NVGAZE: ANATOMY-AWARE AUGMENTATION FOR LOW-LATENCY, NEAR-EYE GAZE ESTIMATION

Michael Stengel, Alexander Majercik

AGENDA

Part I (Michael) 25 min

- Eye tracking for near-eye displays
- Synthetic dataset generation
- Network training and results

Part II (Alexander) 15 min

- Fast Network Inference using cuDNN
- Deep Learning Best Practice

NVGAZE TEAM

Michael Stengel
New Experiences Group

Joohwan Kim
New Experiences Group

Alexander Majercik
New Experiences Group

Shalini De Mello
Perception & Learning

Morgan McGuire
New Experiences Group

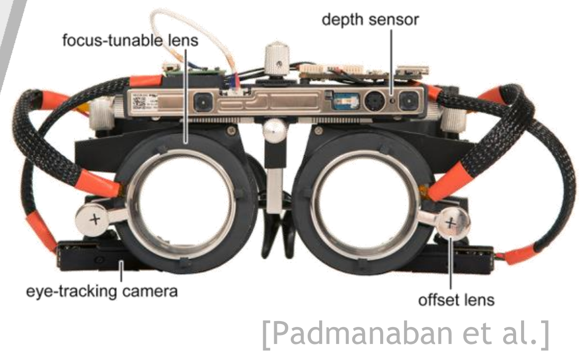
Samuli Laine
New Experiences Group

David Luebke
VP of Graphics Research

EYE TRACKING FOR NEAR-EYE DISPLAYS

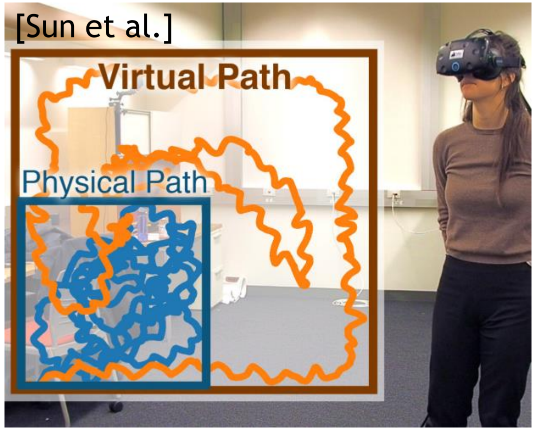
Michael Stengel

EYE TRACKING IN VR/AR

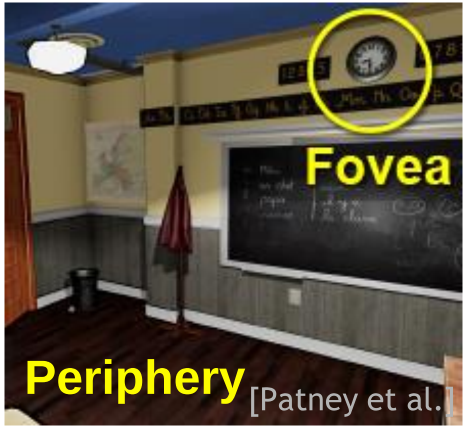


Computational Displays

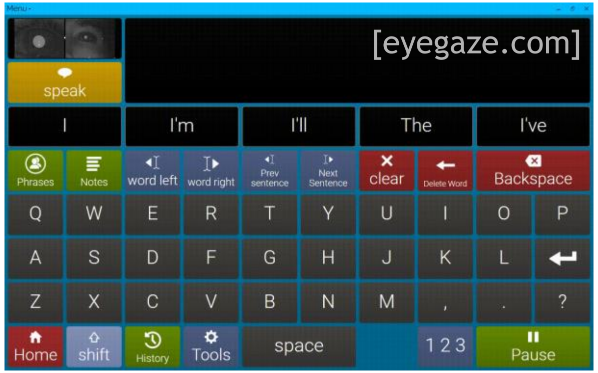
Avatars



Perception



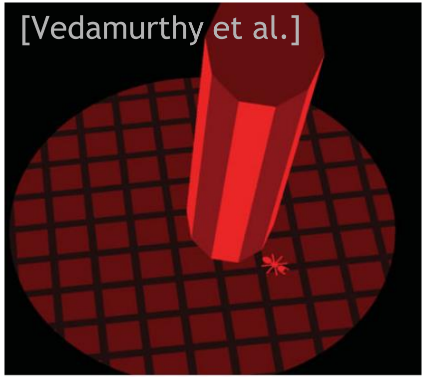
Foveated Rendering
Dynamic Streaming



Gaze Interaction

User State Evaluation

Attention Studies



Health Care

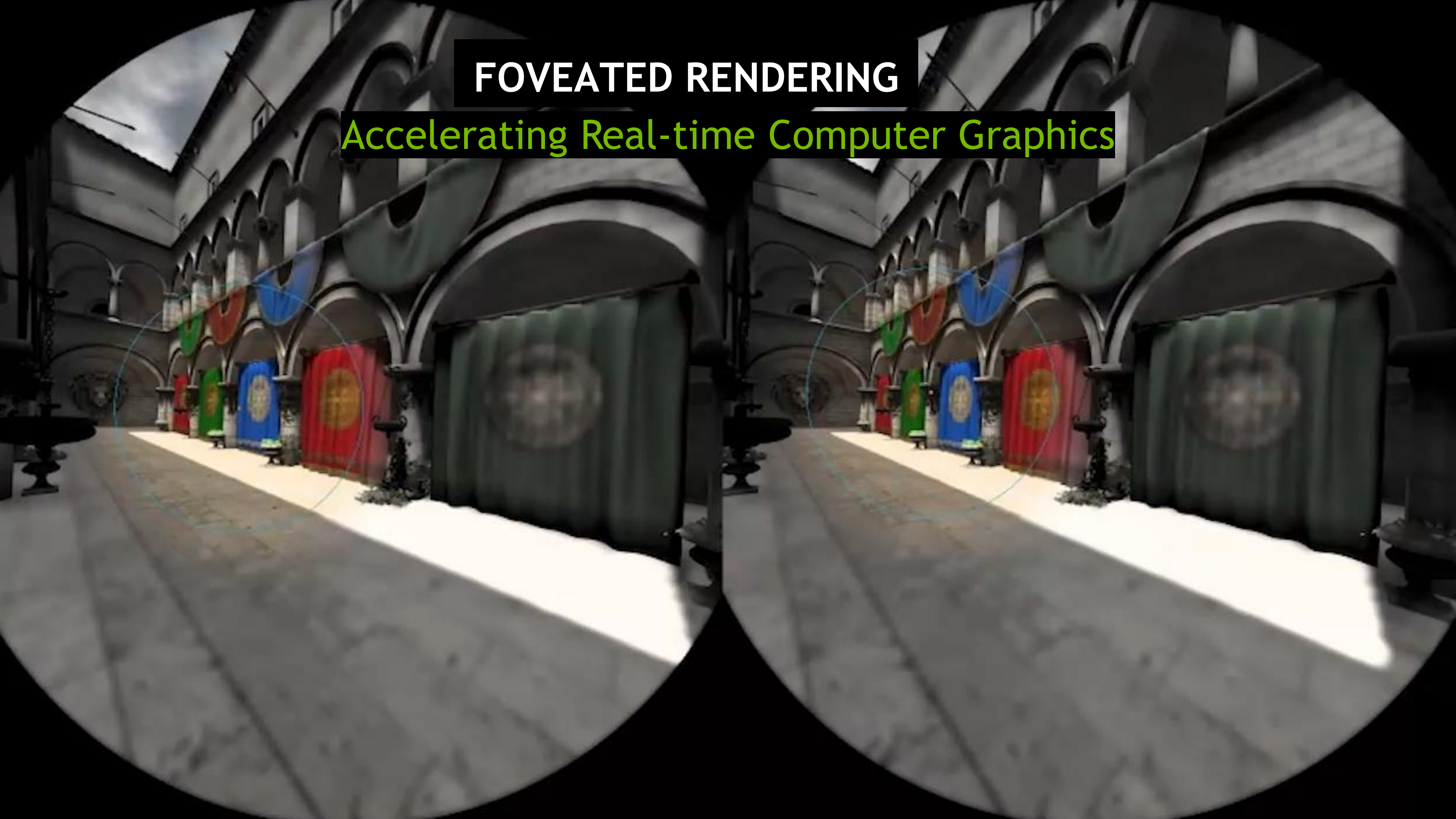
SUBTLE GAZE GUIDANCE

Enlarging virtual spaces through redirected walking

[Sun et al., Siggraph'18]

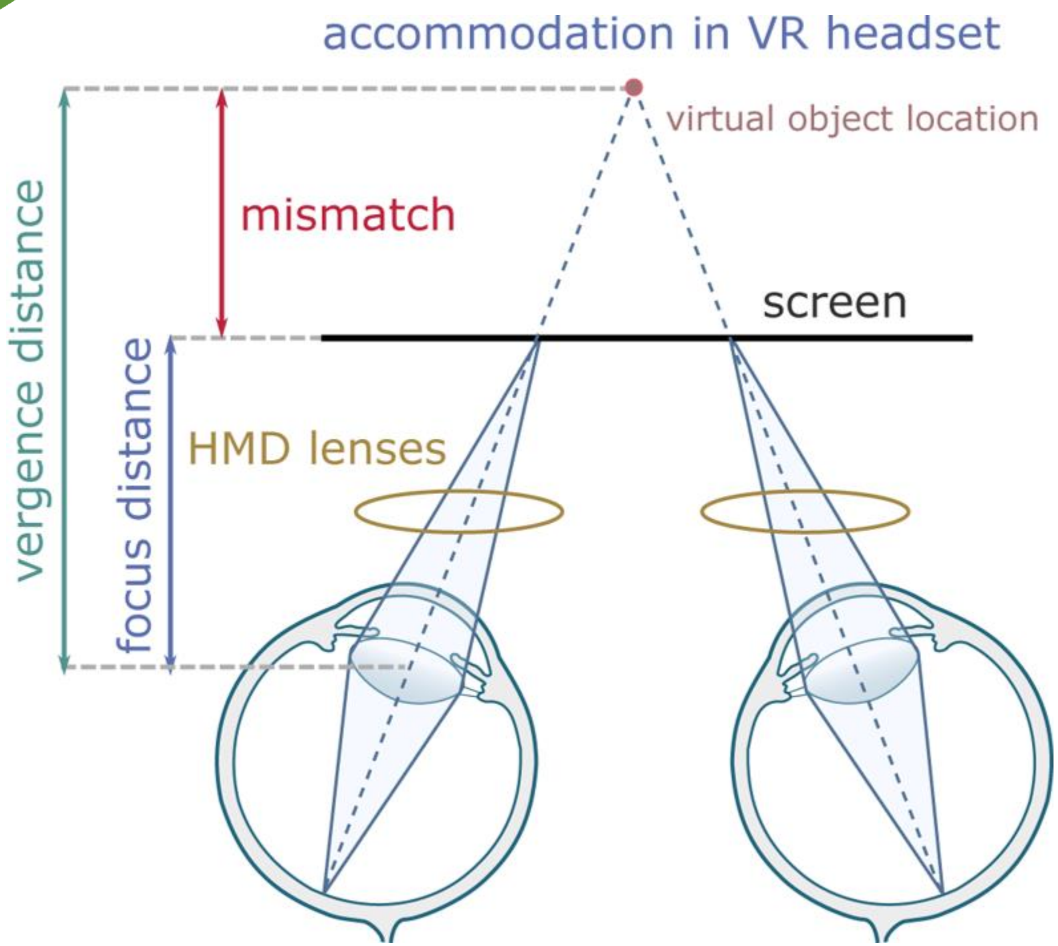
FOVEATED RENDERING

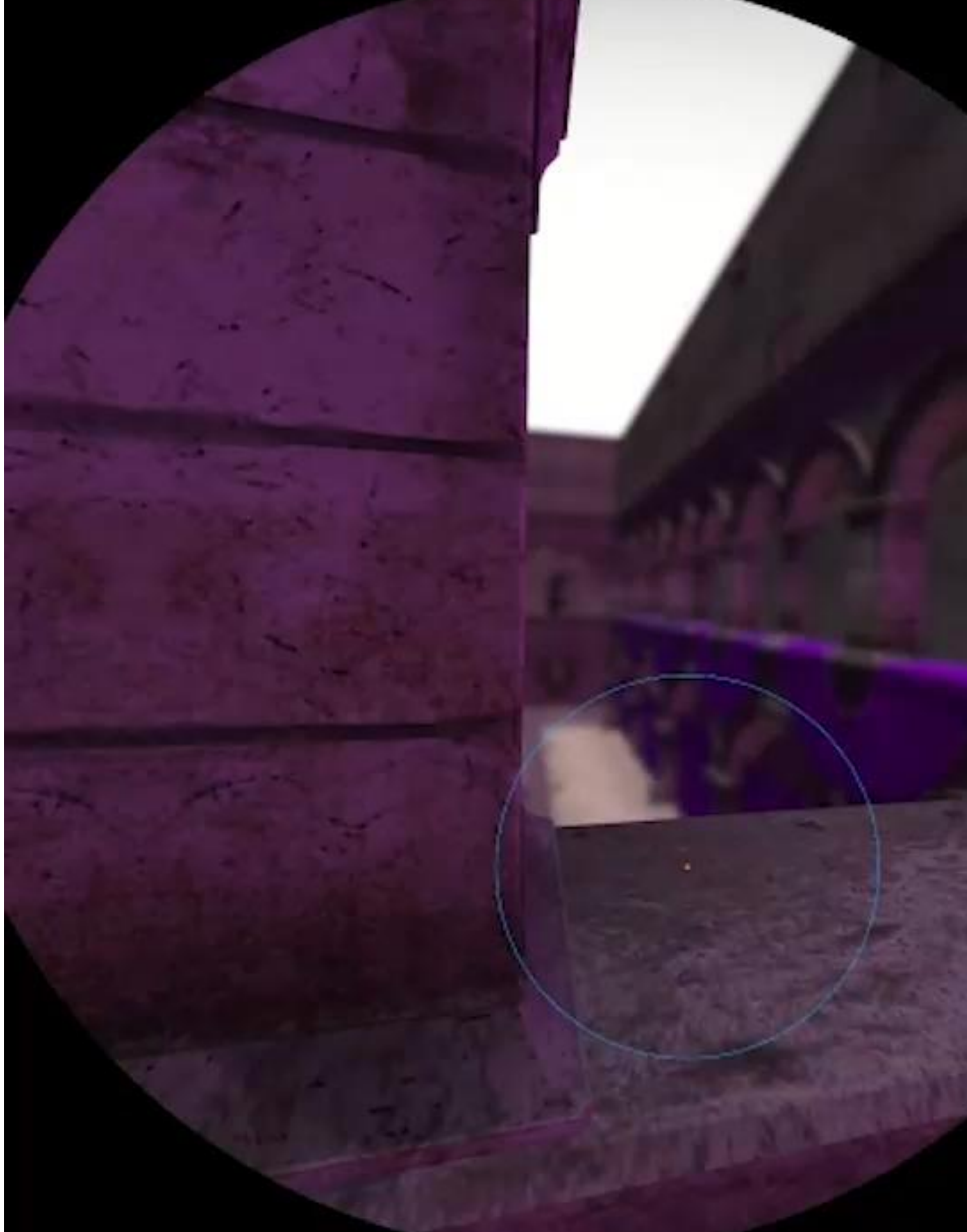
Accelerating Real-time Computer Graphics



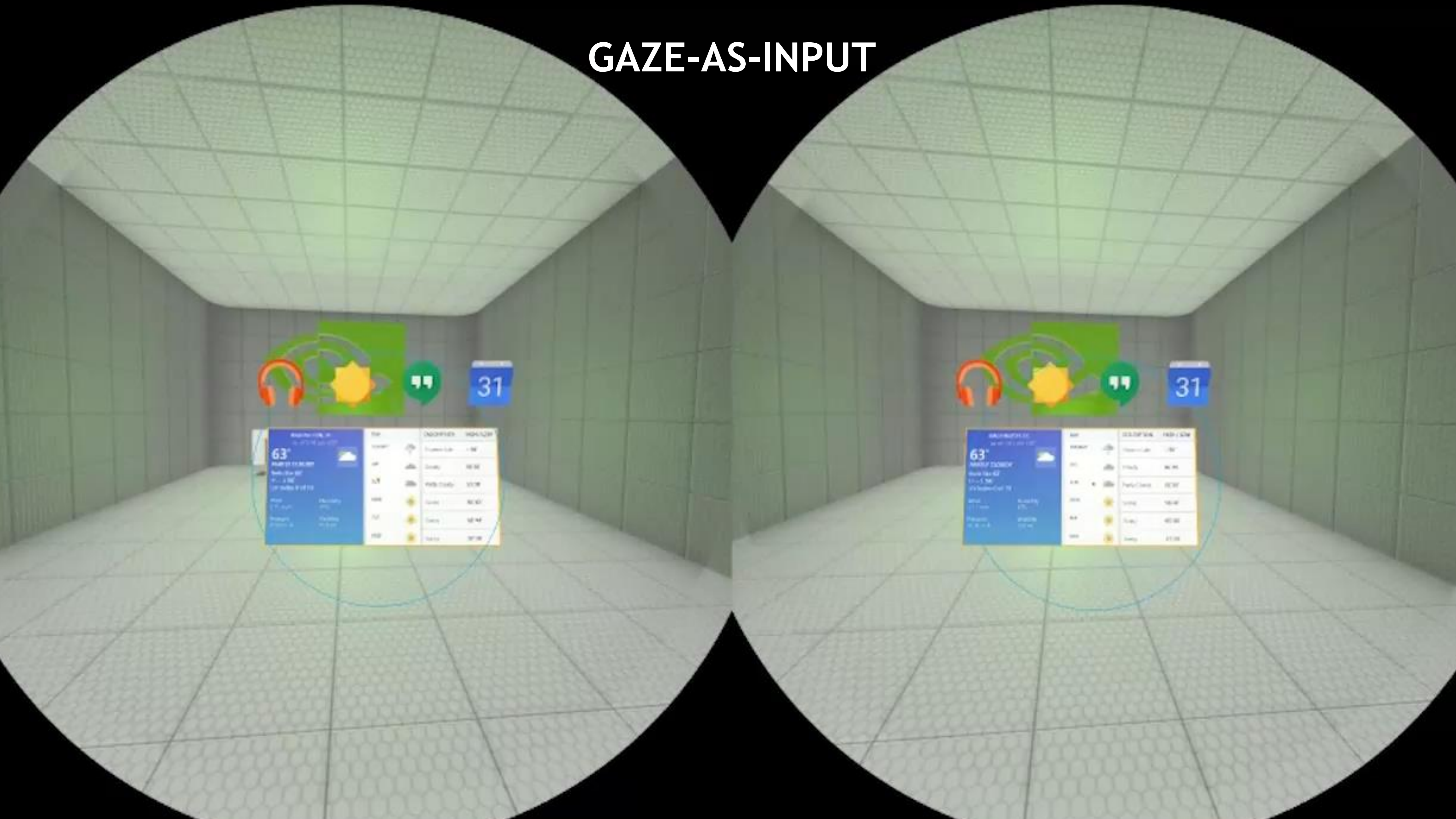
Enhancing Depth Perception

ACCOMMODATION SIMULATION





GAZE-AS-INPUT

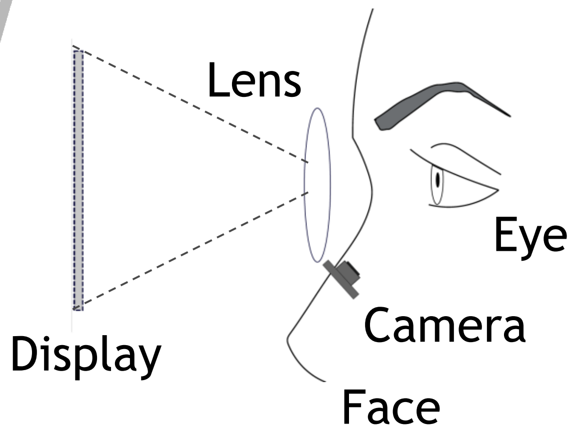


LABELED
REALITY

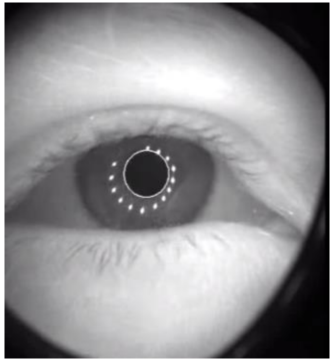
EYE TRACKING IN VR/AR

WORKING PRINCIPLE

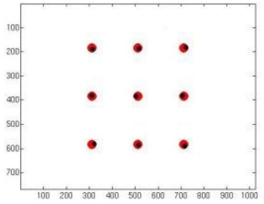
- How do video-based eye tracking systems work?



