Deep Learning for Virtual Shopping

Dr. Jürgen Sturm
Group Leader RGB-D
metaio GmbH

metaio

Augmented Reality with the Metaio SDK: IKEA Catalogue App

Metaio: Augmented Reality

- Metaio SDK for iOS, Android and Windows
  - Marker-based 3D tracking
  - Image-/Template-based 3D tracking
  - SLAM (natural image features, local+global bundle adjustment, relocalization)
  - Edge-based tracking (given a CAD model)
  - Hybrid tracking (combining one or more of the above)
  - Face detection, tracking, augmentation using machine learning ← this talk!
- Company profile
  - World leading augmented reality provider, >100k active developers world-wide
  - 130+ employees, 60 developers, 30 computer vision & machine learning researchers
  - 4 locations: Munich (HQ), San Francisco, New York, Dallas
  - My role: Group leader RGB-D (face tracking and augmentation)

Motivation

- Market for online shopping is growing quickly
- Certain product categories are more difficult to sell online:
  - Furniture
  - Eyewear, earrings, makeup, hair colorant, cloth, shoes ← this talk
- Customer wants to see it / try it out him/herself before buying

→ **How can we support the customer in his buying decision?**

Furniture

Sunglasses

Earrings

Challenges

- Detect and track human faces in real-time
  - Large variations in appearance (viewpoint, illumination, skin color)
  - Face is articulated (many DOF)
  - Hair can have arbitrary shape
- Automatic placement of objects
- Achieve rendering realism (occlusions, illumination, dynamics)

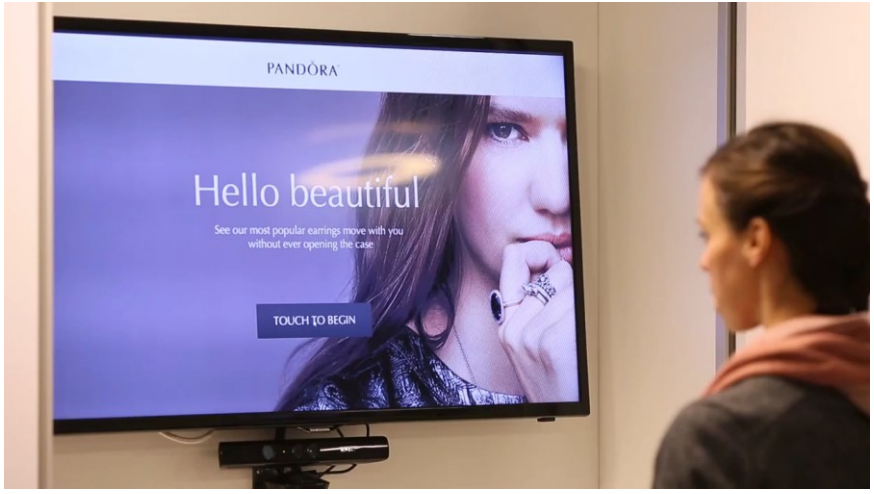
Furniture

Sunglasses

Earrings

Face Tracking and Reconstruction in 3D

- First result: Pandora Jewelery
  - RGB-D camera (Kinect)
  - Turn head slowly left and right to generate model
  - Place earrings manually on the head
  - Try out the live demo at our booth!

