GTC 2017

Acceleration of Multi-Object Detection and Classification Training Process

with NVidia IRay SDK

Tatiana Surazhsky, Ilya Shapiro, Leonid Bobovich, Michael Kemelmakher

SAP Innovation Center Israel
SAP Labs Israel

Outline

Introduction

Previous work

Rendering workflow – IRay SDK for annotation and rendering

Specialization and finetuning

Deep learning and “synthetic features”

Conclusions

Introduction

Our task: Brand Impact – detect brands exposure in videos/images

Our method: Deep CNN

- Training
- Inference

This talk: employ IRay Rendering Engine & SDK

- Automatically **generate images** for training stage
- Automatically **annotate objects of interest**

Introduction – cont'd

"One needs only to buy a **GPU**, arm oneself with **enough training data**,
and turn the crank to see head-spinning improvements on most computer vision benchmarks."

– from "Learning Dense Correspondence via 3D-guided Cycle Consistency" by T. Zhou et. al.

Hardware set up:

- NVidia GDX-1
- NVidia Tesla P100 (pascal architecture)
- NVLink with 8 GPU

Enough training data:

- Need annotated images!
- Mostly done manually… bottleneck!

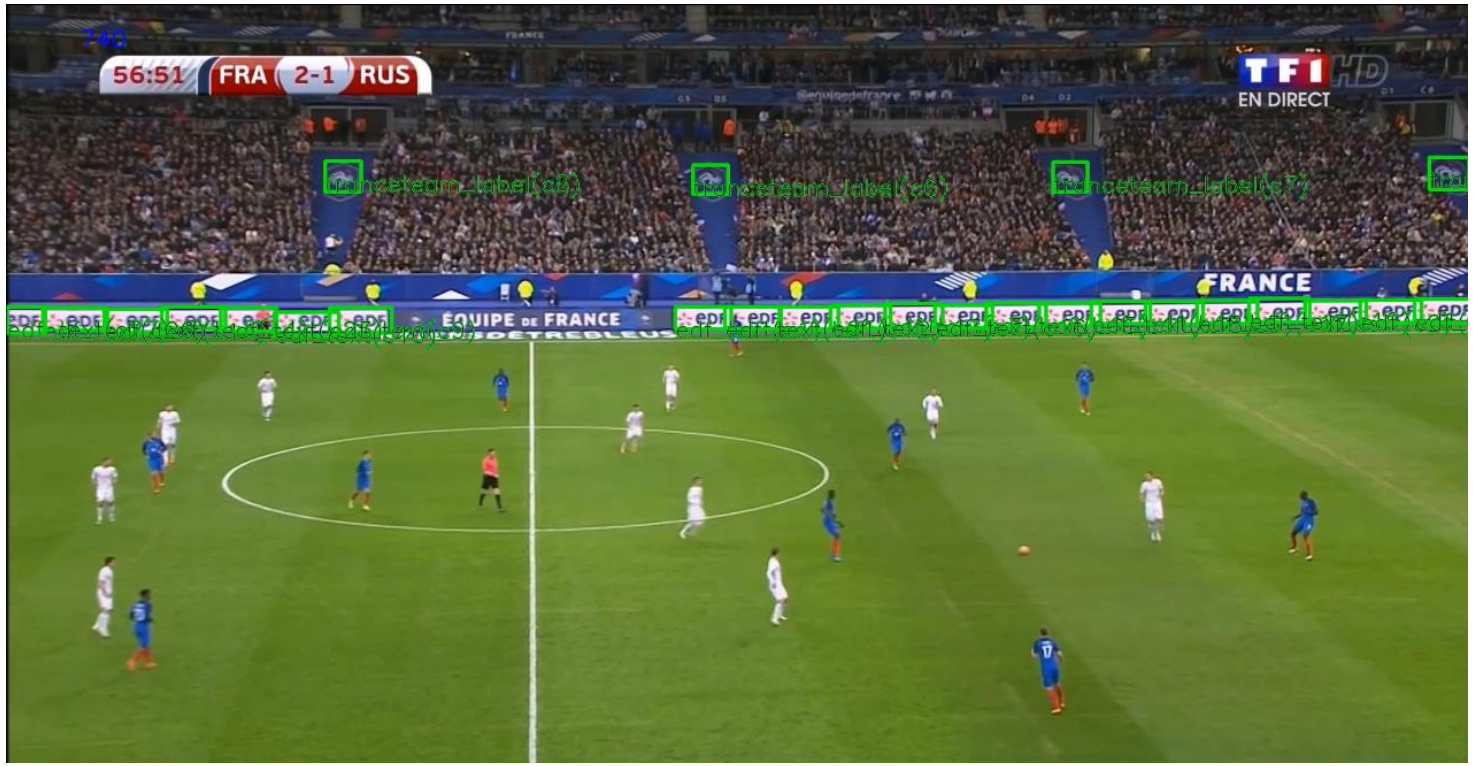
Motivation – Problem to solve

- We are looking at example of multi-object detection and classification task
- State of the art: deep learning
  - Classification
  - Localization
  - Detection
  - Segmentation
- Brand Intelligence example
  - Detect multiple objects (logo)
  - Gather statistics
    - What
    - When
    - Where