Eye capture

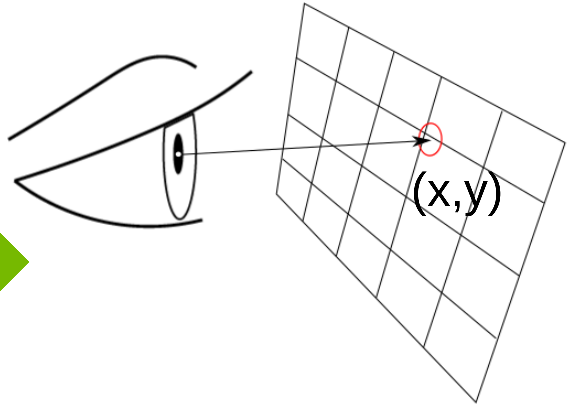


Pupil localization



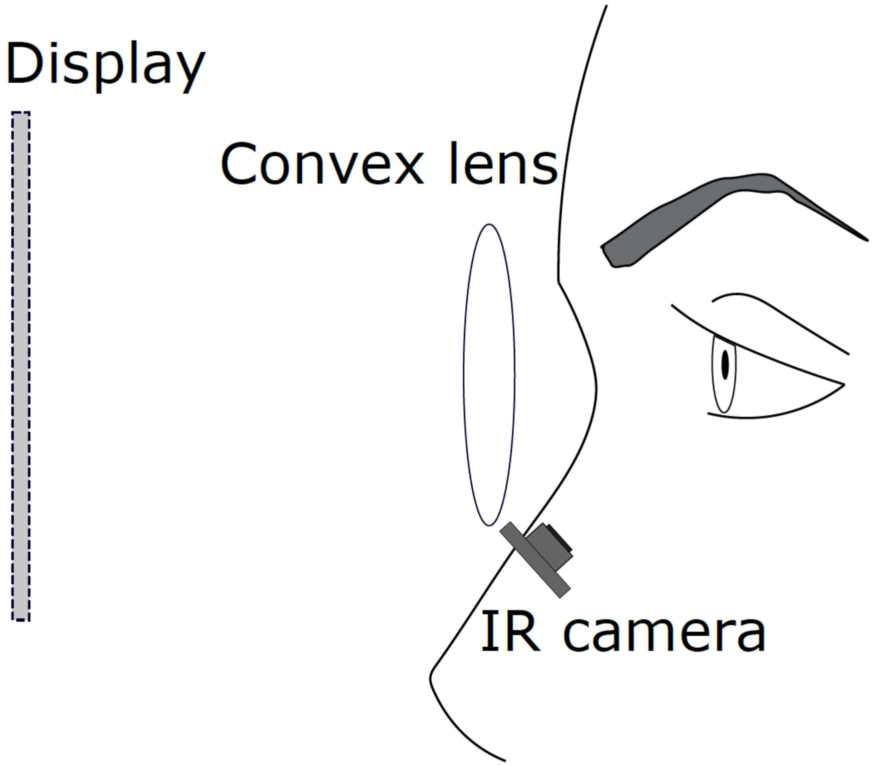
$$x = a_x \delta x + b_x \delta y + c_x$$
$$y = a_y \delta x + b_y \delta y + c_y$$

Domain mapping using calibration parameters

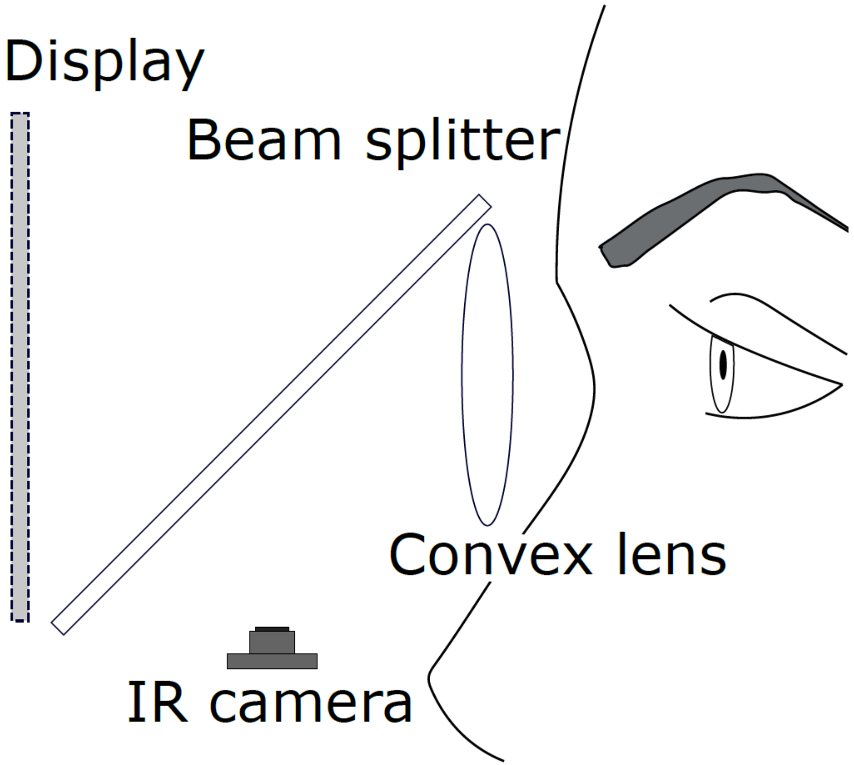


3d gaze vector or 2d point of regard

ON-AXIS VS OFF-AXIS GAZE TRACKING



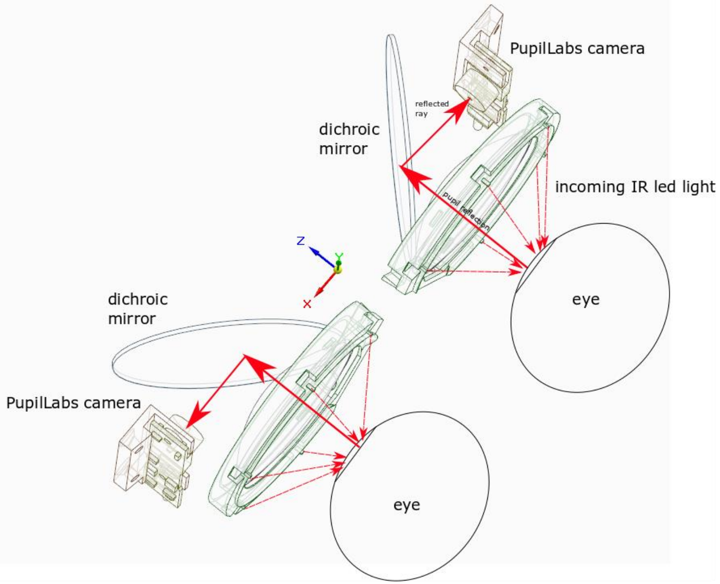
Camera view off-axis

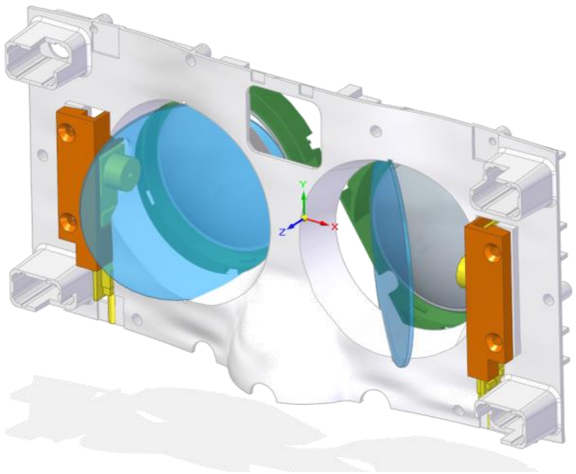


Camera view on-axis

ON-AXIS GAZE TRACKING

Eye tracking prototype for Virtual Reality headsets





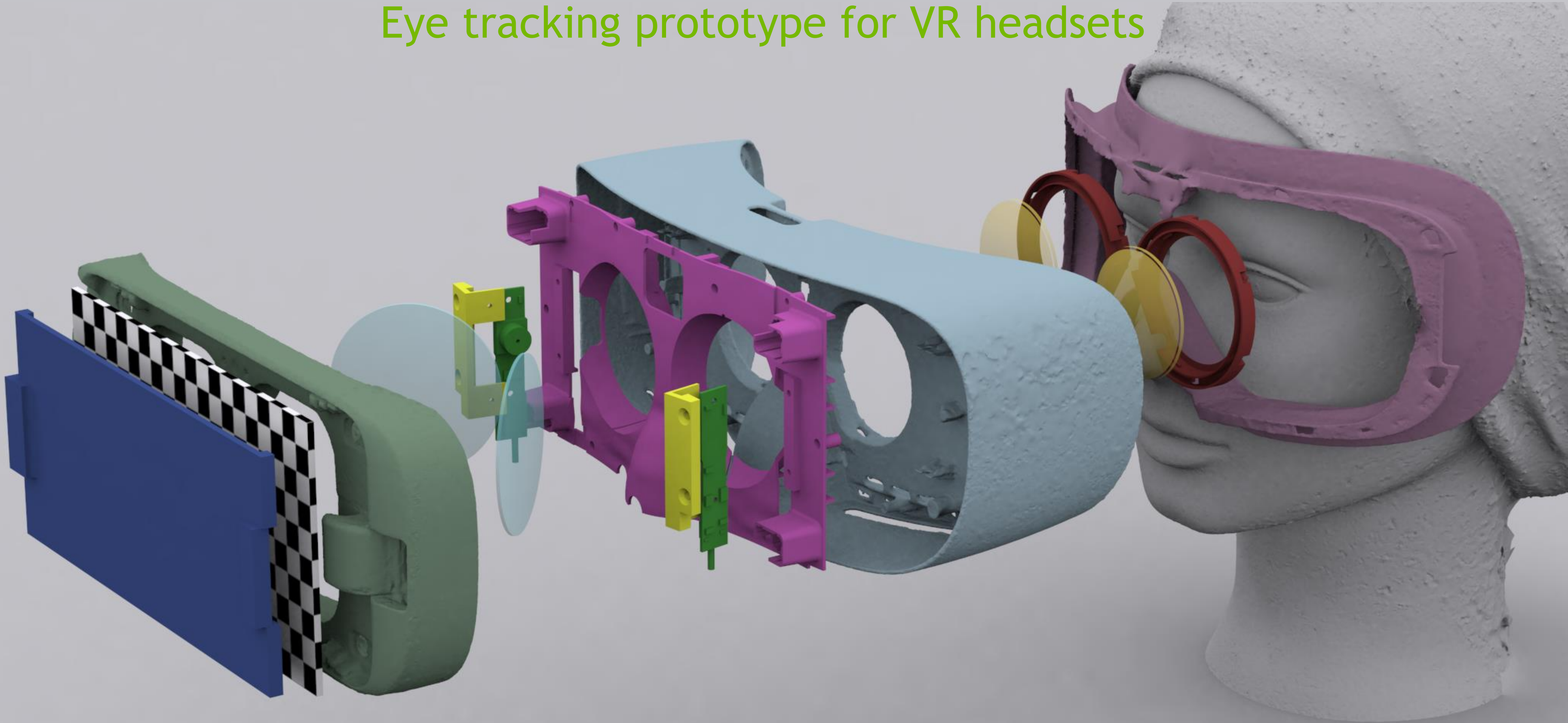
Components for **on-axis eye tracking integration**

Eye tracking cameras, dichroic mirrors, infrared illumination, VR glasses frame

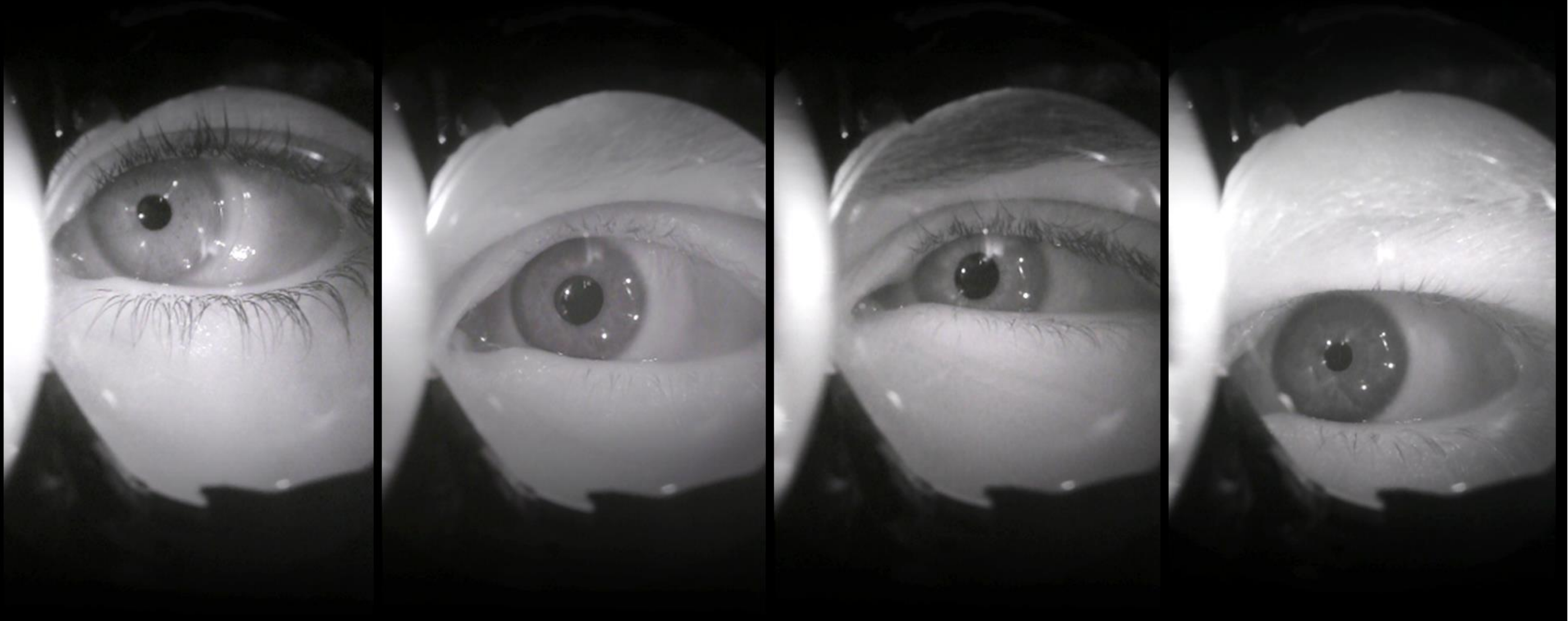
Modded GearVR with integrated gaze tracking

ON-AXIS GAZE TRACKING

Eye tracking prototype for VR headsets

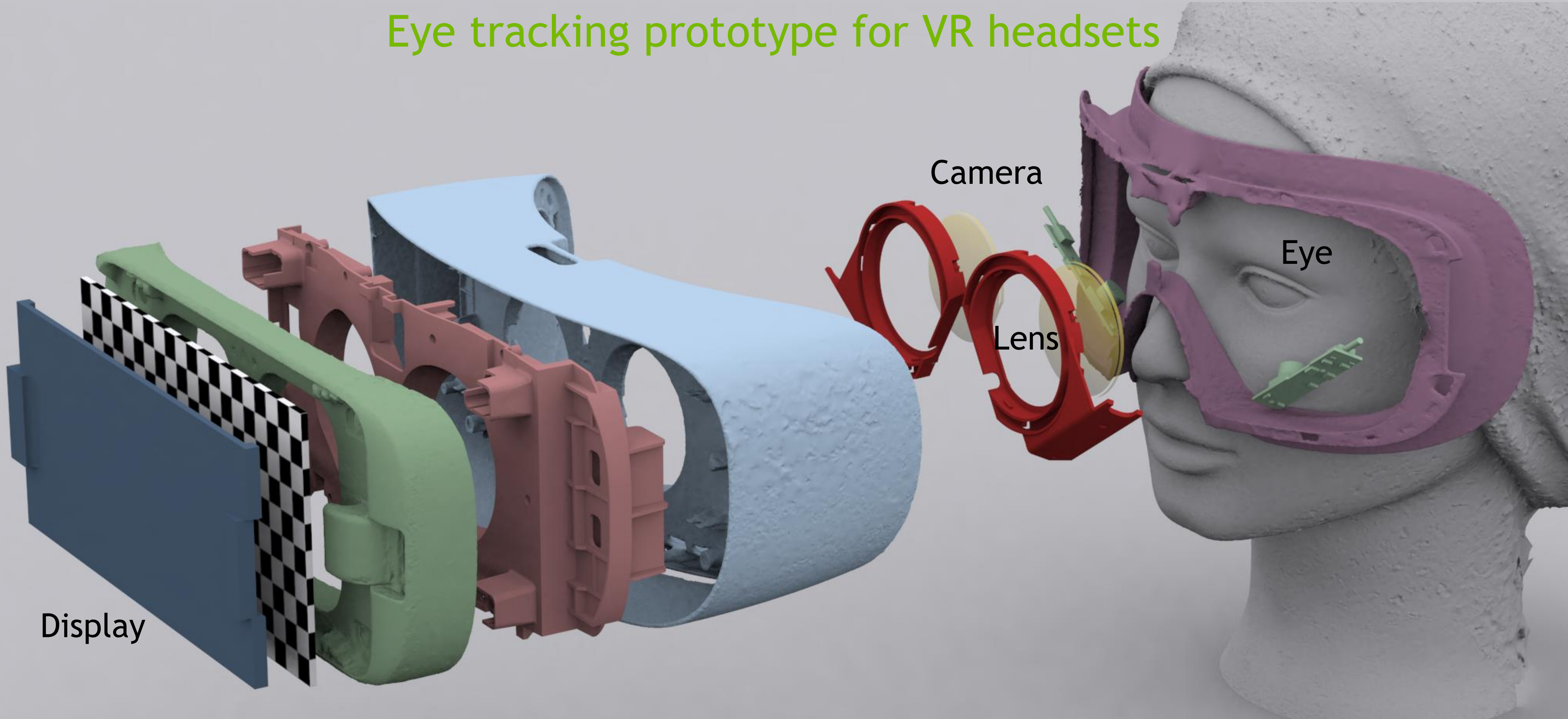


ON-AXIS EYE TRACKING CAMERA VIEW



OFF-AXIS GAZE TRACKING

Eye tracking prototype for VR headsets



OFF-AXIS GAZE TRACKING

Eye tracking prototype for VR headsets

EYE TRACKING IN VR/AR

CHALLENGES FOR MOBILE VIDEO-BASED EYE TRACKERS

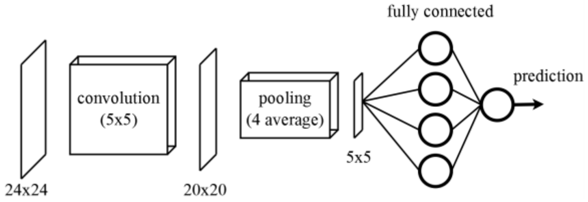
- Changing illumination conditions (over-exposure and hard shadows)
- Occlusions from eyes lashes, skin, blink, glasses frame
- Varying eye appearance : flesh, mascara and other make-up
- Reflections
- Camera view and noise (blur, defocus, motion)
- drifting calibration (single-camera case) due to HMD or glasses motion
- End-to-end latency → Reaching low latency AND high robustness is hard !
- Capturing training data is expensive

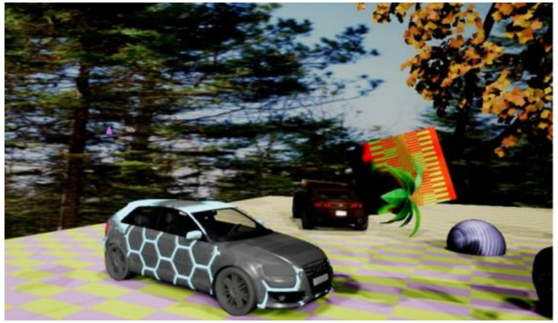
PROJECT GOALS

- Deep learning based gaze estimation
- Higher robustness than previous methods
- Target accuracy is < 2 degrees of angular error (over full field of view!)
- Fast inference ranging in a few milliseconds even on mobile GPU
- Compatibility to any captured input (on-axis, off-axis, near-eye, remote, etc., dark pupil tracking only, glint-free tracking)
- Explore usage of *synthetic* data
- Can we learn increase calibration robustness ?