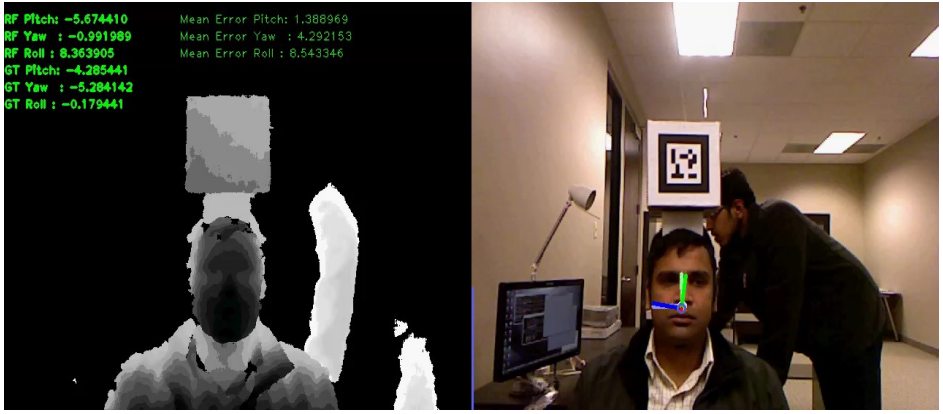


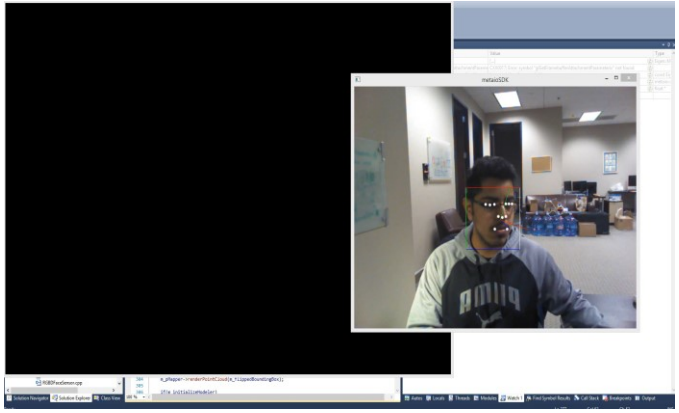
3D Face Reconstruction using Bump Maps

- Reconstruct head while user turns left and right
- Use 3D geometry as an occlusion model

3D Head Pose Estimation using Random Forests

- Train a random forest that predicts 3D head pose from depth images
  - Patch-wise classification head/no-head
  - Head pixels vote for center + head orientation
  - Estimate head center using mean shift and average inlier votes
- Recorded huge training dataset
- Allows for automatic (re-)initalization and recovery after tracking loss





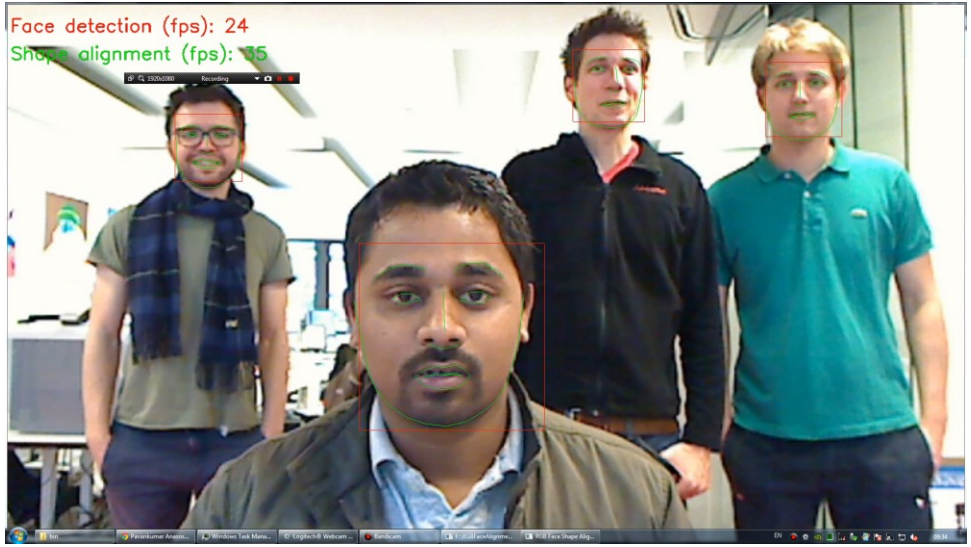
metaio

Physics Simulation

- Add physics / gravity to the models for increased realism
- Unity engine

How can we make this work on smartphones?