CNN bottleneck: manual work

CNN input:

- Train with “ground truth”: labeled images
- More training data -> better results
  - How much more?
    Hundreds thousands
    annotated images are good…
- *Better* training data?
- Annotation example
- Solution?




Allianz
Gigaset
SAP
Lufthansa
Audi Q3
Continental
adidas
koffer-direkt.de

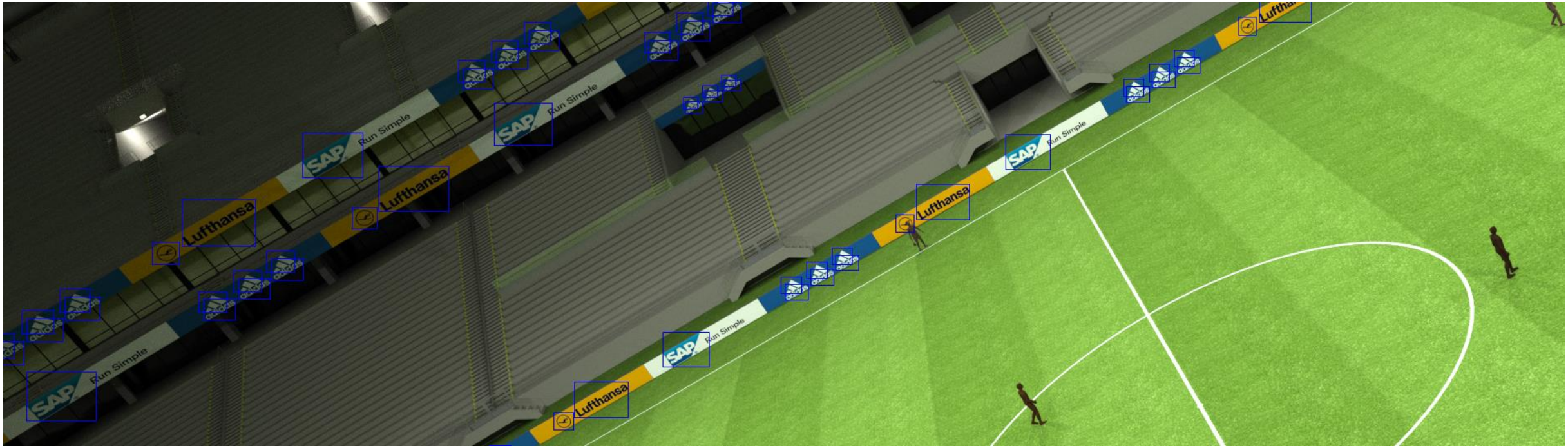
Previous work

- Playing for data: ground truth from computer games (S. Richter et. al.)
  - Solve segmentation problem
  - Semi-automatic
- Render for CNN: viewpoint estimation in images using CNNs trained with rendered 3D model views (Hao Su et.al.)
  - ShapeNet: large 3D object library
  - Vary texture
  - Vary background
  - Different crops
- Learning dense correspondence via 3D-guided cycle consistency (T. Zhou et.al.)
- Detection and Classification of Multiple Objects using an RGB-D Sensor and Linear Spatial Pyramid Matching (M. Dimitriou et.al.)

Rendering a 3D scene

How do you get large task-specific data set?

- Photorealistic rendered scene can look like an image from camera
- While rendering you **know** what you render -> Annotation is “for free”



Rendering engines utilizing NVidia GPUs

From NVidia site: GPU ray tracing partners

- V-Ray – very popular, utilizers CPU better than GPU
- Octane – utilizes CUDA better than OpenCL
- Arion – path tracing render engine
- furryball RT – real time ray tracer
- Indigo – simulates physics of light
- thea presto render
- Moskito – no proprietary materials, works with Autodesk library
- Redshift – biased render engine

NVidia Rendering engines

Mental Ray

Photo realistic, physically based rendering engine

Used in film, broadcast, digital publishing & design visualizations

Very flexible custom shaders

IRay “in 3 parts”

Photo realistic, physically based

IRT – interactive, converging to almost photo realistic
(different number of iterations and less ray splits)