RELATED RESEARCH

- PupilNet [Fuhl et al., 2017]
  - 2-pass CNN-based method running in 8 ms (CPU) performing pupil localization task
  - 1st pass on low res image (96x72 pixels)
  - 2nd pass on full-res image (VGA resolution)
  - trained on 135k manually labeled real images
  - Higher robustness than previous ‘hand-crafted’ pupil detectors
- Domain Randomization [Tremblay et al., Nvidia, 2018]
  - Image and label generator for automotive setting
  - Randomized objects force network to learn essential structure of cars independent of view and lighting condition



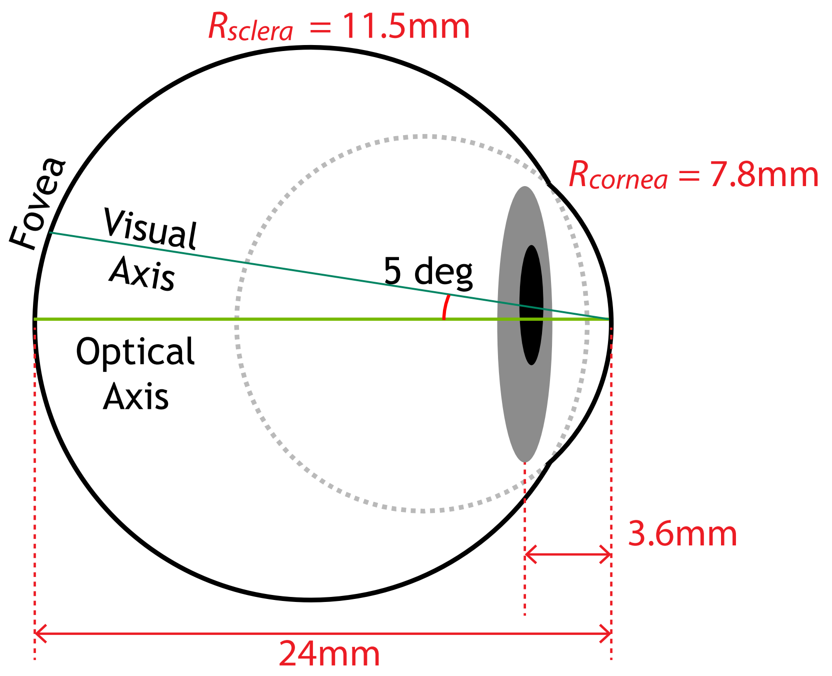


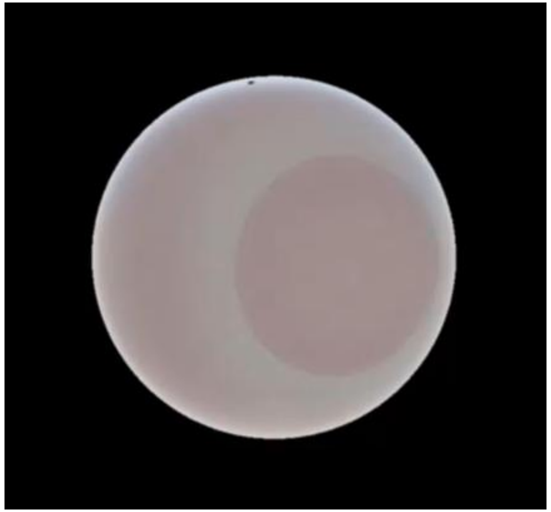
NVGAZE
SYNTHETIC EYES DATASET

GENERATING TRAINING DATA

1: Eye Model

We adopted the eye model from Wood et al. 2015 * and modified it to more accurately represent human eyes.

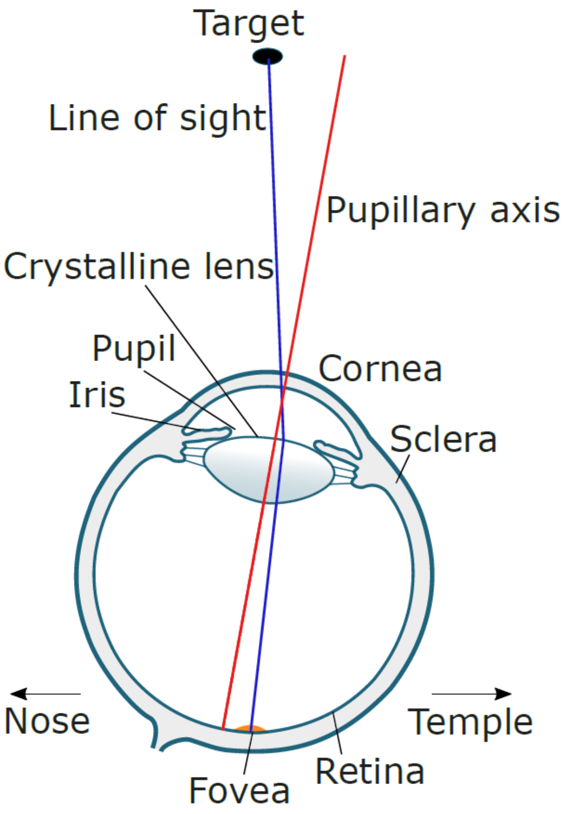


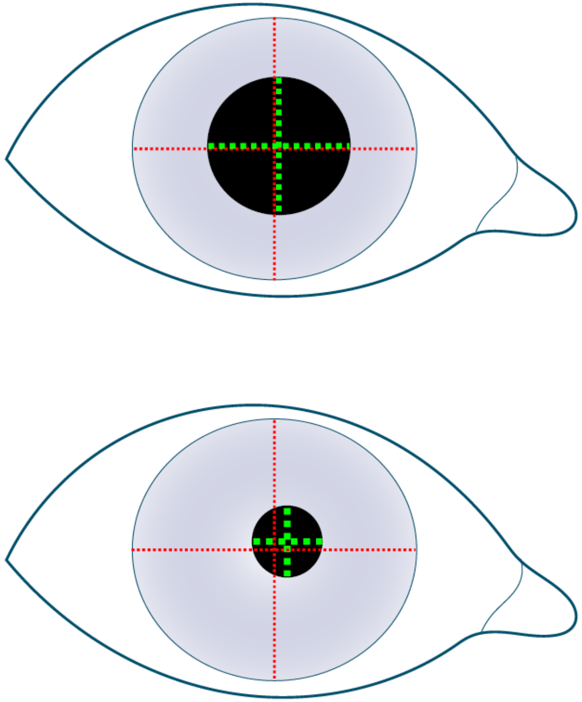


* Wood, E., Baltrušaitis, T., Zhang, X., Sugano, Y., Robinson, P., & Bulling, A. “Rendering of eyes for eye-shape registration and gaze estimation”, *ICCV* 2015.

GENERATING TRAINING DATA

2: Pupil Center Shift





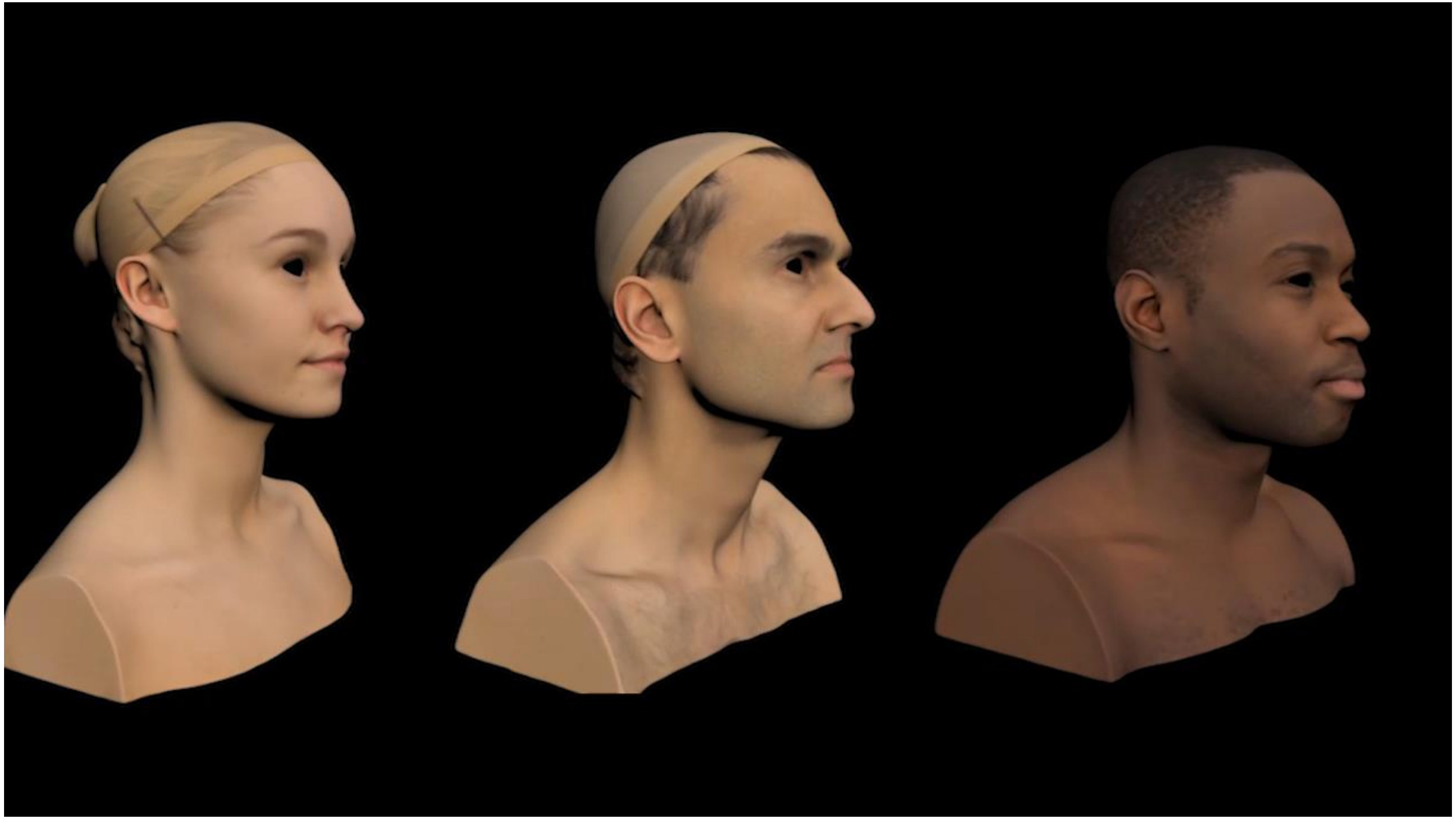
Pupil center is off from iris center, and it moves as pupil changes in size.

Average displacements:
8mm pupil: 0.1 mm nasal and 0.07 mm up
6mm pupil: 0.15 mm nasal and 0.08 mm up
4mm pupil: 0.2 mm nasal and 0.09 mm up

This is known to cause gaze tracking error of up to 5 deg in pupil-glint tracking methods.

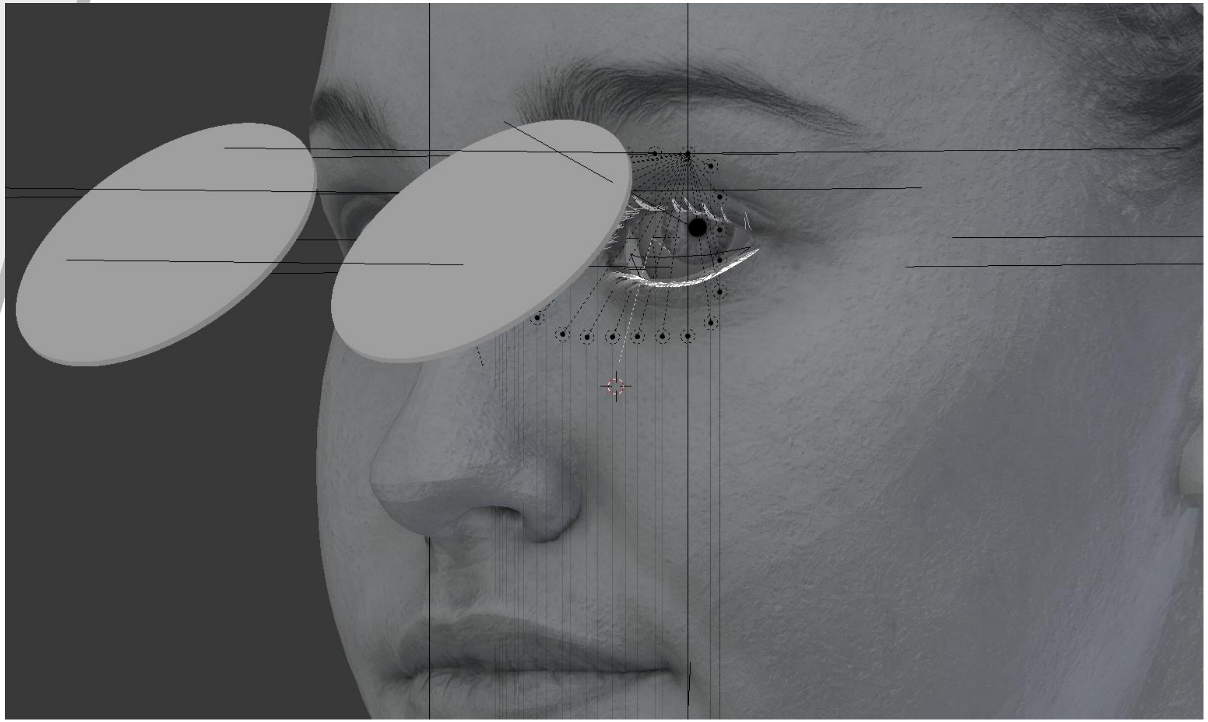
GENERATING TRAINING DATA

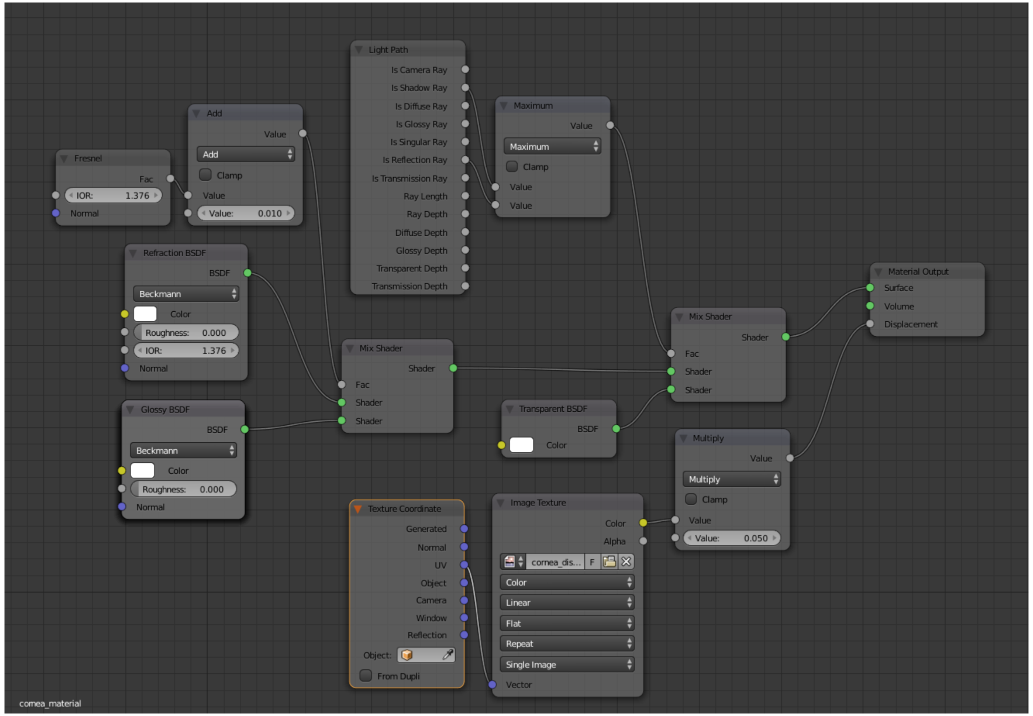
2: Scanned faces



GENERATING TRAINING DATA

2: Combining Eye and Head Models

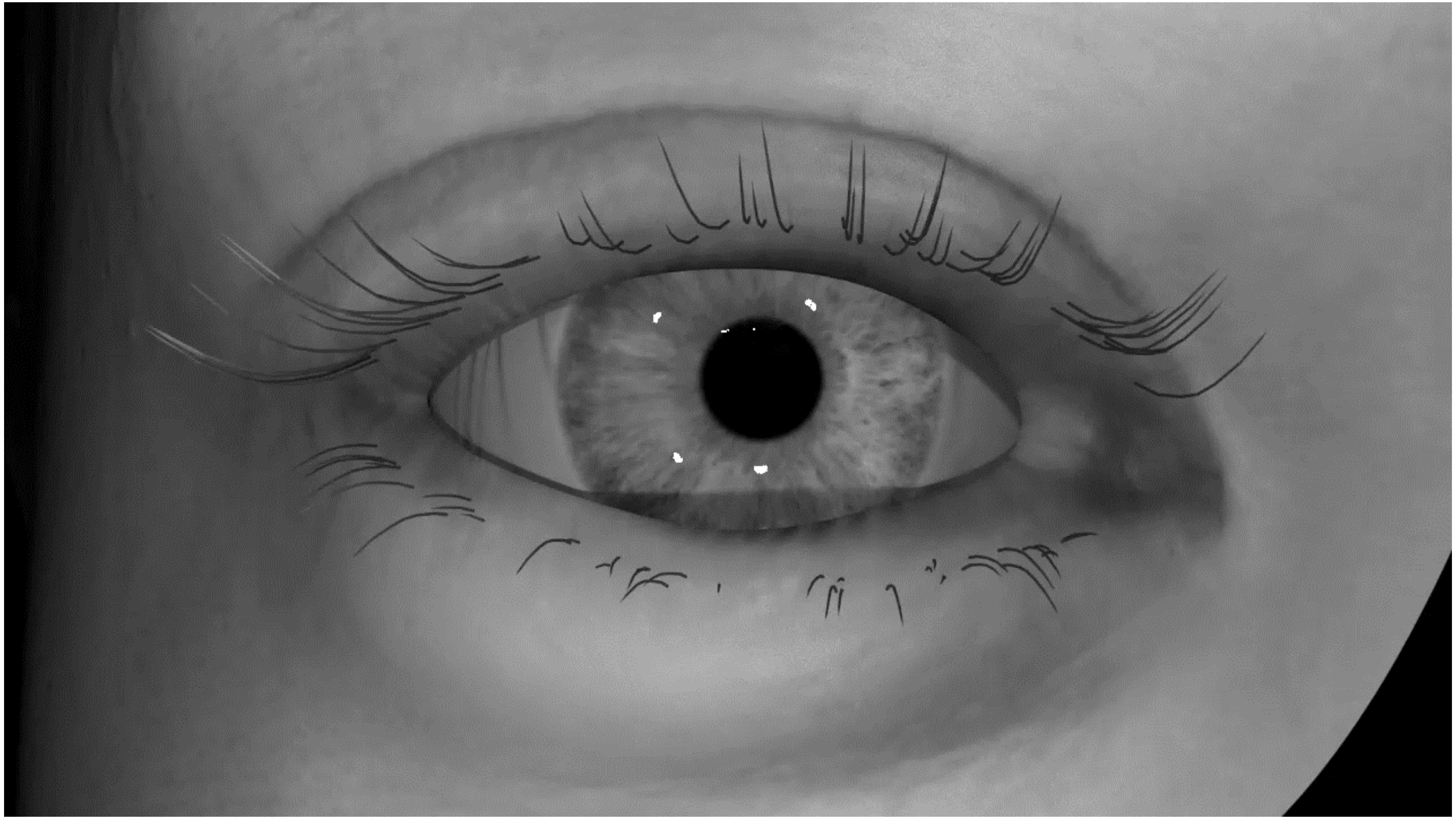




- **10 scanned faces** with photorealistic eye, adopted the eye model from Wood et al. 2015
- **physical** material properties for cornea, sclera and skin under **infrared lighting** conditions