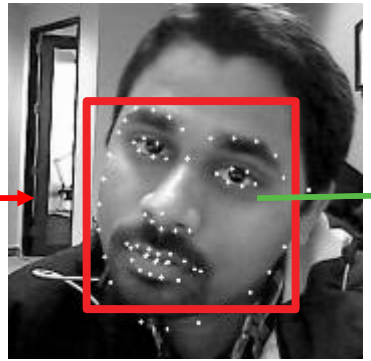
- Smartphones don‘t have depth cameras
- Is this possible with a monocular camera?
- Face alignment using random regression forests + ridge regression
- Localize 68 landmarks on human face



2D Face Alignment

- Given
  - RGB image
  - 2D face detector bounding box
- Wanted
  - Face shape [redacted]
- Learn a regressor that predicts shape [Ren et al., CVPR '14; Tzimiropoulos et al. ICCV '13, …]

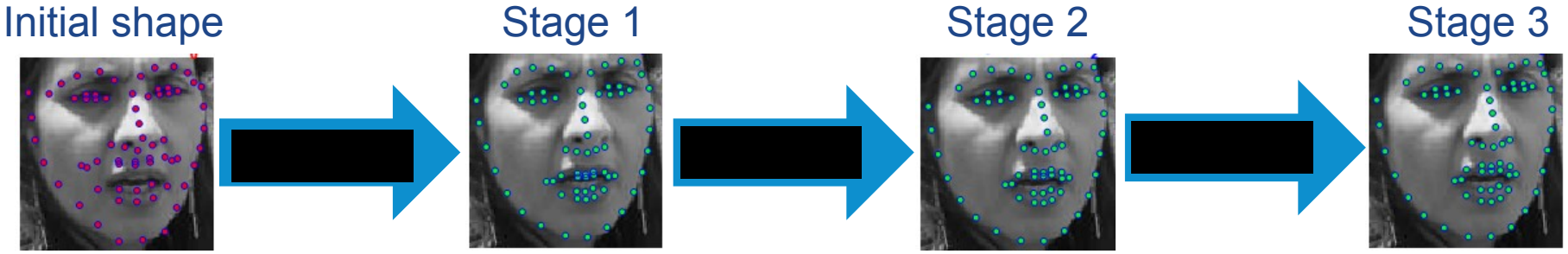
Input:
2D face bounding box



Output:
68 landmarks localized on face
(including eyes, mouth, outline)

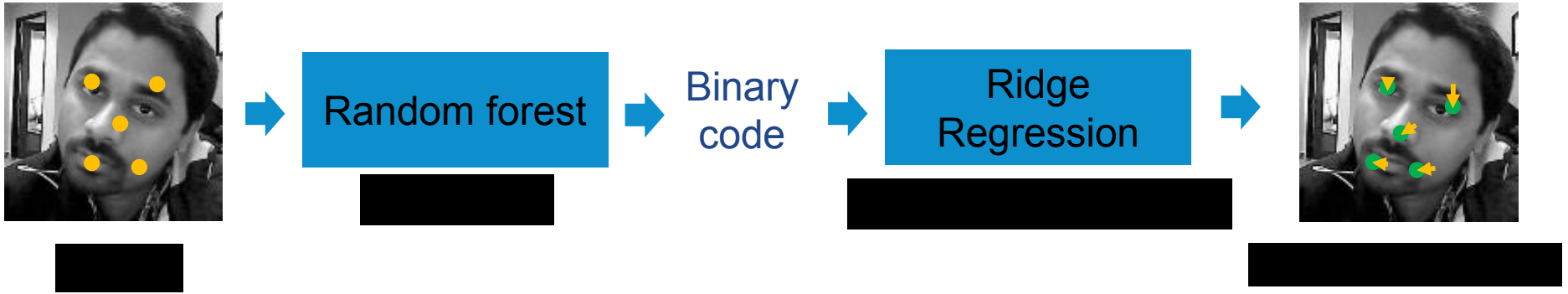
Approach

- Learn a regressor to predict shape increment
  - Input: Image I and initial shape [redacted]
  - Output: Shape increment [redacted]
- Apply a cascade of regressors
  - Contains T regressors [redacted]
  - At every stage, compute [redacted]