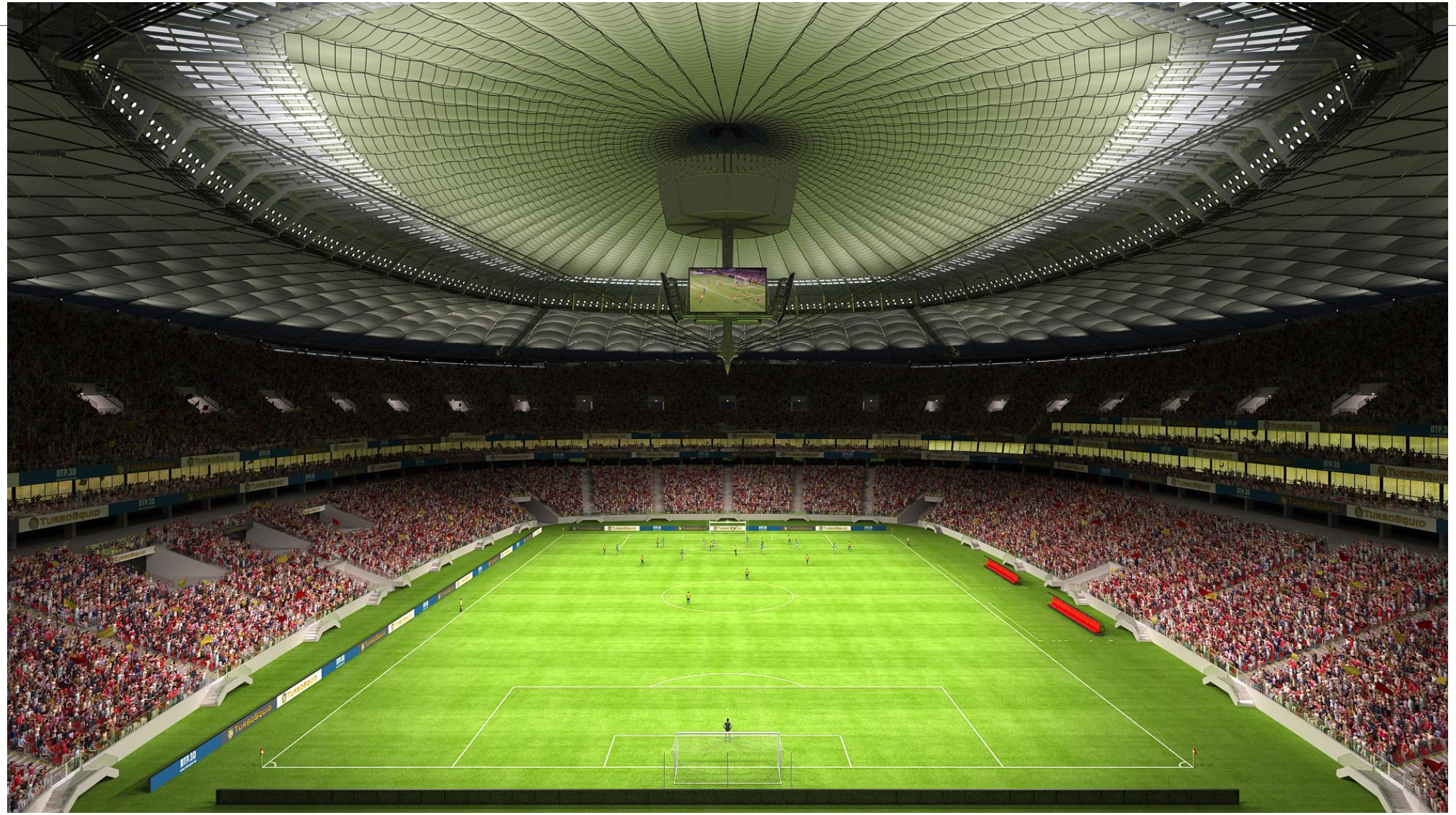
Real time – OpenGL based

- Too many lights present a problem for RT and IRT

Rendering engines for photorealistic results – IRay

- Benefits over other engines (for our purposes)
  - Works with NVidia hardware (well)
  - Server queuing (VCA) – rendered images can be directly passed to the training nets
  - Platform independent **C++ SDK**
  - IViewer with **source code available** and platform agnostic
  - IViewer is meant for developers:
    easy to extend and modify