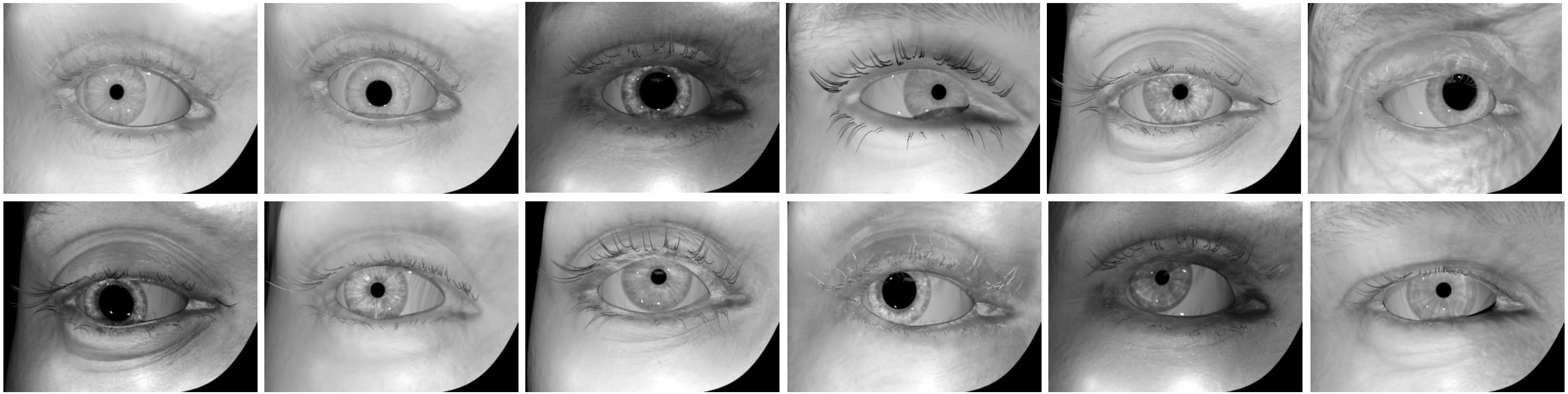
GENERATING TRAINING DATA

2: Synthetic Model



GENERATING TRAINING DATA

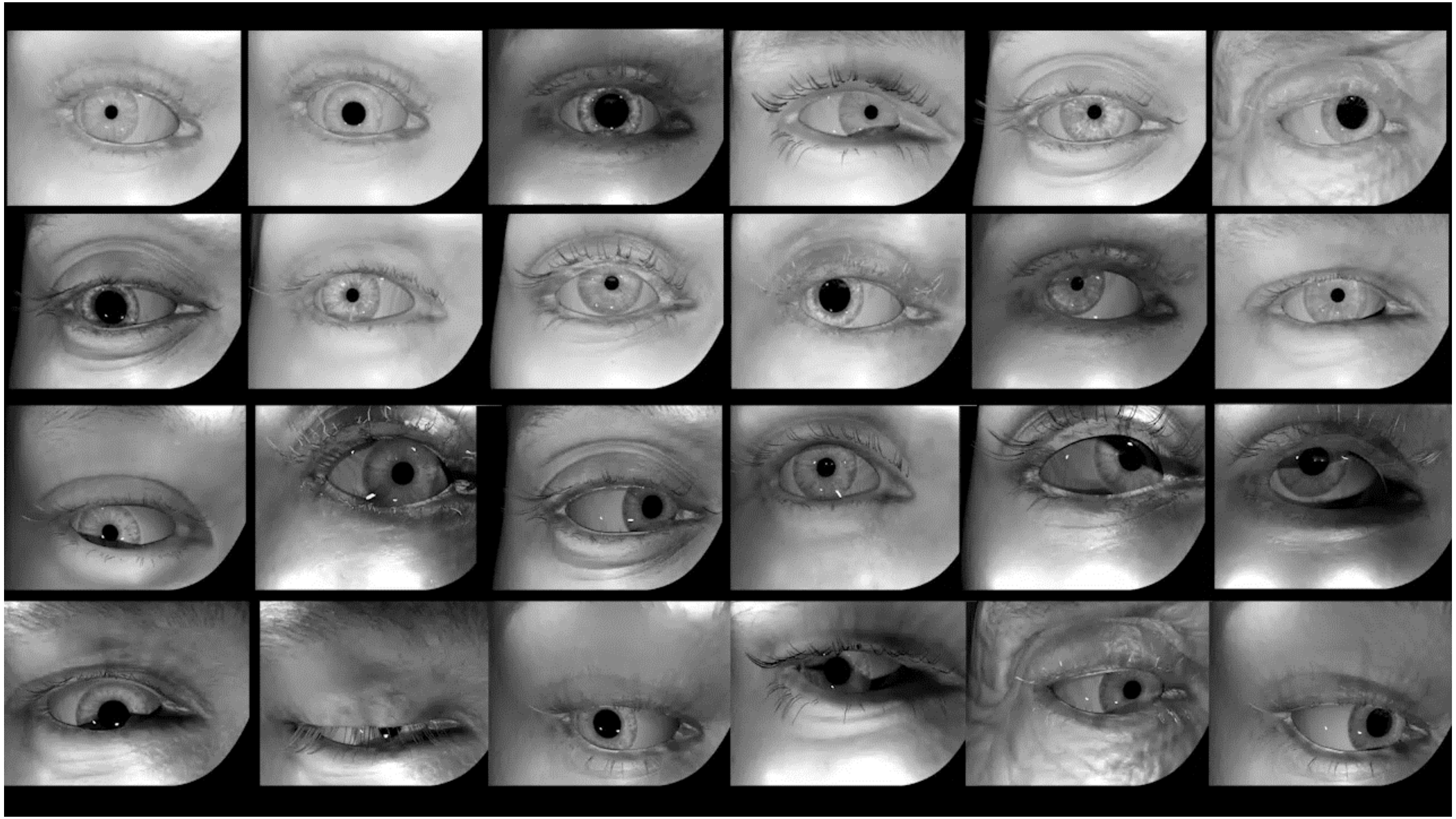
3: Dataset



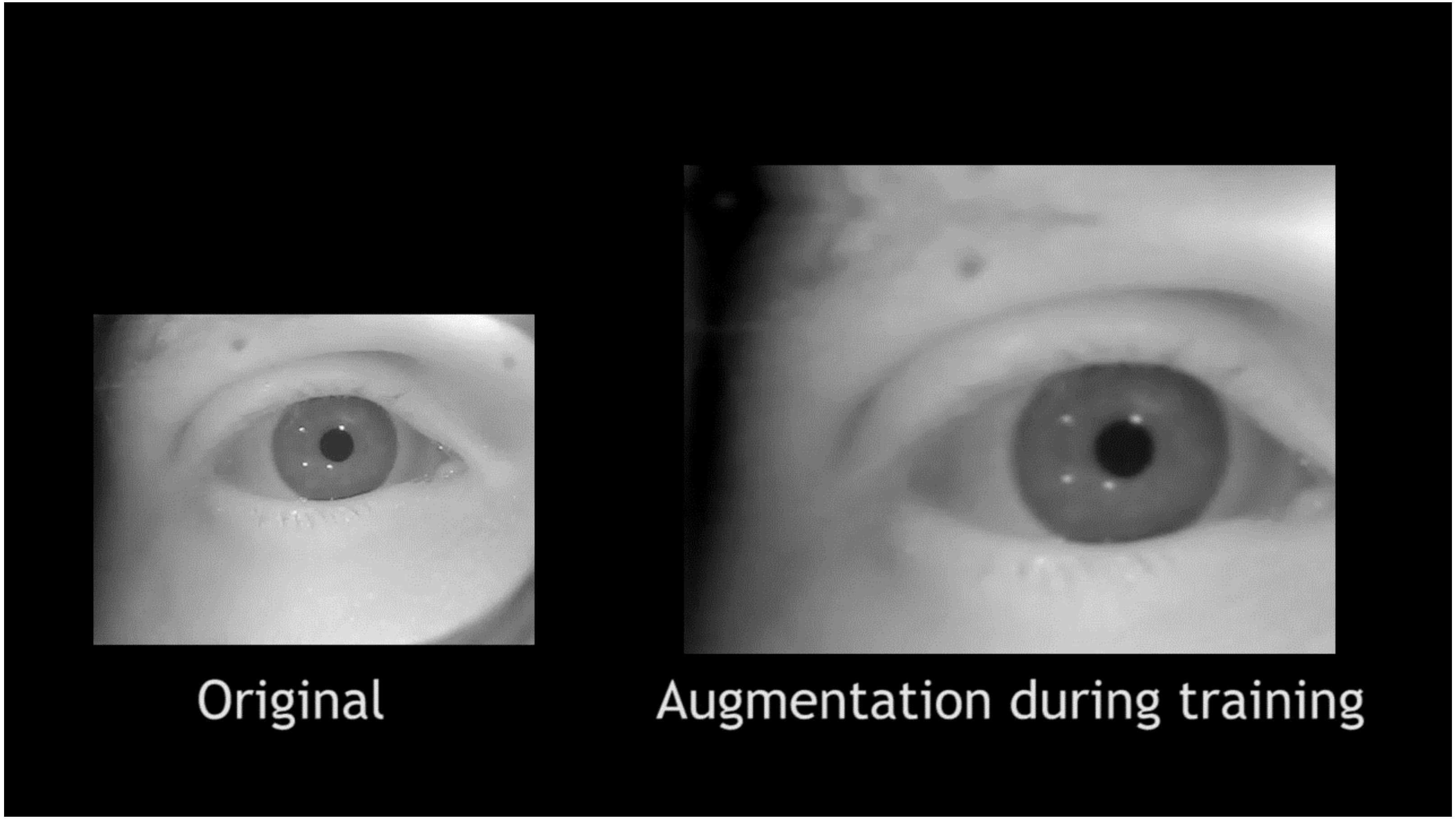
- **4M Synthetic HD eye images** for animated eye (400K images per subject) are generated using Blender on Multi-GPU cluster.
- Render engine used is **Cycles** as physically accurate path tracer.

GENERATING TRAINING DATA

3: Dataset

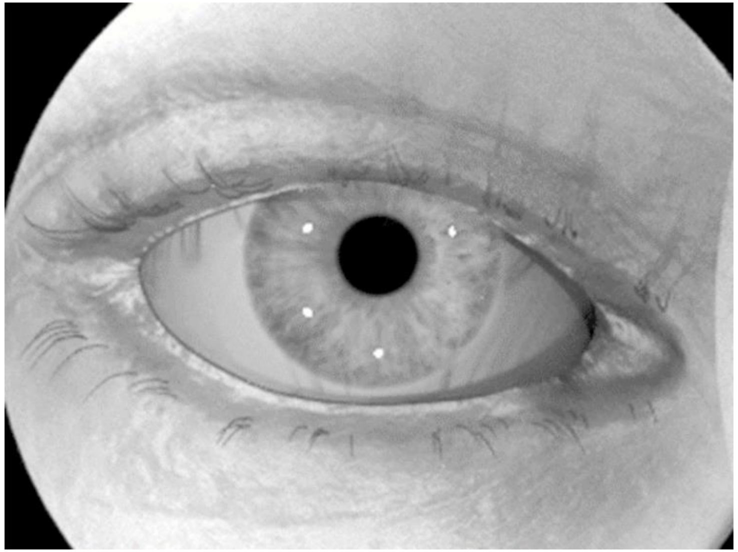


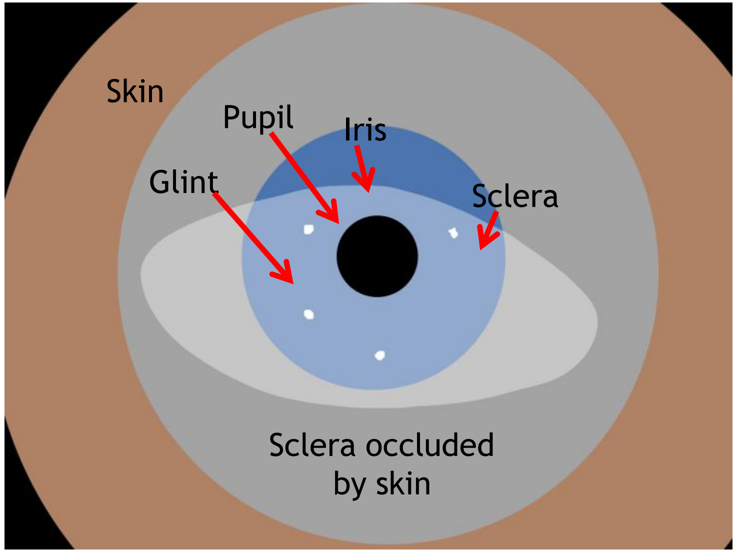
ANATOMY-AWARE AUGMENTATION



GENERATING TRAINING DATA

4: Region Labels

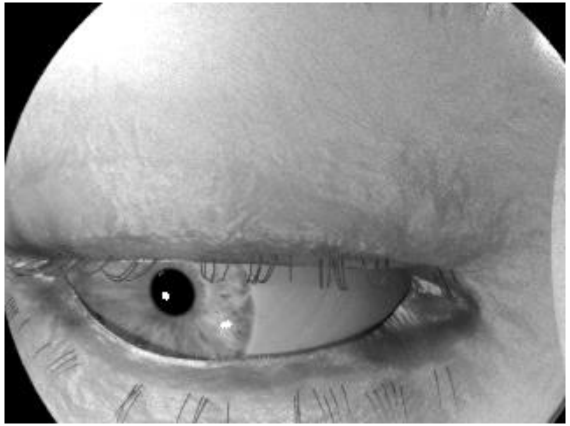




- **Region maps** are generated out of images with self-illuminating material.
- Refractive effect of air-cornea layer is accounted for.
- **Synthetic ground truth** is available even if regions are occluded by skin (during blink).

ANATOMY-AWARE AUGMENTATION

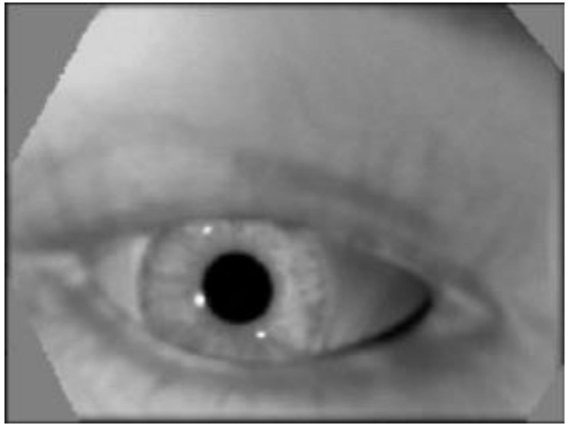
Original Synthetic Image



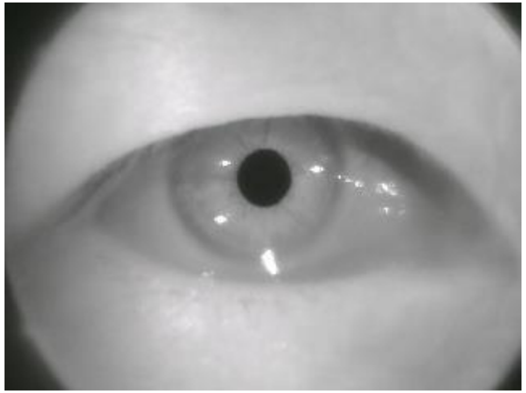
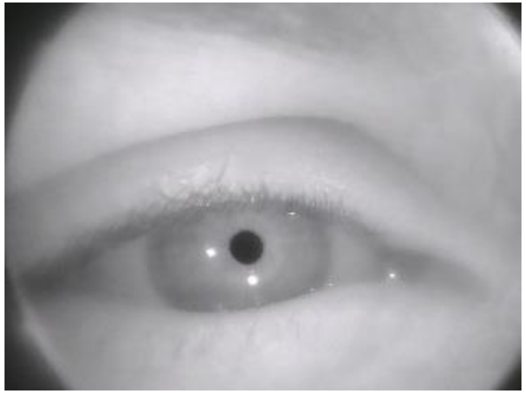
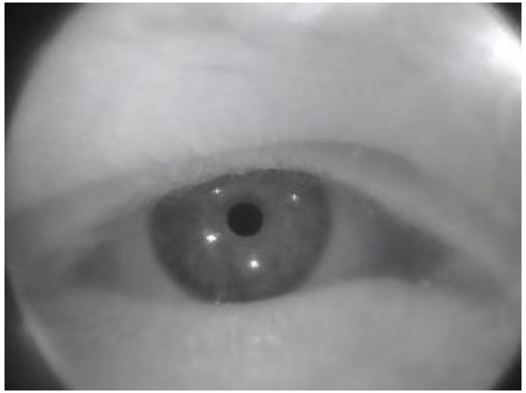
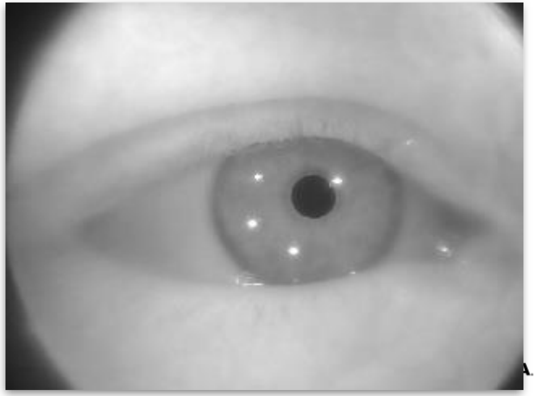
Region-wise
- Contrast scaling
- Blur
- Intensity offset

