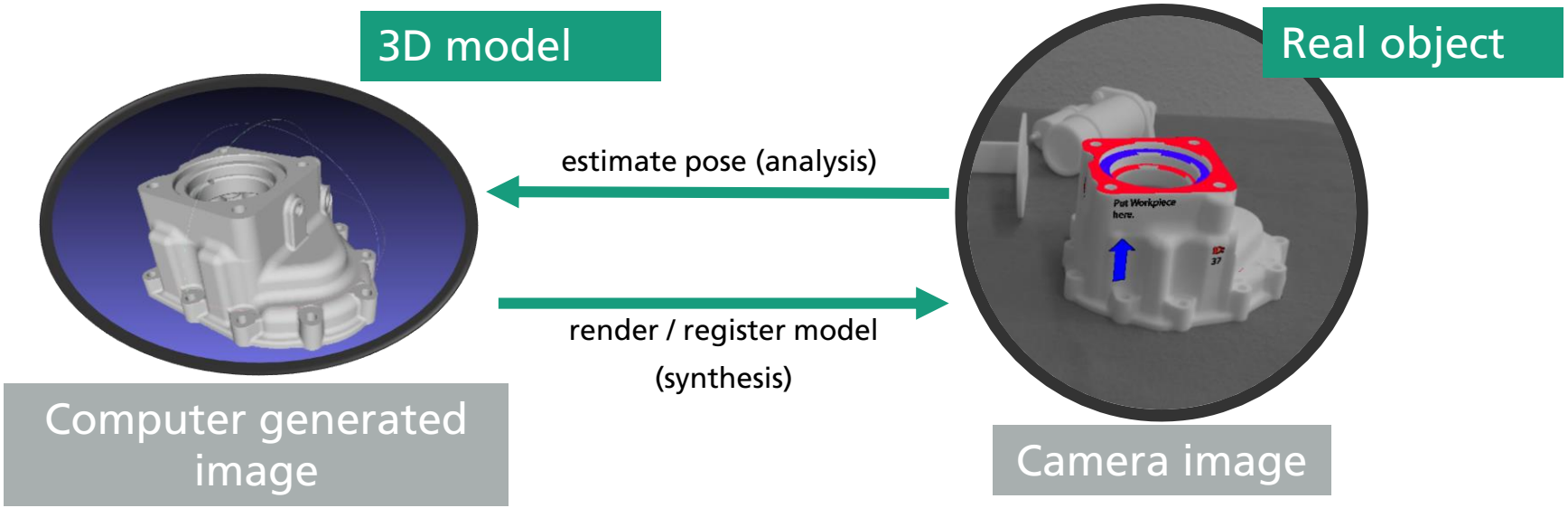


- **Edges** are dominant features of weakly textured objects
- **Optical-flow** between rendered simulation and camera image (using edge images)



Tracking from the ego-motion perspective

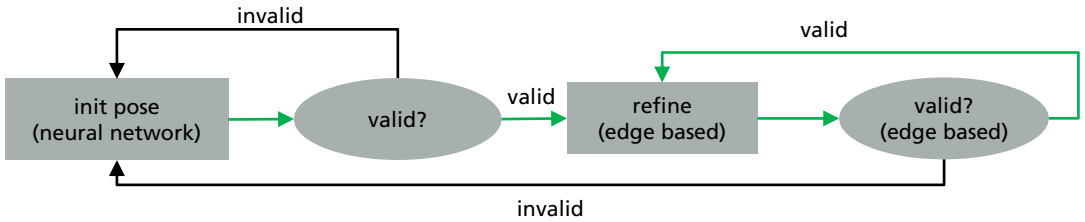
Analysis by synthesis

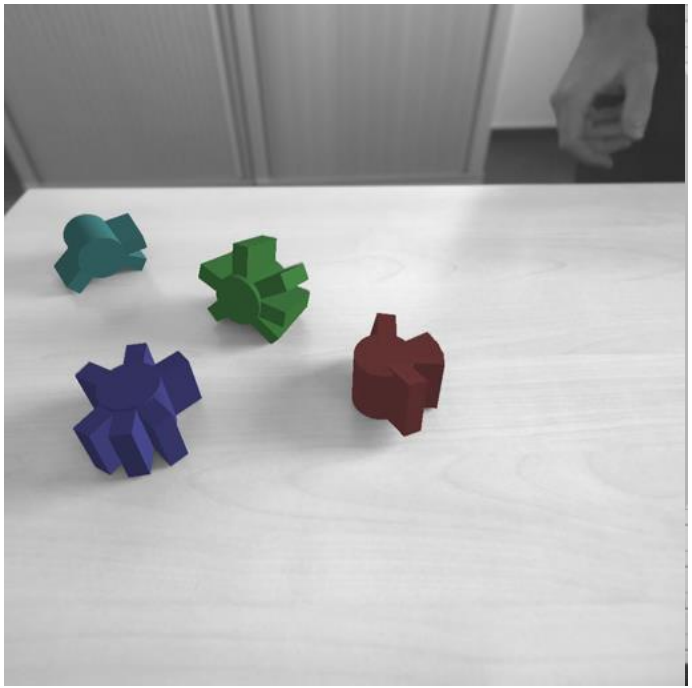


Fraunhofer HHI

Pose validation

- Use local tracking as much as possible
- Continuous validation of edge registration error to trigger reinitialization
- Avoid pose drift

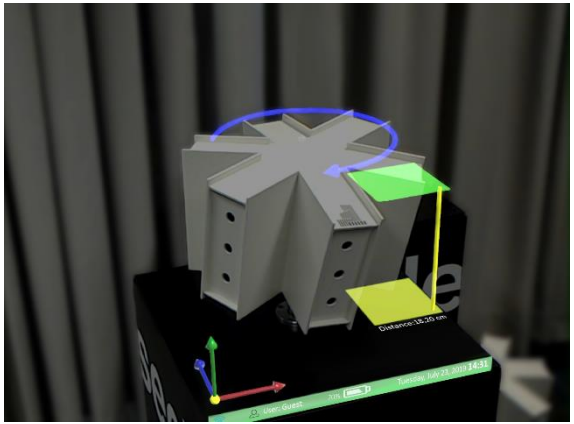




AR-Applications

Application in DTwin

AR assistance in facade construction



Warehouse scenario

Construction site

Niklas Gard

Fraunhofer HHI

Thank you for your attention!

Niklas Gard
niklas.gard@hhi.fraunhofer.de

Fraunhofer HHI, Computer Vision & Graphics

Einsteinufer 37
10587 Berlin

www.hhi.fraunhofer.de/vit/cvg

DIGITAL TWIN

Gefördert durch:
Bundesministerium für Wirtschaft und Energie
aufgrund eines Beschlusses des Deutschen Bundestages