Original scene



Workflow

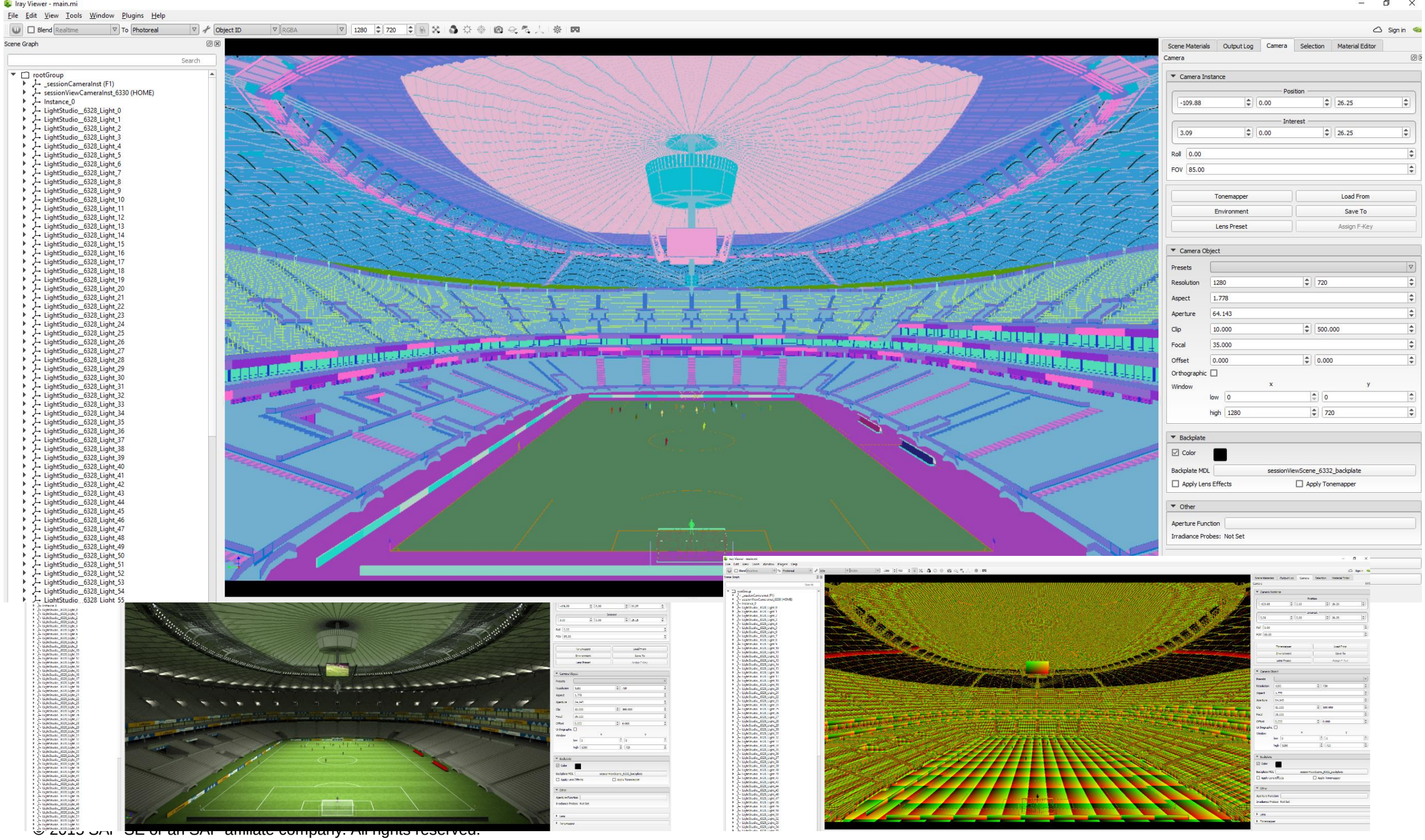
1. Obtain a relevant 3D scene
   a) Commercially obtained, TurboSquid stadium model
   b) Created with 3Ds Max studio, V-Ray materials
   c) Consists of more than 1 500 000 polygons