Global
- Contrast scaling
- Gaussian noise

Augmented Synthetic Image

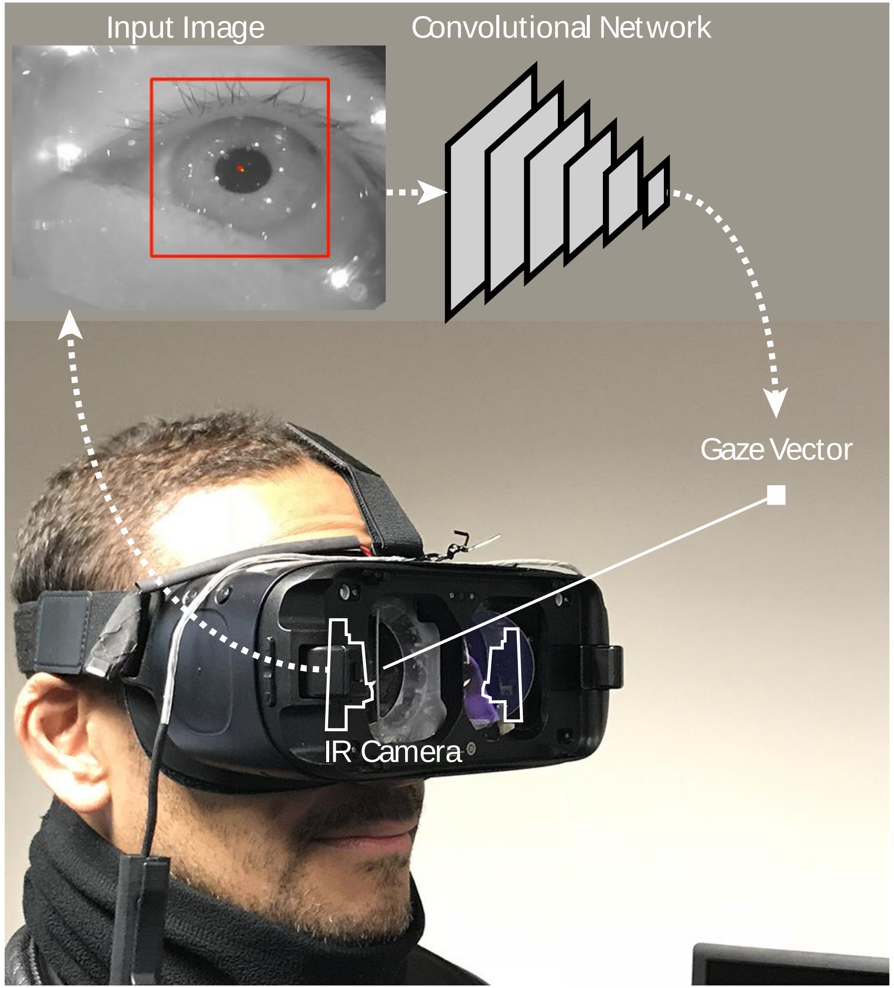


Samples of real images for comparison

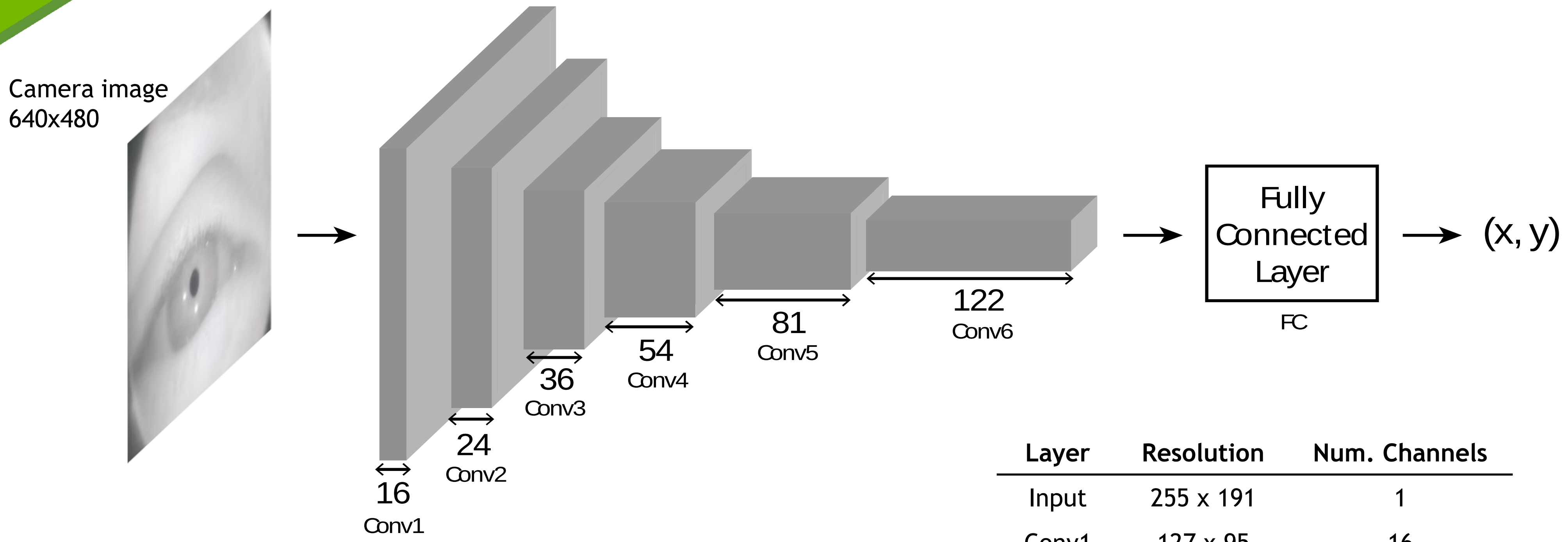





NVGAZE NETWORK

NVGAZE INFERENCE OVERVIEW



NETWORK ARCHITECTURE



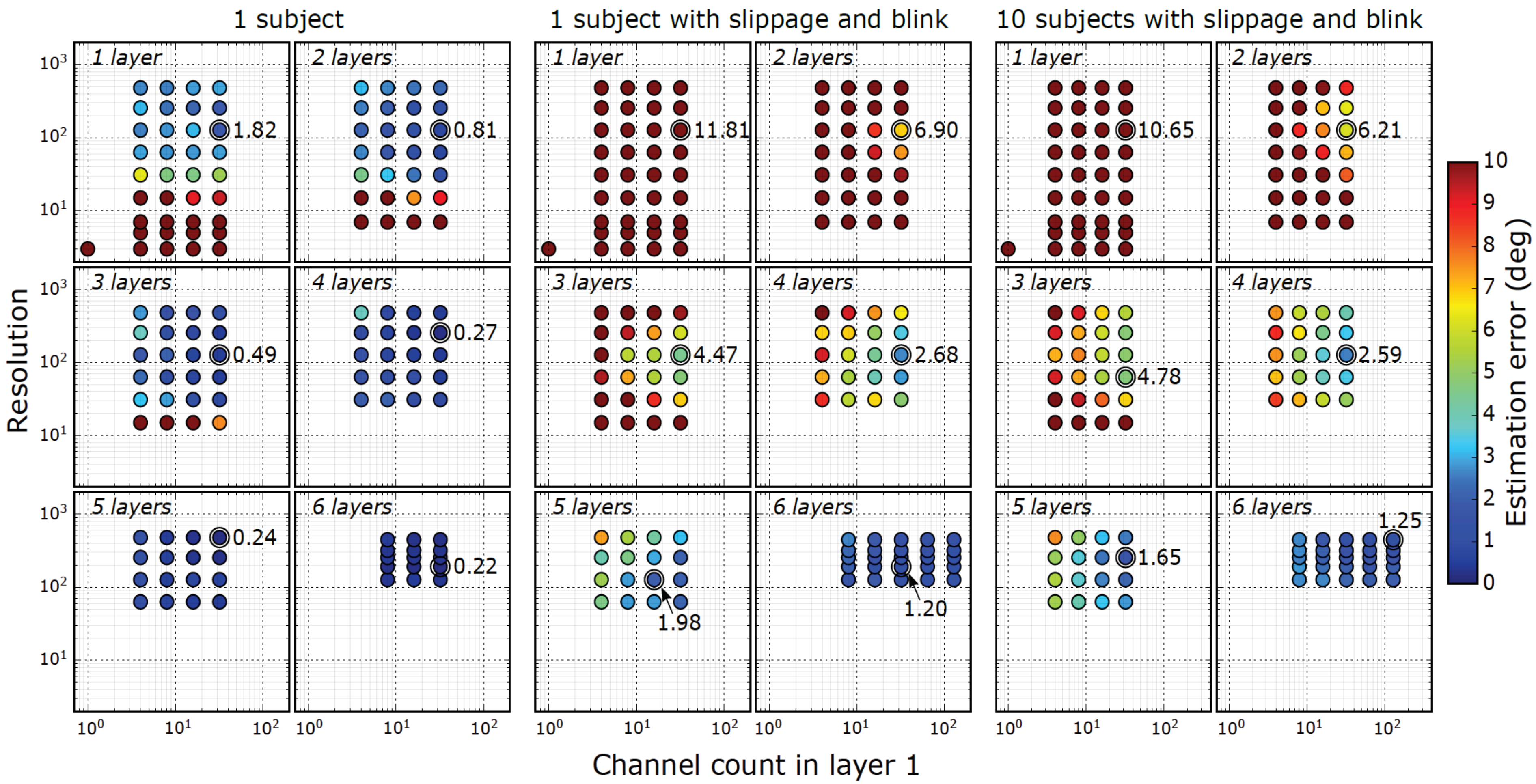
Layer	Resolution	Num. Channels
Input	255 x 191	1
Conv1	127 x 95	16
Conv2	63 x 47	24
Conv3	31 x 23	36
Conv4	15 x 11	54
Conv5	7 x 5	81
Conv6	3 x 2	122

Fully convolutional network

In reference design, each layer has …

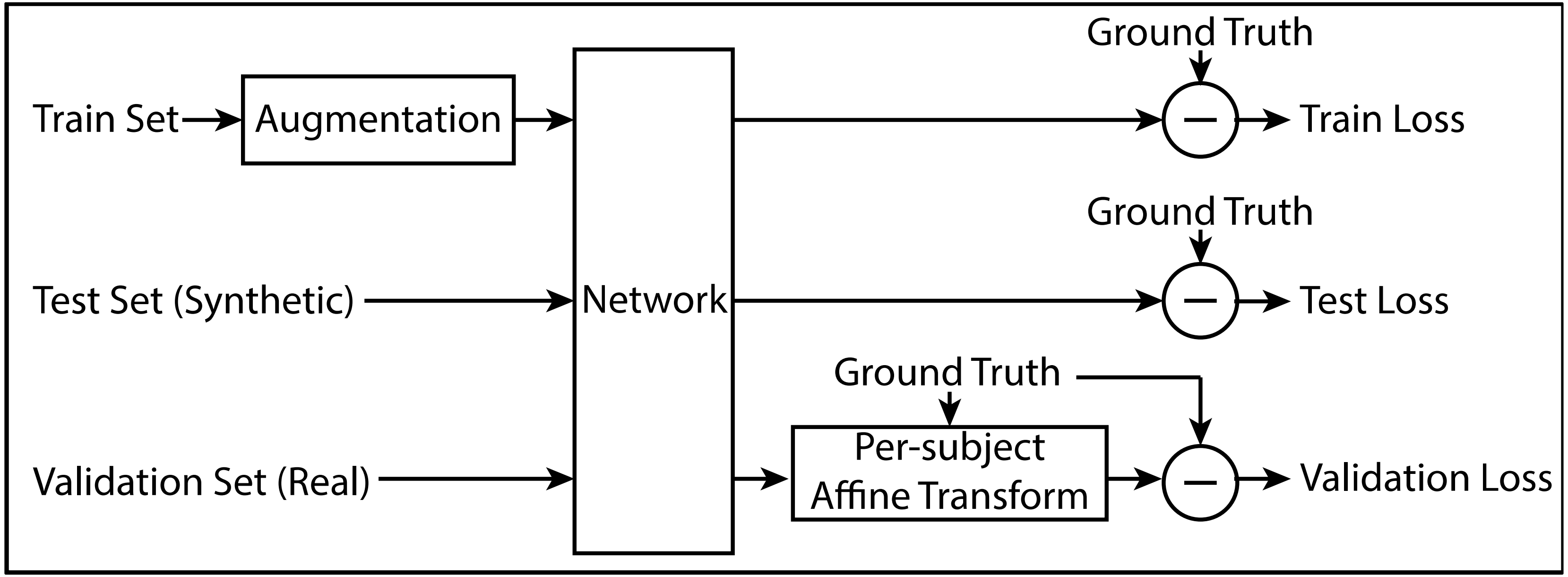
- Stride of 2
- No padding
- 3x3 Conv. kernel

NETWORK COMPLEXITY ANALYSIS



TRAINING AND VALIDATION

Loss function



- Trained on a **10 synthetic subjects + 3 real subjects**. No fine-tuning.
- Ramp-up and ramp-down for **50 epochs** at the beginning and end.
- Adam optimizer with MSE loss

NEURAL NETWORK PERFORMANCE

Gaze Estimation

Accuracy / Near Eye Display

- 2.1 degrees of error in average across real subjects
  - Error is almost evenly distributed across the entire tested visual field
- 1.7 degrees best-case accuracy when trained for single subject

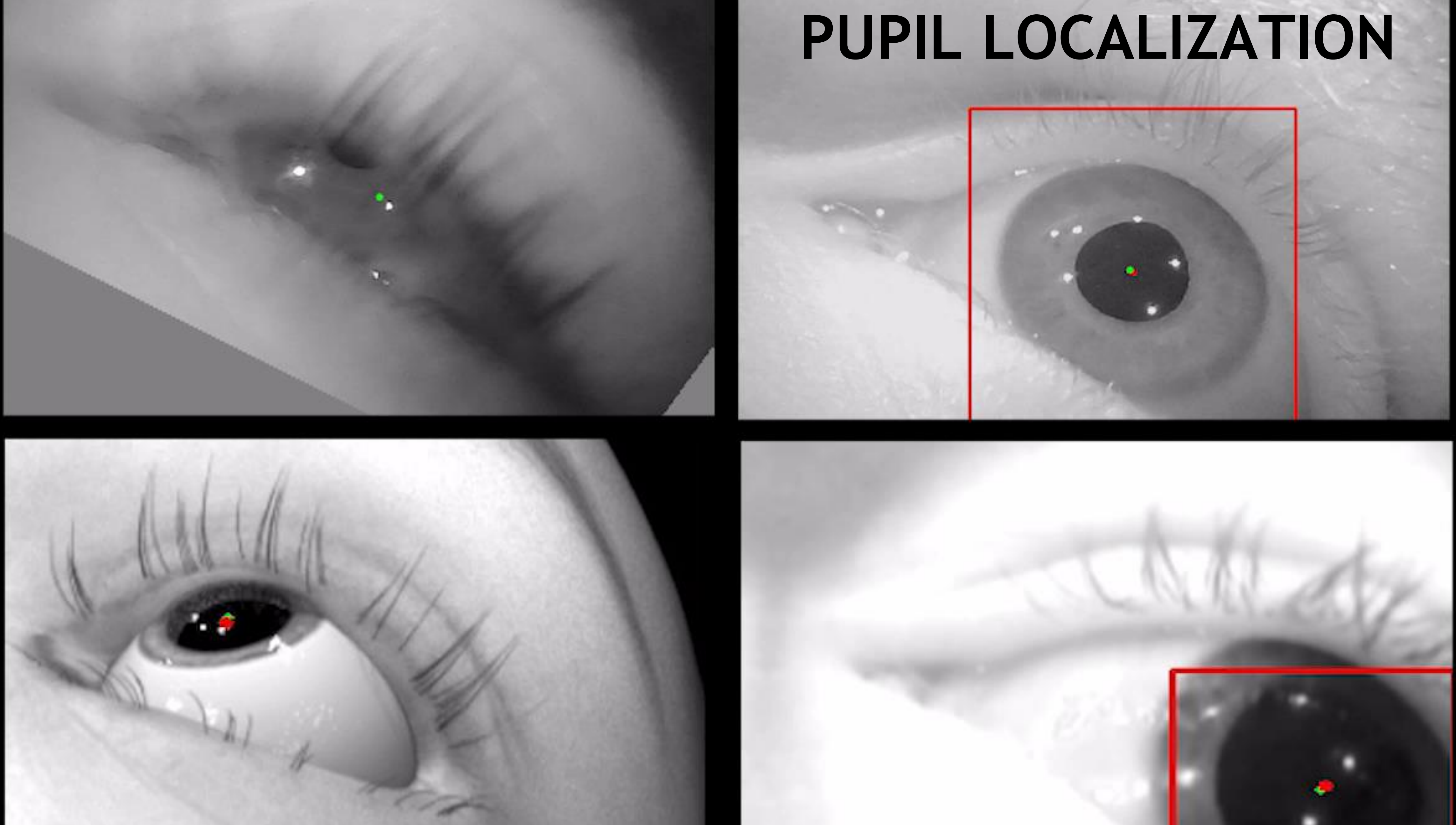
Accuracy / Remote Gaze Tracking

- 8.4 degrees average accuracy for remote gaze tracking (same accuracy as state of the art by Park et al., 2018) but **100x** faster

Latency for gaze estimation

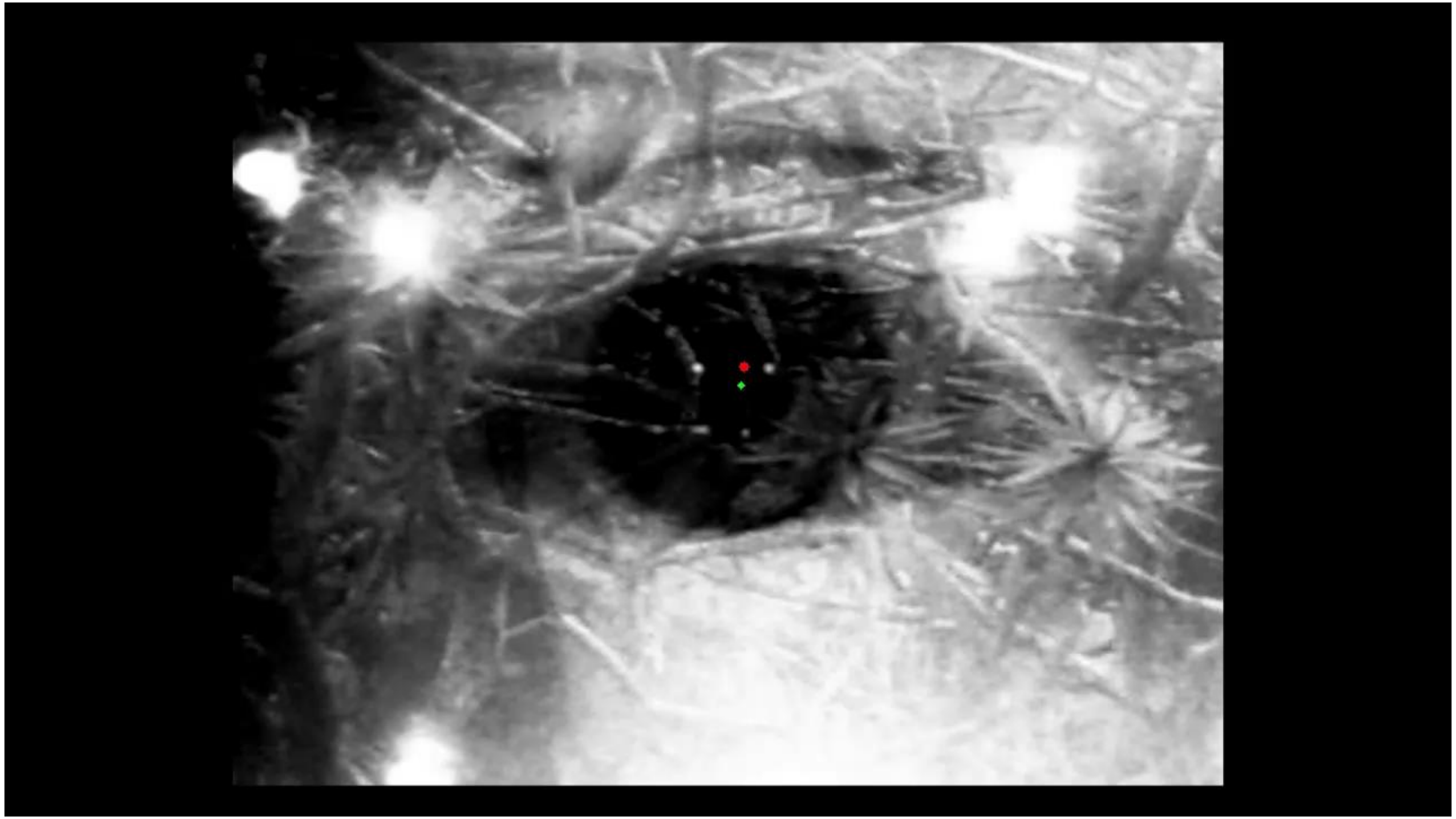
- <1 milliseconds for inference and data transfer between CPU and GPU space
- cuDNN implementation running on TitanV or Jetson TX2
- bottleneck is camera transfer @ 120 Hz