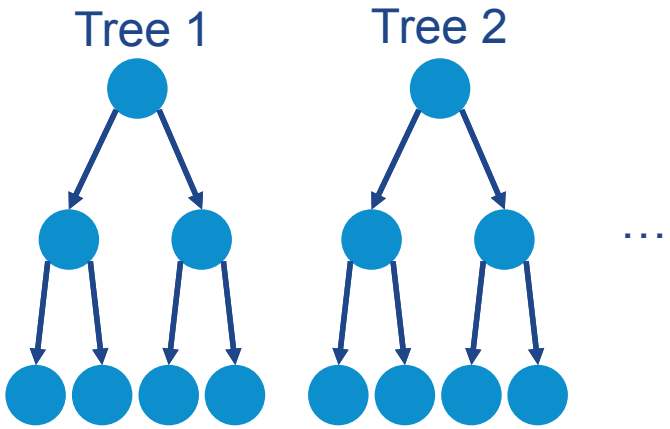
Regression Model

- Learn meaningful binary features using random forest
- Extract binary feature vector (relative to current shape estimate)
- Learn mapping from binary feature vectors to shape increment
- Apply linear shape update based on extracted feature vector



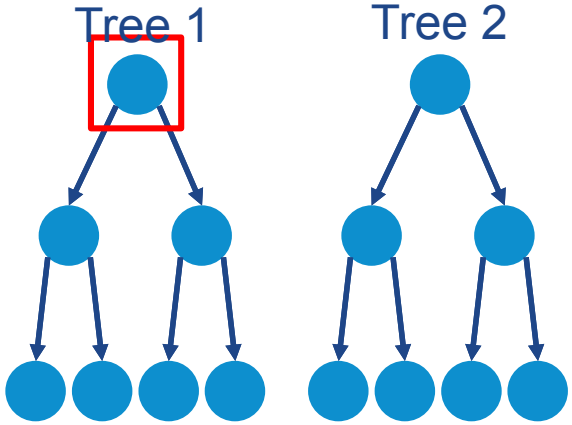
Random Forest

- Train one random forest per landmark
- Random forests consist of multiple random trees
- Feature: Compare pixel intensity relative to landmark positions
- Select features that maximize information gain
- Concatenate output to obtain a binary feature vector



Random Forest

- Train one random forest per landmark
- Random forests consist of multiple random trees
- Feature: Compare pixel intensity relative to landmark positions
- Select features that maximize information gain
- Concatenate output to obtain a binary feature vector

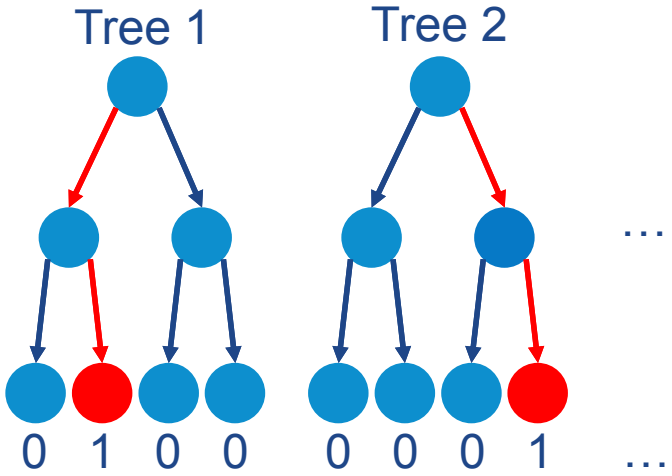


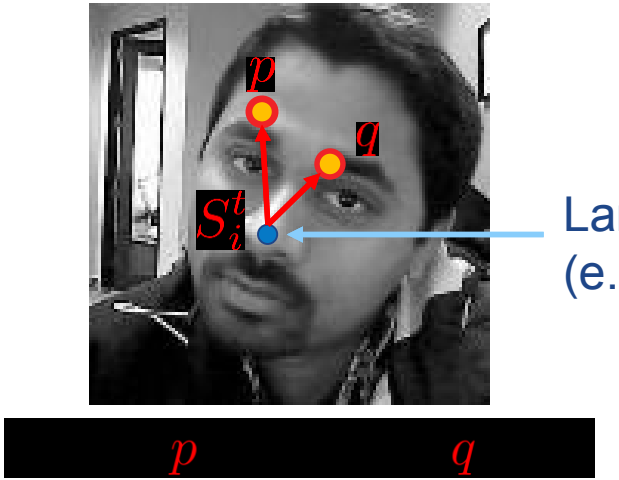
…

Landmark ■
(e.g., nose tip)