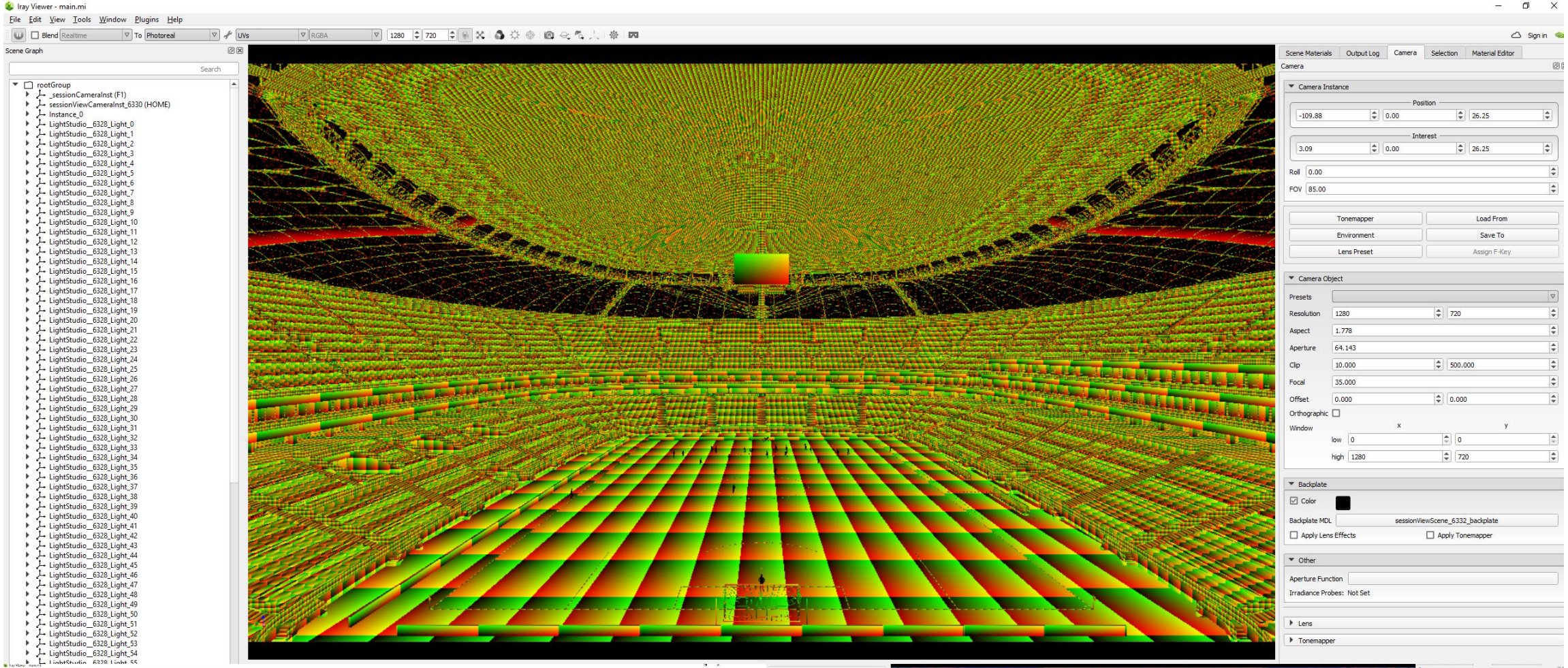
2. Render photo realistic images for training – simulated data set

3. Annotate images:
   for each entity instance
   a) define type (class)
   b) Location and size (x, y, width, height)


Lufthansa
SAP
Run Simple


Allianz
Lufthansa
Audi Q3




Iray Viewer - main.mi
File Edit View Tools Window Plugins Help
Blend Realtime To Photoreal UVs RGBA 1280 720
Sign in
Scene Graph
Search
rootGroup
_sessionCameraInst (F1)
sessionViewCameraInst_6330 (HOME)
Instance_0
Scene Materials
Output Log
Camera
Selection
Material Editor
Camera
Camera Instance
Position
-109.88 0.00 26.25
Interest
3.09 0.00 26.25
Roll 0.00
FOV 85.00
Tonemapper
Load From
Environment
Save To
Lens Preset
Assign F-Key
Camera Object
Presets
Resolution 1280 720
Aspect 1.778
Aperture 64.143
Clip 10.000 500.000
Focal 35.000
Offset 0.000 0.000
Orthographic
Window
x y
low 0 0
high 1280 720
Backplate
Color
Backplate MDL sessionViewScene_6332_backplate
Apply Lens Effects
Apply Tonemapper
Other
Aperture Function
Irradiance Probes: Not Set
Lens
Tonemapper

Determine object of interest

Adidas: text and triangular image

SAP logo

Lufthansa: text and logo

Occlusions

Do we want to ignore occlusions?