PUPIL LOCALIZATION



NEURAL NETWORK PERFORMANCE

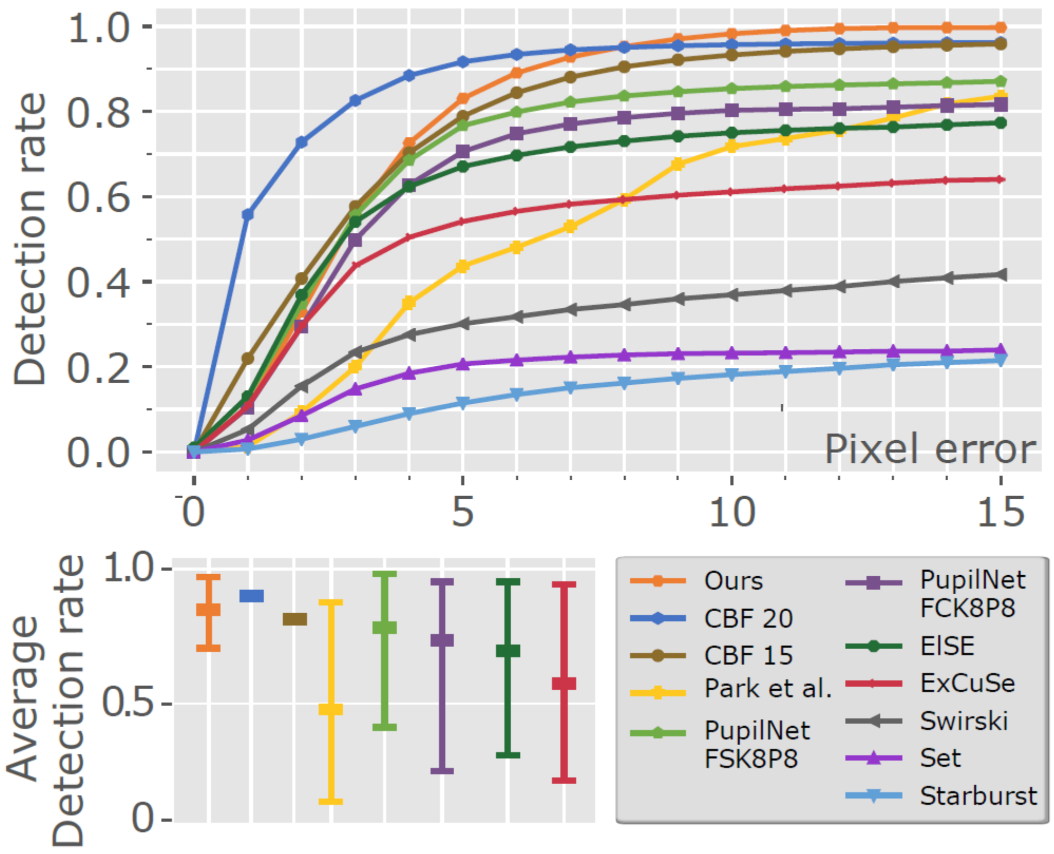
Pupil Location Estimation

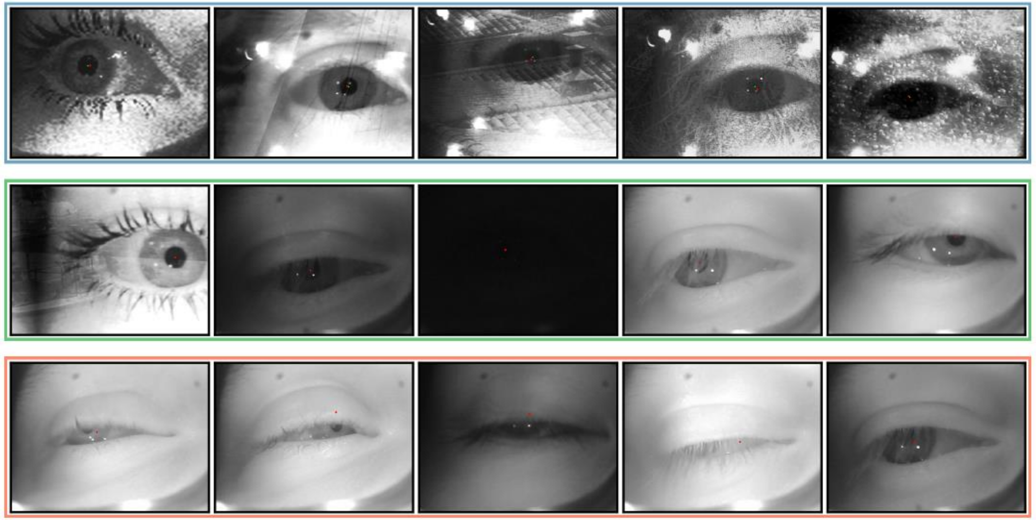



Error = 1.31 px
Error = 4.69 px
Error = 1.40 px
Error = 4.79 px
Error = 0.53 px
Error = 1.05 px
Error = 1.94 px
Error = 3.36 px
Error = 7.38 px
Error = 0.36 px
Error = 3.41 px
Error = 0.86 px
Error = 2.74 px
Error = 2.72 px
Error = 2.83 px
Error = 1.17 px
Error = 3.04 px
Error = 1.44 px
Error = 3.27 px

NEURAL NETWORK PERFORMANCE

Pupil Location Estimation





Layer index	1	2	3	4	5	6	7
Kernel size	9×9	7×7	5×5	5×5	3×3	3×3	3×3
Output channels	24	36	52	80	124	256	512

Our network is more accurate, more robust and requires less memory than others.

OPTIMIZING FOR FAST INFERENCE

Alexander Majercik

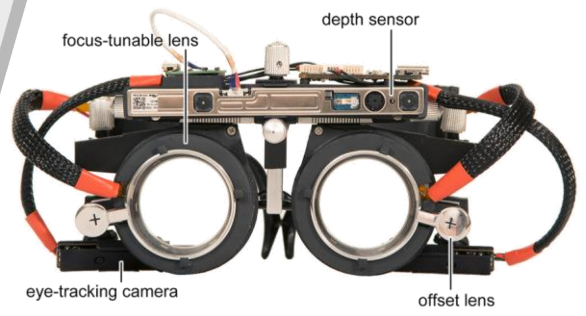
PROJECT GOALS

- Deep learning based gaze estimation
- Higher robustness than previous methods
- Target accuracy is <2 degrees of angular error
- Fast inference ranging in a few milliseconds even on mobile GPU
- Compatibility to any captured input (on-axis, off-axis, near-eye, remote, etc., dark pupil tracking only, glint-free tracking)
- Explore usage of *synthetic* data (large dataset >1,000.000 images)
- Can we learn increase calibration robustness ?

PROJECT GOALS

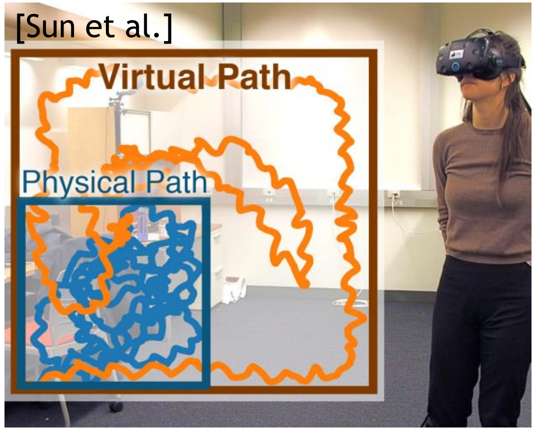
- Deep learning based gaze estimation
- Higher robustness than previous methods
- Target accuracy is <2 degrees of angular error
- Fast inference ranging in a few milliseconds even on mobile GPU
- Compatibility to any captured input (on-axis, off-axis, near-eye, remote, etc., dark pupil tracking only, glint-free tracking)
- Explore usage of *synthetic* data (large dataset >1,000.000 images)
- Can we learn increase calibration robustness ?

NETWORK LATENCY REQUIREMENTS

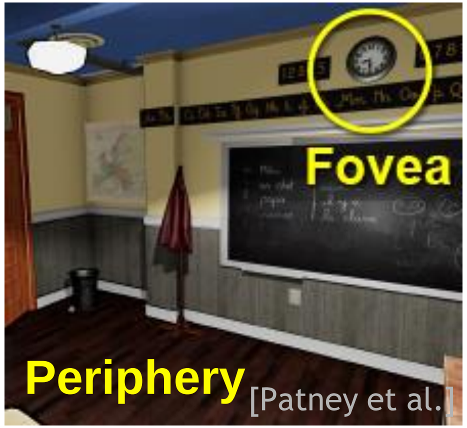


Computational Displays

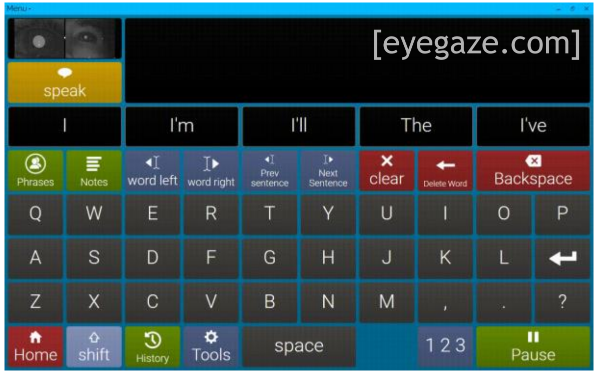
Avatars



Perception



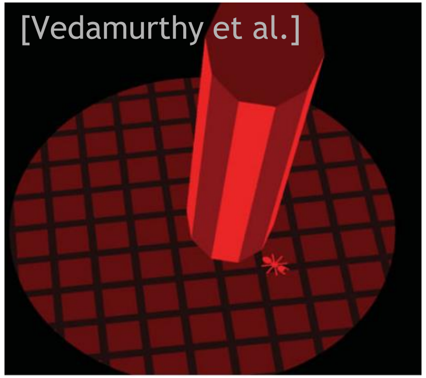
Foveated Rendering
Dynamic Streaming



Gaze Interaction

User State Evaluation

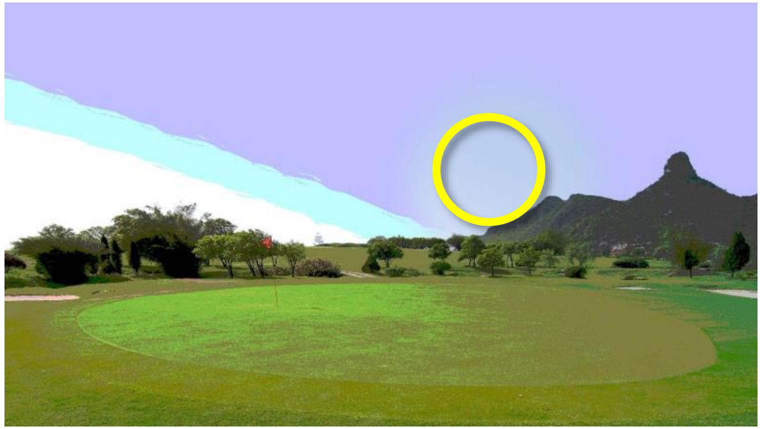
Attention Studies



Health Care

NETWORK LATENCY REQUIREMENTS

Human Perception



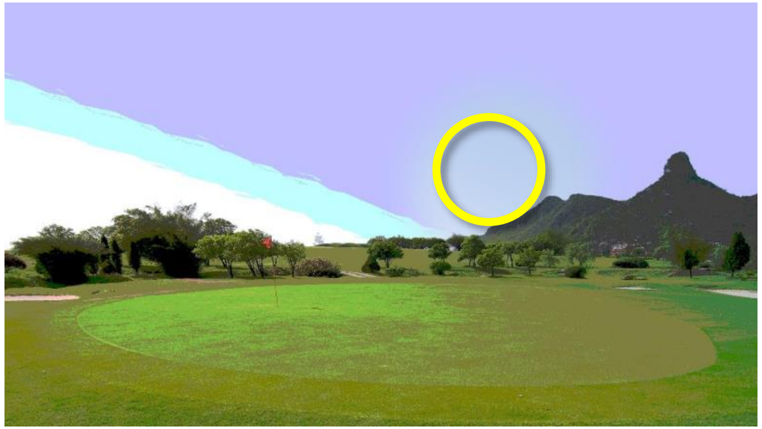
60 ms To Get it Right
Gaze-Contingent Rendering
and Human perception

Esports

Esports Research at NVIDIA

NETWORK LATENCY REQUIREMENTS

Human Perception



60 ms To Get it Right
Gaze-Contingent Rendering
and Human perception

Esports

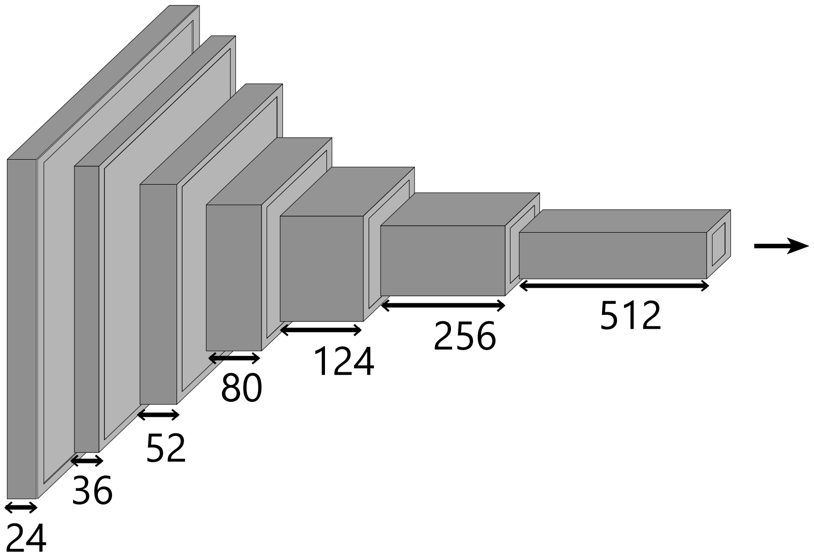
Esports Research at NVIDIA

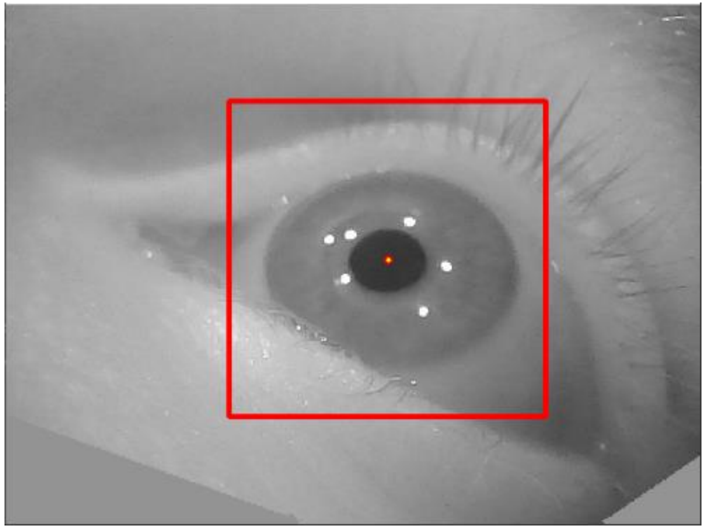
BOTTOM LINE: Network should run in ~1ms!

Fast inference is also *training* problem

NETWORK DESIGN FOR FAST INFERENCE

- 7 layer stacked convolutional network
- Input: 293x293 eye image, Output: pupil position in image space





NETWORK DESIGN FOR FAST INFERENCE

Key Design Decisions

NETWORK DESIGN FOR FAST INFERENCE

Key Design Decisions

- Convolutions and FC layers only

NETWORK DESIGN FOR FAST INFERENCE

Key Design Decisions

- Convolutions and FC layers only
- No max pooling

NETWORK DESIGN FOR FAST INFERENCE

Key Design Decisions

- Convolutions and FC layers only
- No max pooling
- ReLU activation

NETWORK DESIGN FOR FAST INFERENCE

Key Design Decisions

- Convolutions and FC layers only
- No max pooling
- ReLU activation
- Data-directed approach

NETWORK DESIGN FOR FAST INFERENCE

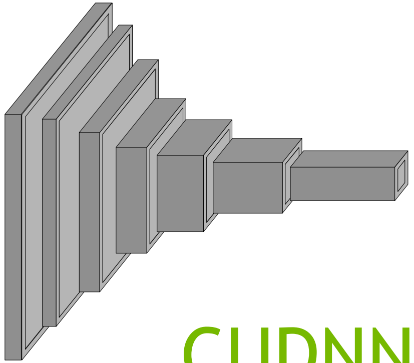
Data-directed approach

NVIDIA GTC

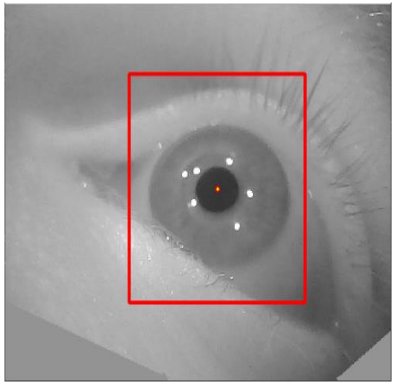
Better Training -> Simpler Network -> Run Faster

```
checkCUDNN(cudnnConvolutionBiasActivationForward(
    m_cudnn,                        // Handle to cuDNN context
    &alpha,                         // Scaling factor
    m_input->getTensorDesc(),       // Input tensor description
    m_input->getData_fp32(),        // Input tensor data
    m_filterDesc,                   // Filter description
    m_weights_fp32,                 // Weights
    m_convDesc,                     // Convolution description
    m_convAlgo,                     // Convolution algorithm
    workspace,                      // Workspace
    workspaceSize,                  // Workspace size
    &beta,                          // Scaling factor
    m_outTensorDesc,                // Optional added tensor description
    getData_fp32(),                 // Optional added tensor data
    m_biasTensor,                   // Bias description
    m_bias_fp32,                    // Bias data
    m_activation,                   // Activation function
    m_outTensorDesc,                // Output tensor description
    getData_fp32()));               // Output data
```