Random Forest

- Train one random forest per landmark
- Random forests consist of multiple random trees
- Feature: Compare pixel intensity relative to landmark positions
- Select features that maximize information gain
- Concatenate output to obtain a binary feature vector



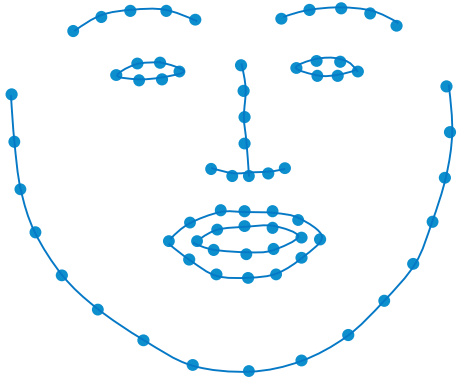


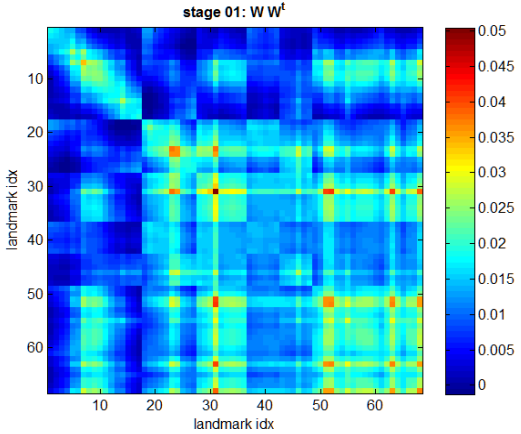
Ridge Regression

- Learn mapping from binary features to shape increments using ridge regression

data term regularizer

- Weighting matrix W models correlations between landmarks

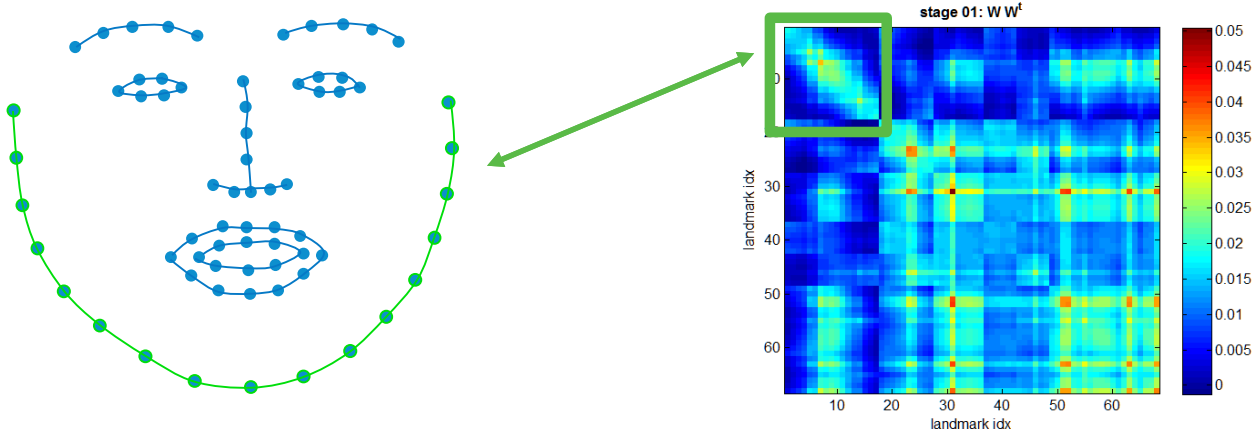