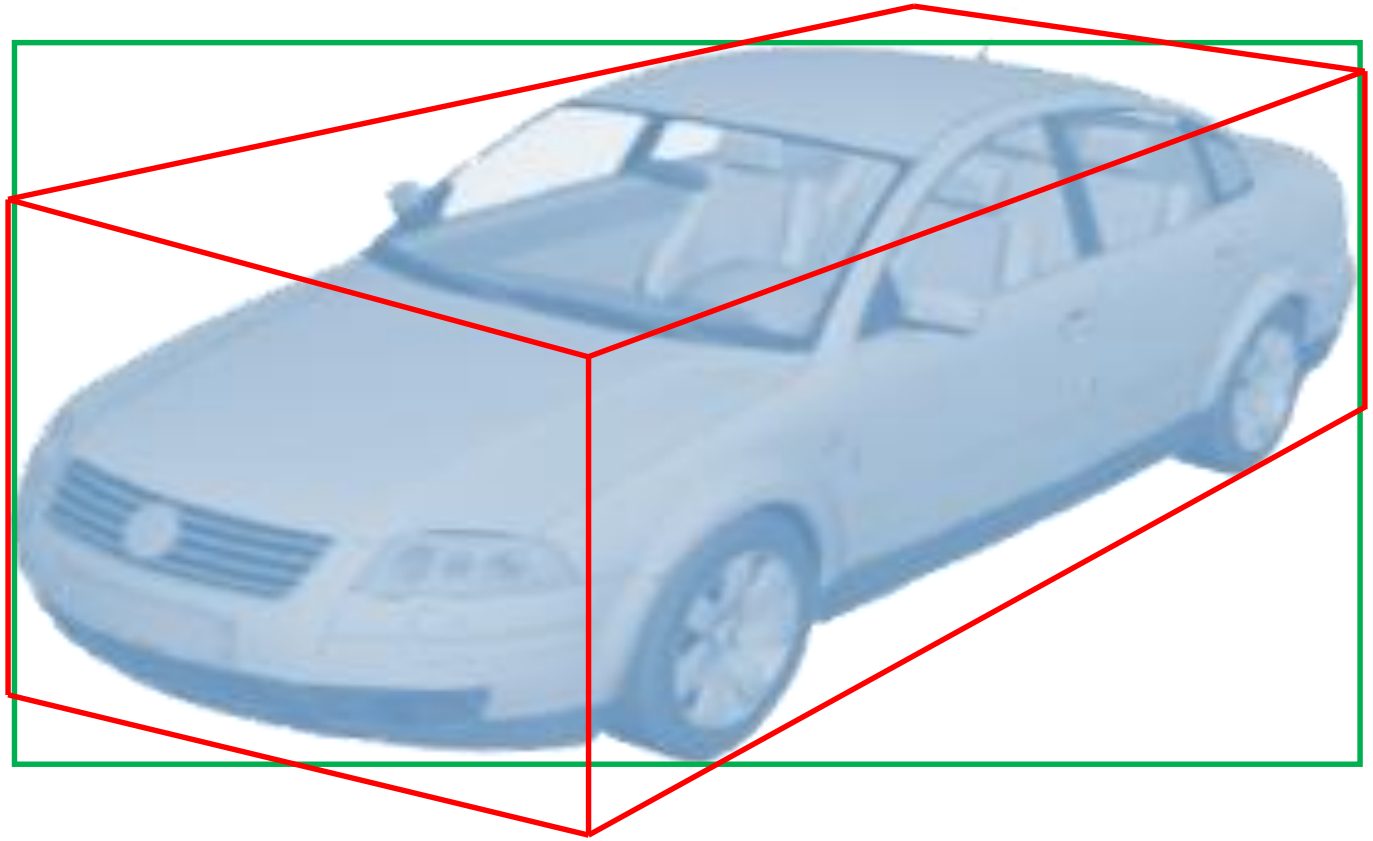
Probably yes!

Occlusions handled!

Annotate rendering only objects of interest

Render the whole scene

Object Bounding Box vs. Image Bounding Box



Rendered scene – photorealistic or not?

Very clean images: good or bad?

Is there such thing as “synthetic features”?

Motion blur: good enough?

Camera parameters: physically correct?