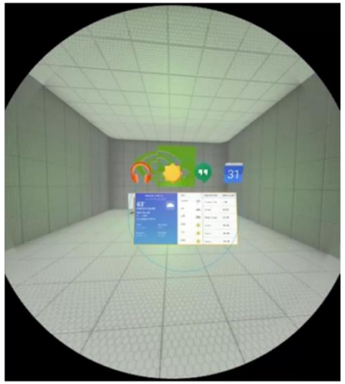
CPU



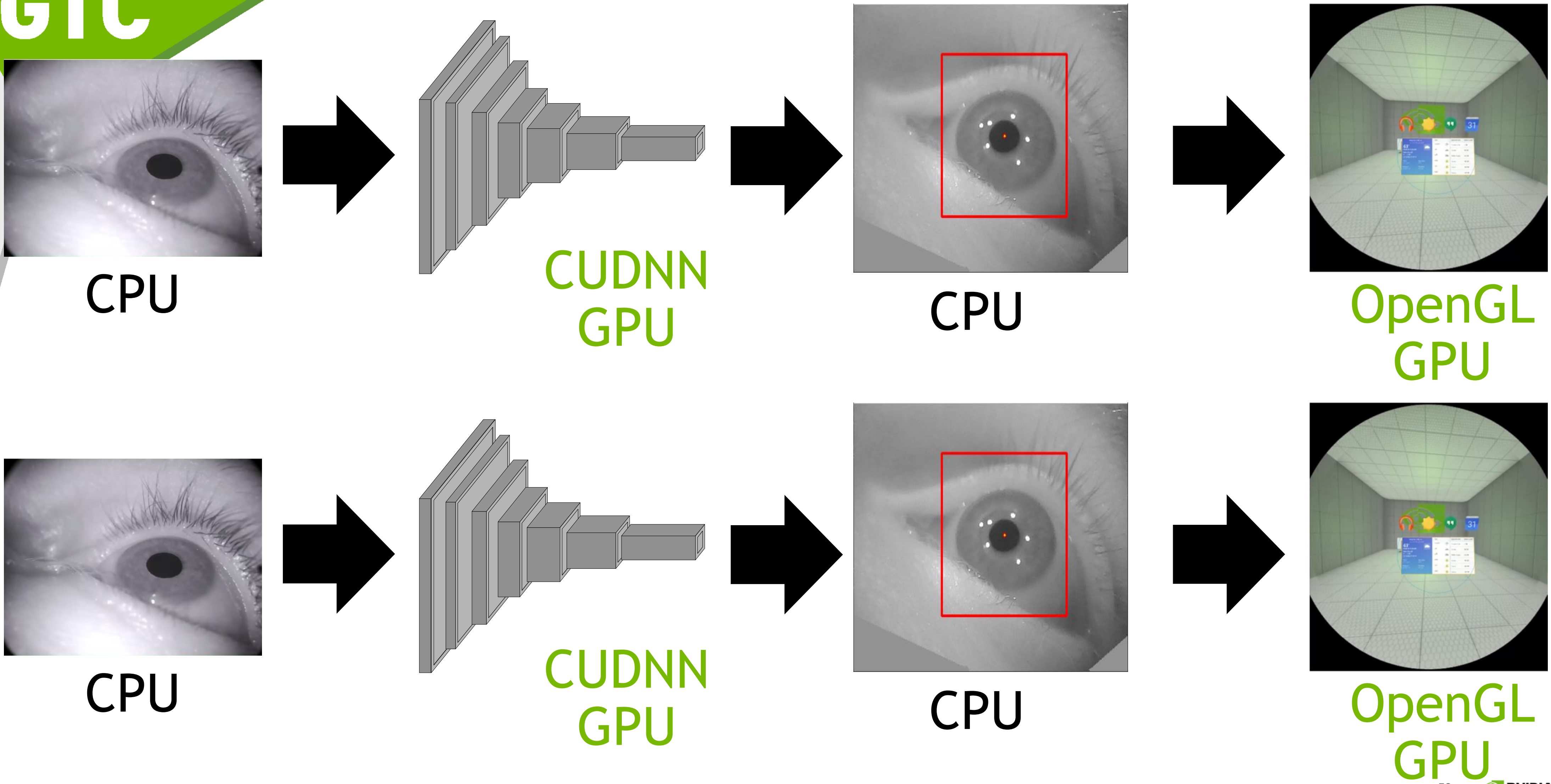
CUDNN
GPU



CPU



OpenGL
GPU


CPU
CUDNN
GPU
CPU
OpenGL
GPU
CPU
CUDNN
GPU
CPU
OpenGL
GPU

FAST INFERENCE WITH NVIDIA CUDNN

Optimizing the pipeline

- GPU Programming Best Practices

FAST INFERENCE WITH NVIDIA CUDNN

Optimizing the pipeline

- GPU Programming Best Practices:
  - Minimize CPU-GPU copy

FAST INFERENCE WITH NVIDIA CUDNN

Optimizing the pipeline

- GPU Programming Best Practices:
  - Minimize CPU-GPU copy
  - Minimize kernel launches (pack work into your kernels efficiently)

FAST INFERENCE WITH NVIDIA CUDNN

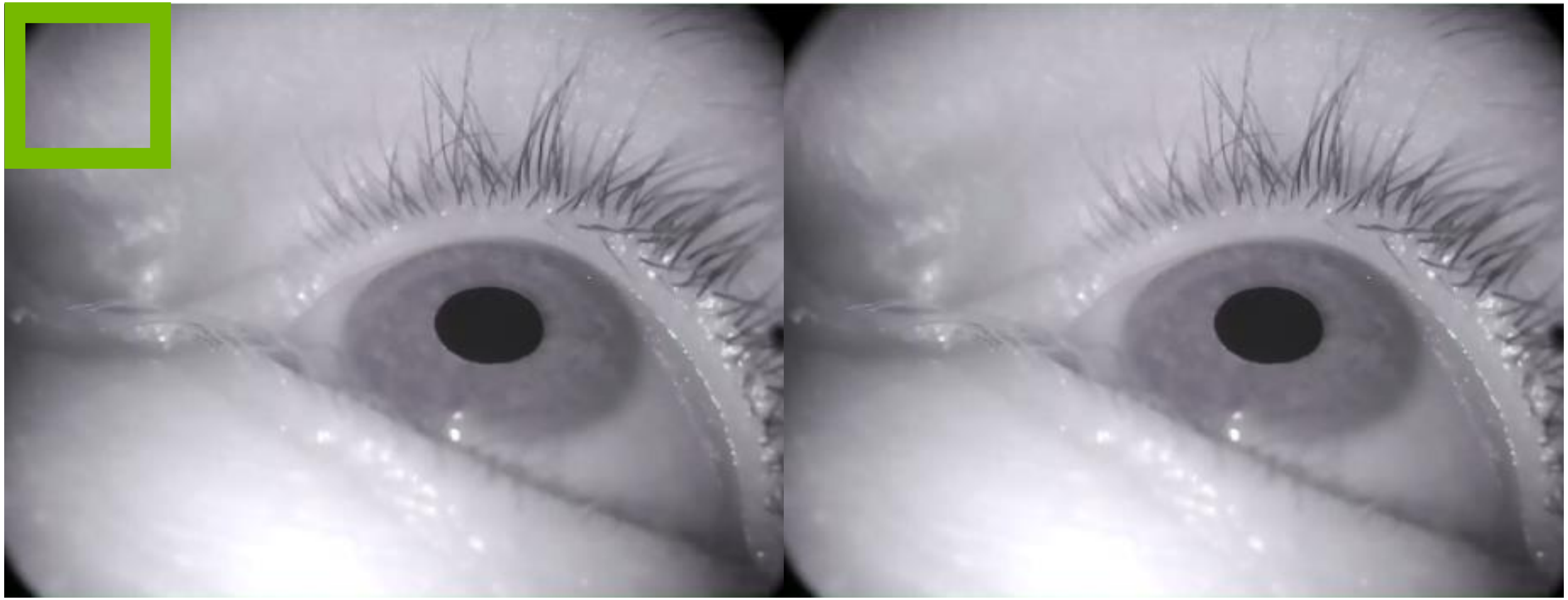
Optimizing the pipeline

- GPU Programming Best Practices:
  - Minimize CPU-GPU copy
  - Minimize kernel launches (pack work into your kernels efficiently)
- To do both…combine the eye images into a single pass!

FAST INFERENCE WITH NVIDIA CUDNN

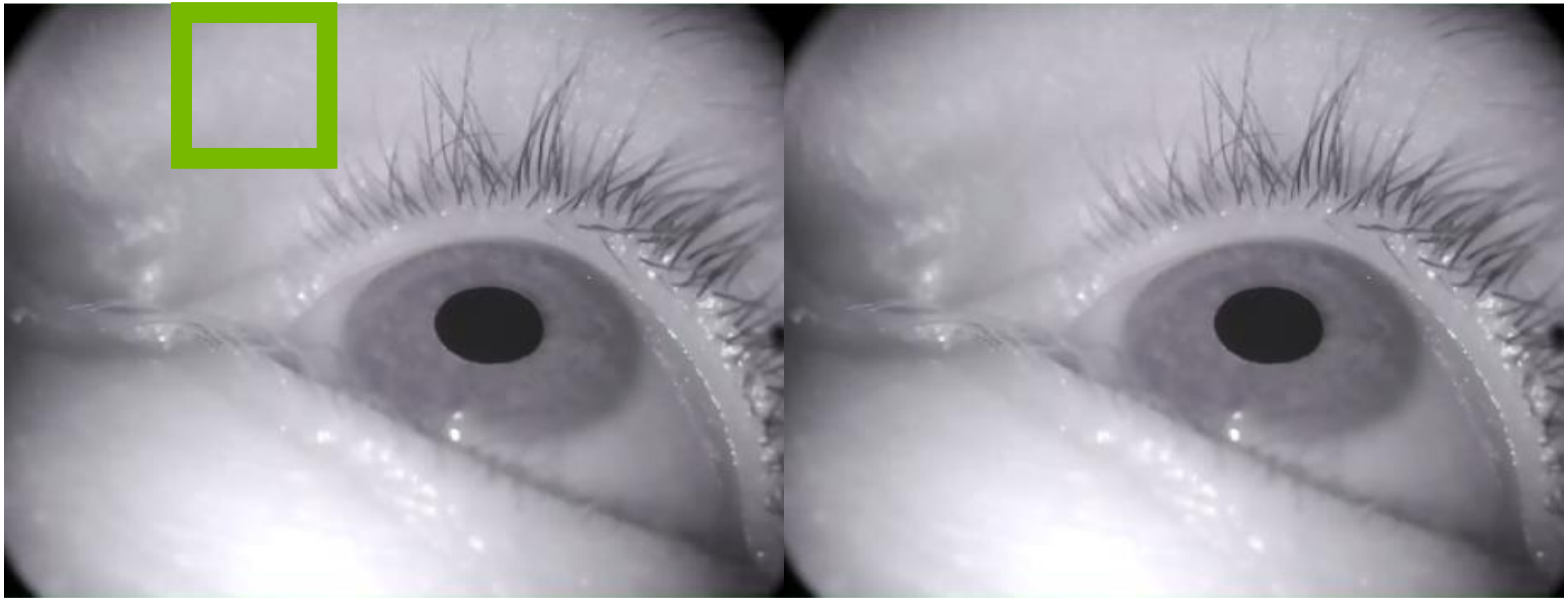
Merging the input images

Convolution kernel



FAST INFERENCE WITH NVIDIA CUDNN

Merging the input images

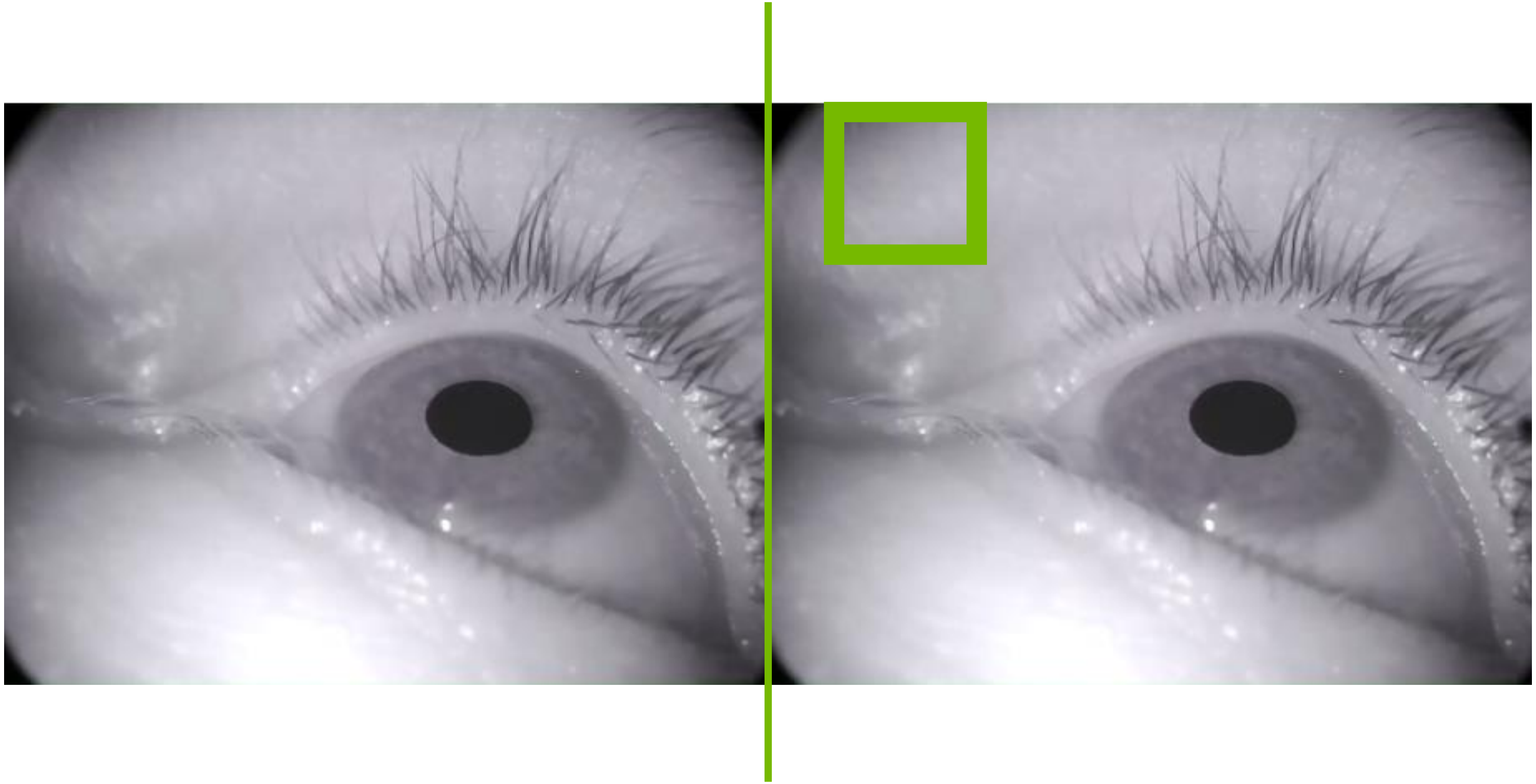


FAST INFERENCE WITH NVIDIA CUDNN

Merging the input images

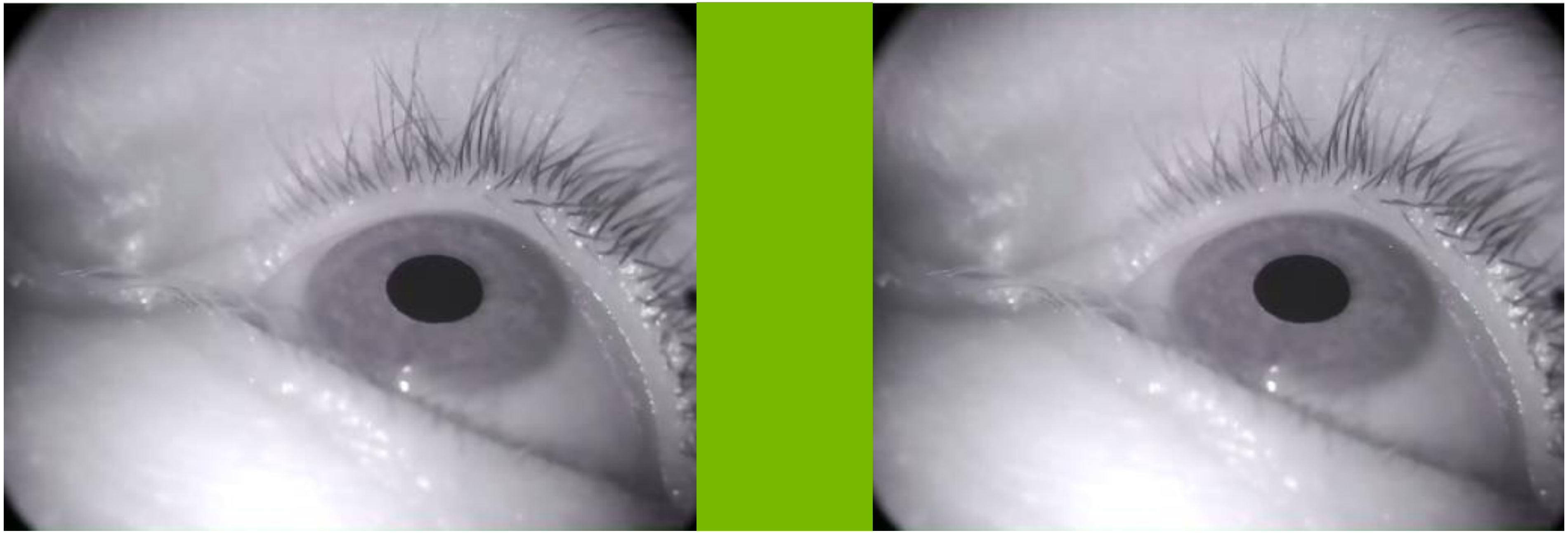
FAST INFERENCE WITH NVIDIA CUDNN

Merging the input images

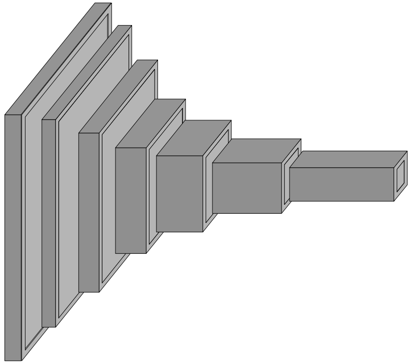


FAST INFERENCE WITH NVIDIA CUDNN

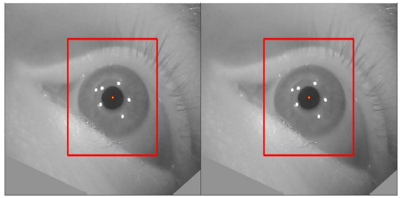
Merging the input images



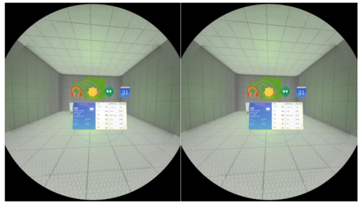
CPU



CUDNN
GPU



CPU



OpenGL
GPU

FAST INFERENCE WITH NVIDIA CUDNN

Results

Method	Time (ms)
Single Image (Python based DL framework)	
Single Image (cuDNN)	
Concatenated input (cuDNN)	

FAST INFERENCE WITH NVIDIA CUDNN

Results

Method	Time (ms)
Single Image (Python based DL framework)	~6
Single Image (cuDNN)	
Concatenated input (cuDNN)	

FAST INFERENCE WITH NVIDIA CUDNN

Results

Method	Time (ms)
Single Image (Python based DL framework)	~6
Single Image (cuDNN)	0.748
Concatenated input (cuDNN)	

FAST INFERENCE WITH NVIDIA CUDNN

Results

Method	Time (ms)
Single Image (Python based DL framework)	~6
Single Image (cuDNN)	0.748
Concatenated input (cuDNN)	1.022

SUMMARY

- Network Latency Requirements
  - Foveated rendering, human perception esports
  - Network has to execute in ~1ms!
- Network Design for Fast Inference (During Training!)
  - Simple network (stacked convolution, no max pooling, relu)
  - Complexity is in the *data!*
- Fast Inference Using NVIDIA cuDNN
  - Follow GPU best practices to optimize your pipeline around your well-designed network

Try the NvGaze Demo:

VR Theater
SJCC Expo Hall 3, Concourse Level

Tuesday: 12:00pm - 7:00pm
Wednesday: 12:00pm - 7:00pm
Thursday: 11:00am - 2:00pm

REFERENCES

NVGaze: An Anatomically-Informed Dataset for Low-Latency, Near-Eye Gaze Estimation [Kim'19]

Adaptive Image-Space Sampling for Gaze-Contingent Real-time Rendering [Stengel'16]

Perception-driven Accelerated Rendering [Weier'17]

Visualization and Analysis of Head Movement and Gaze Data for Immersive Video in Head-mounted Displays [Loewe'15]

Subtle gaze guidance for immersive environments [Grogorick '17]

Towards virtual reality infinite walking: dynamic saccadic redirection [Sun '18]

Q&A

Michael Stengel

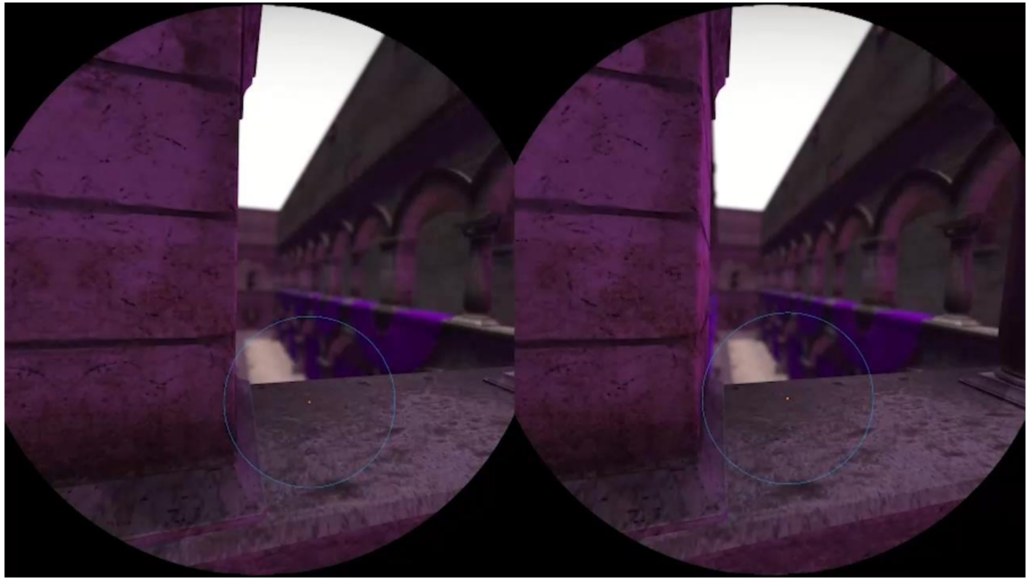
New Experiences Group

mstengel@nvidia.com

Alexander Majercik

New Experiences Group

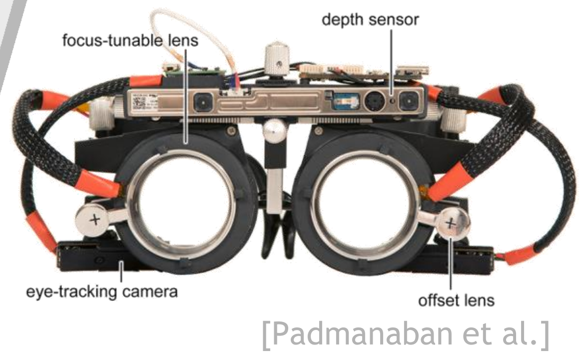
amajercik@nvidia.com



Try out our demo in the Exhibitor Hall !

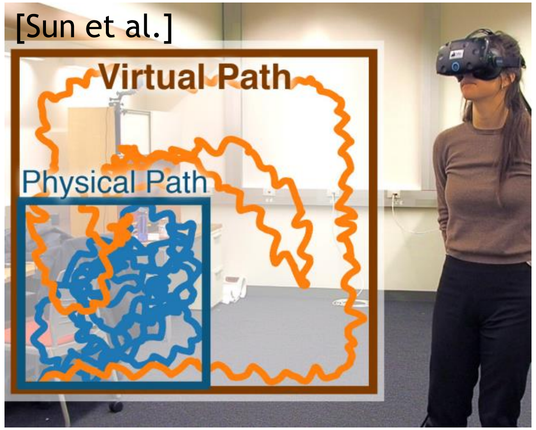
Dataset and model available at sites.google.com/nvidia.com/nvgaze

EYE TRACKING IN VR/AR

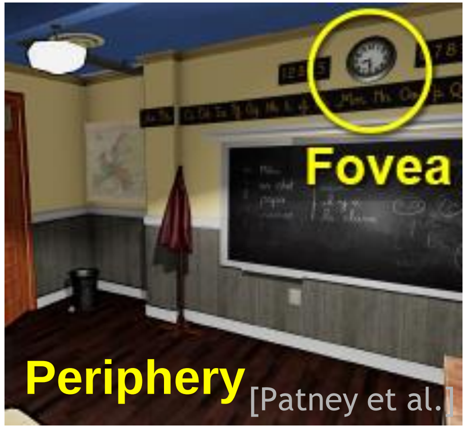


Computational Displays

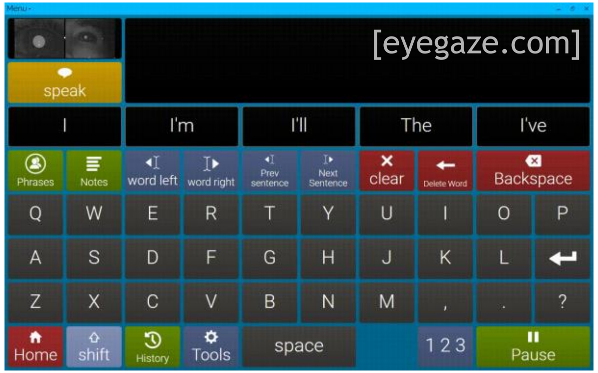
Avatars



Perception



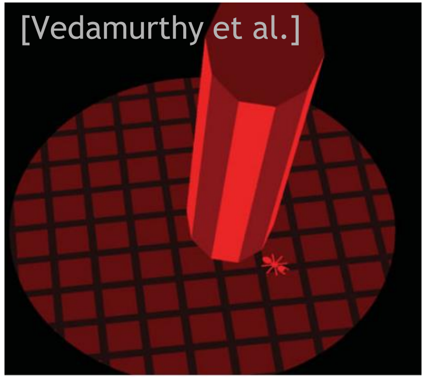
Foveated Rendering
Dynamic Streaming



Gaze Interaction

User State Evaluation

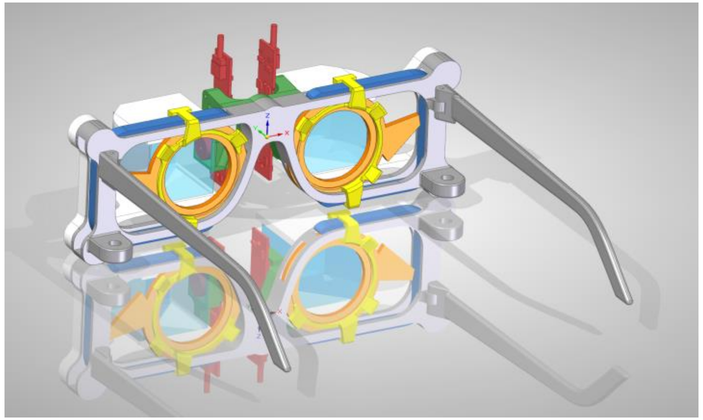
Attention Studies



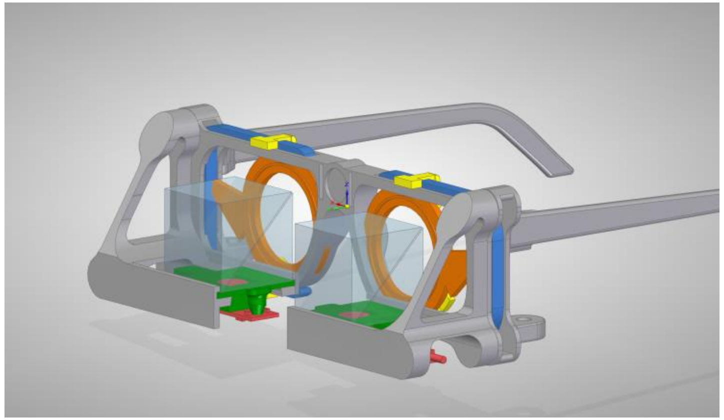
Health Care

ON-AXIS GAZE TRACKING GLASSES

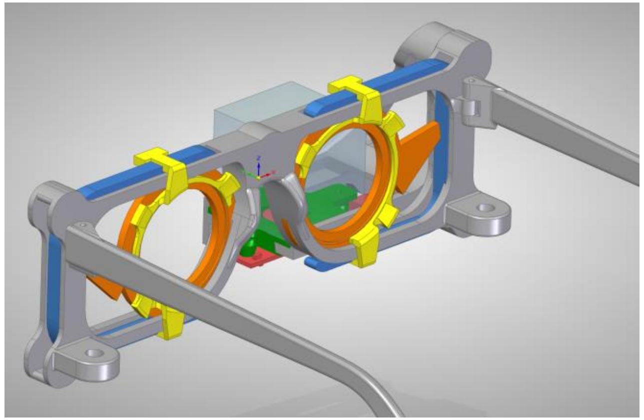
Eye tracking prototype for Augmented Reality glasses



Vertical beam splitter



Horizontal beam splitter

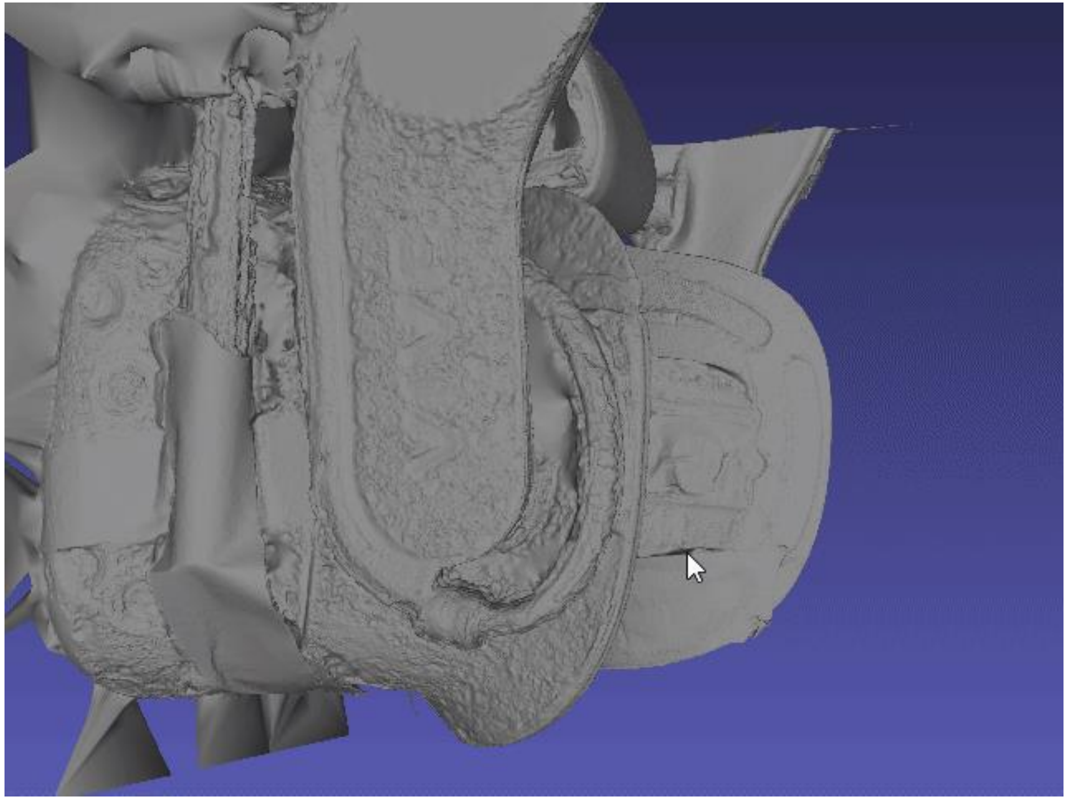


Infared illumination units

Gaze tracking glasses with vertical/horizontal waveguides

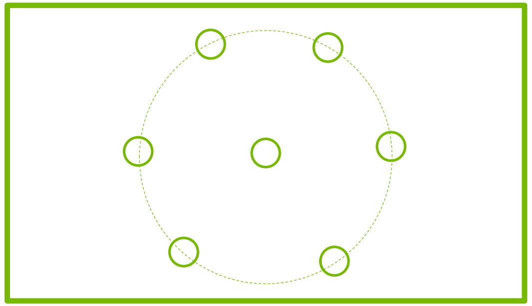
OFF-AXIS GAZE TRACKING

3D Reconstruction Result



GAZE CALIBRATION

- Sparse Pattern sampling (e.g. ring pattern), average over time



Ring target pattern

Calibration Method A – Using calibration network layer

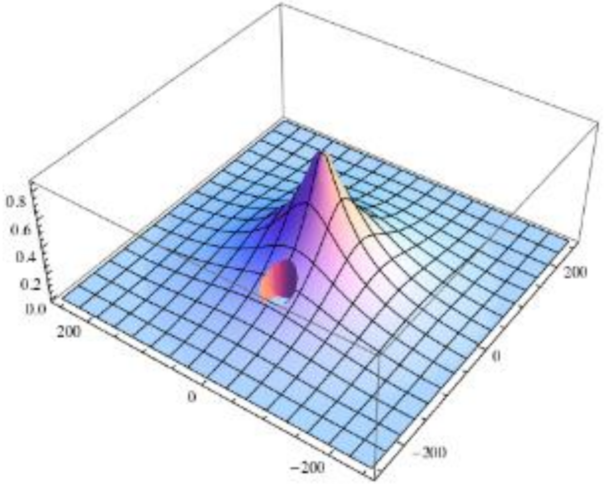
- calibration sets layer weights
- 3d gaze direction directly estimated by network inference

Calibration Method B - Mapping 2d pupil center to 2d screen position

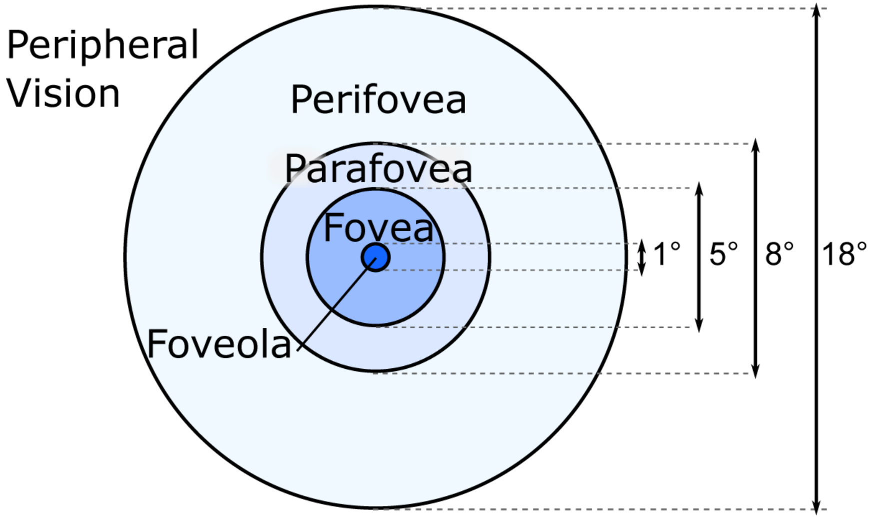
- calibration estimates polynomial mapping functions F_L and F_R
- localized pupil centers (network inference) are mapped using F_L and F_R
- derive 3d gaze vector from binocular 2d screen positions

FOVEATED RENDERING

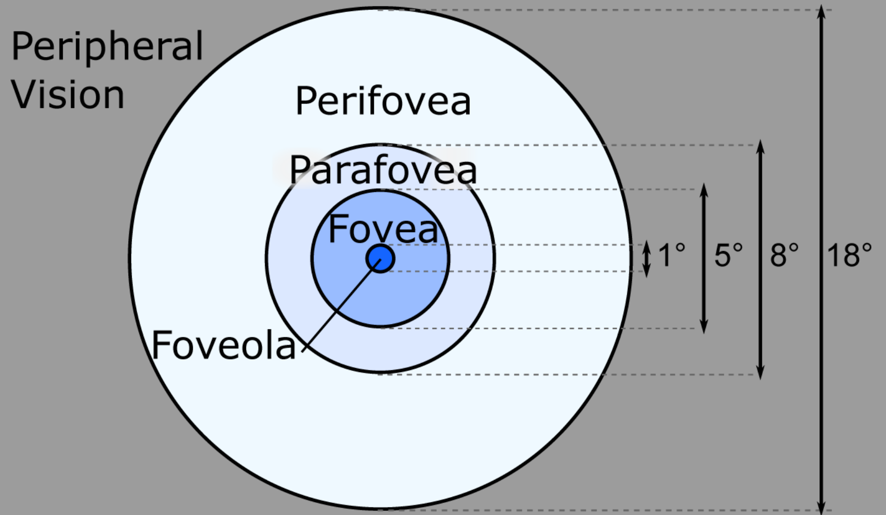
Accelerating Real-time Computer Graphics



Retinal Cone Distribution
[Goldstein,2007]



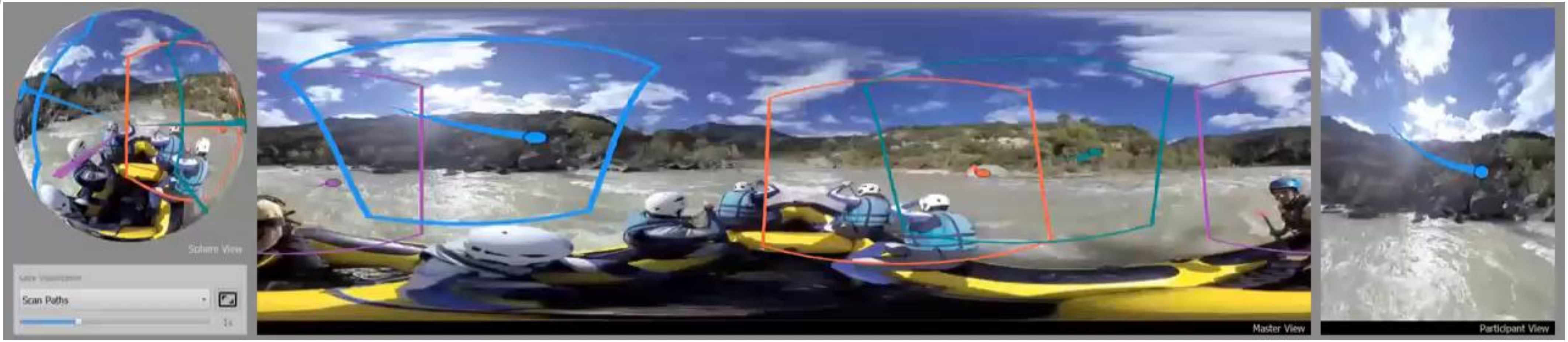
FOVEAL REGION



APPLICATION EXAMPLE FOVEATED RENDERING

ATTENTION ANALYSIS

Generating 3D Saliency Information



[Loewe and Stengel et al. ETVIS'15]