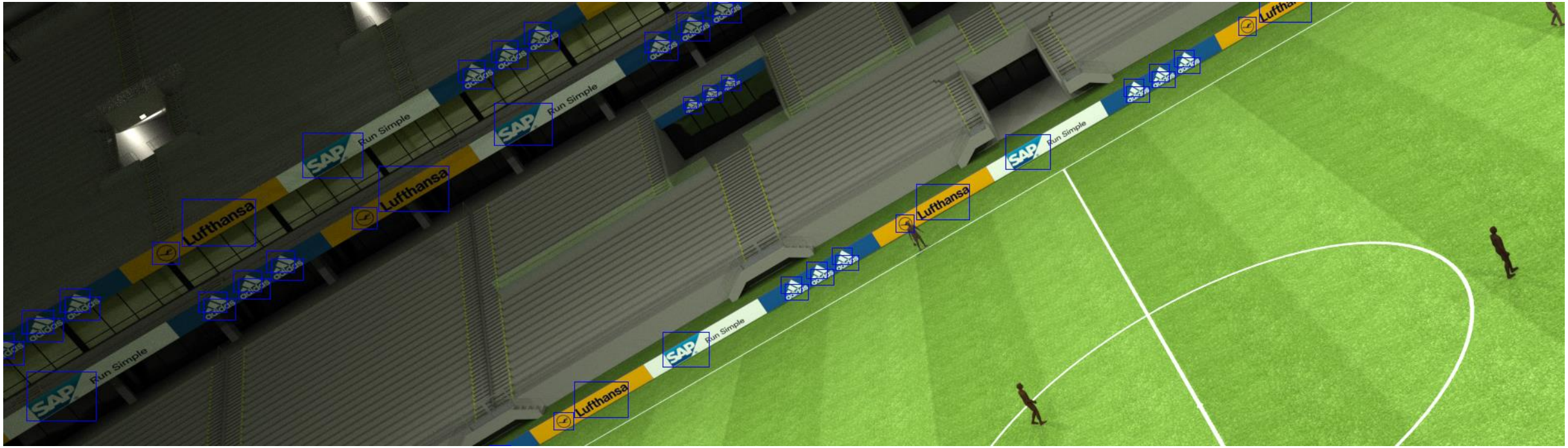
Camera artifacts, image compression artifacts (video): needed or not?

See “Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views” Hao Su et. al.

“Finetuning” the training set

What are the most “difficult” images?

What is a “good” training set?



GANs and what is real? Fans are wanted!

GANs: Generative Adversarial Networks, Ian Goodfellow et.al.

First proposed by Li, Gauci and Gross in 2013

Used in **unsupervised** machine learning

- Avoid annotation troubles

GAN: system of two networks competing against each other

- 1st CNN to generate image
- 2nd CNN to give a score (probability that it's "good/real")

Used for computer games visualizations

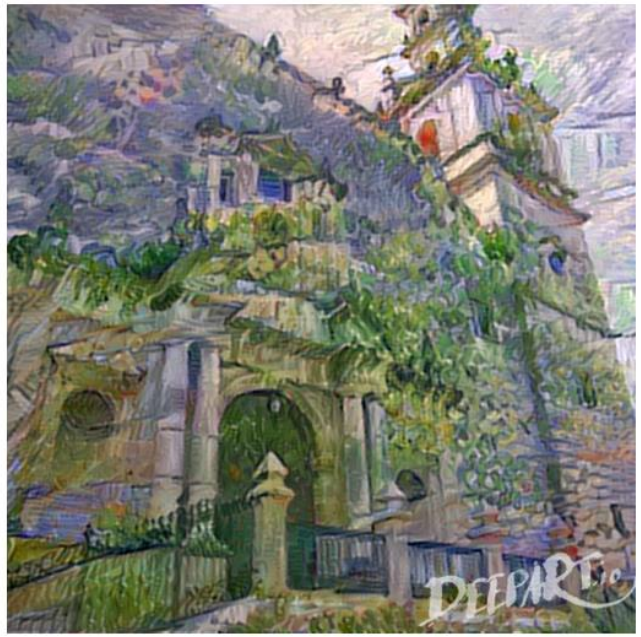
Make synthetic look "real"

GANs: style transfer

Motivation: “Perceptual Losses for Real-Time Style Transfer and Super-Resolution” Justin Johnson et.al.

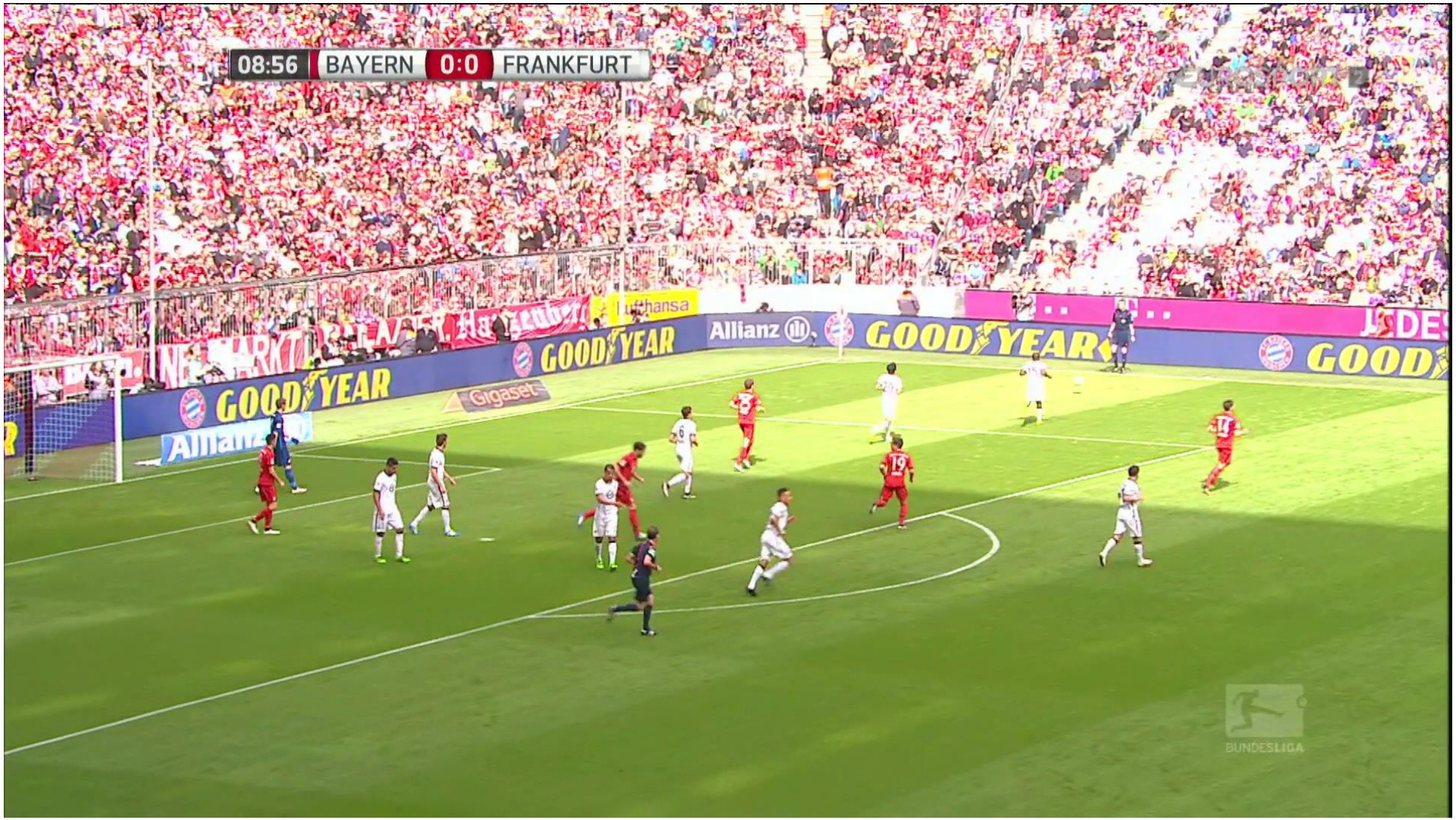
Deep Art

Prisma



GANs and what is real? Fans are wanted! ...#1

GANs and what is real? Fans are wanted! ...#2



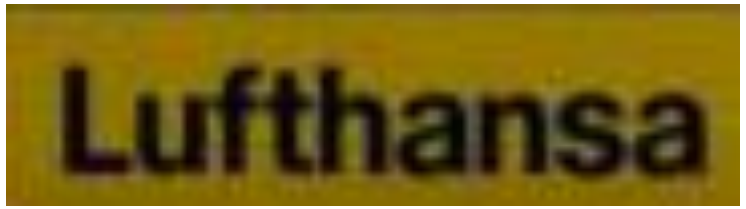
GANs and what is real? Fans are wanted! … #3

GANs: what happens…

Does brand logo stay recognizable by human?

Is the annotation (class, location and size) still valid?

What is changed?



More about object detection…

Hardware setup: NVidia Tesla P40, P100; Maxwell: M40, M60; GDX-1 (16GB memory on each GPU)

Training:

- Our optimized version of Caffe with cuDNN
- DGX-1 – fully utilized
- 10000 iterations x 8 images [1280x720] less than 24 hours
- Small batch processing
- Torque used for jobs dispatch
- 100 GBit InfiniBand for distributed file system and data transfers

Inference:

- Our fully asynchronous implementation intensively using TensorRT and CUVID
- On average **80 FPS 1280x720** images per one P100 GPU card
- 32-128 images per batch per card depending on input resolution

Conclusions

Quality and quantity of the training data is important for best result

We generate automatically annotated training sets with IRay SDK + server

Specialization is easily achieved

Plenty of scenario available

See our demo at SAP booth!

Acknowledgement

Berlin IRay team – IRay SDK and IViewer support, conversion from V-Ray model

SAP Innovation Center Israel team: model adjustment, infrastructure support and GAN experiments

Thank you!

contact:
Tatiana Surazhsky
SAP Innovation Center Israel
tatiana.surazhsky@sap.com