Ridge Regression

- Learn mapping from binary features to shape increments using ridge regression

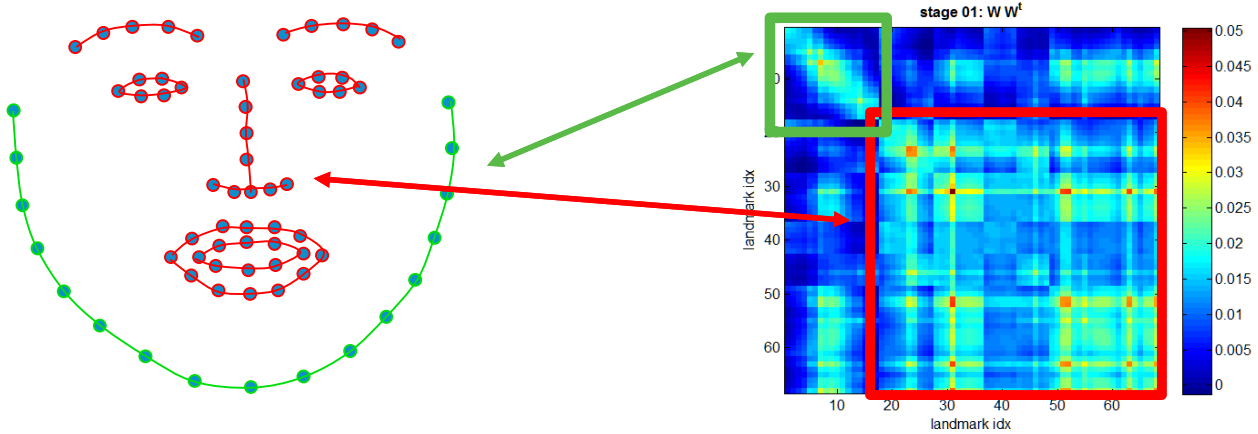
- Weighting matrix W models correlations between landmarks



Ridge Regression

- Learn mapping from binary features to shape increments using ridge regression

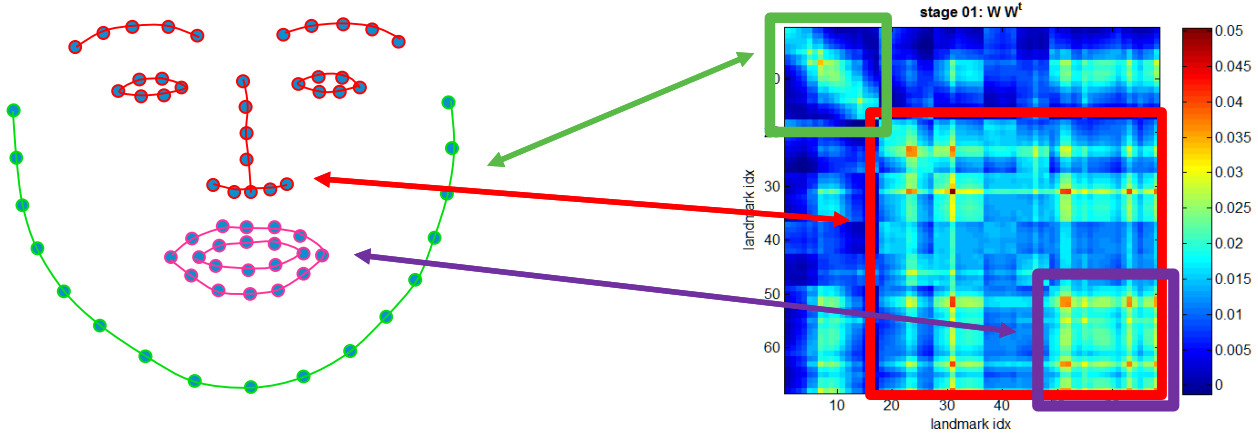
- Weighting matrix W models correlations between landmarks



Ridge Regression

- Learn mapping from binary features to shape increments using ridge regression

- Weighting matrix W models correlations between landmarks

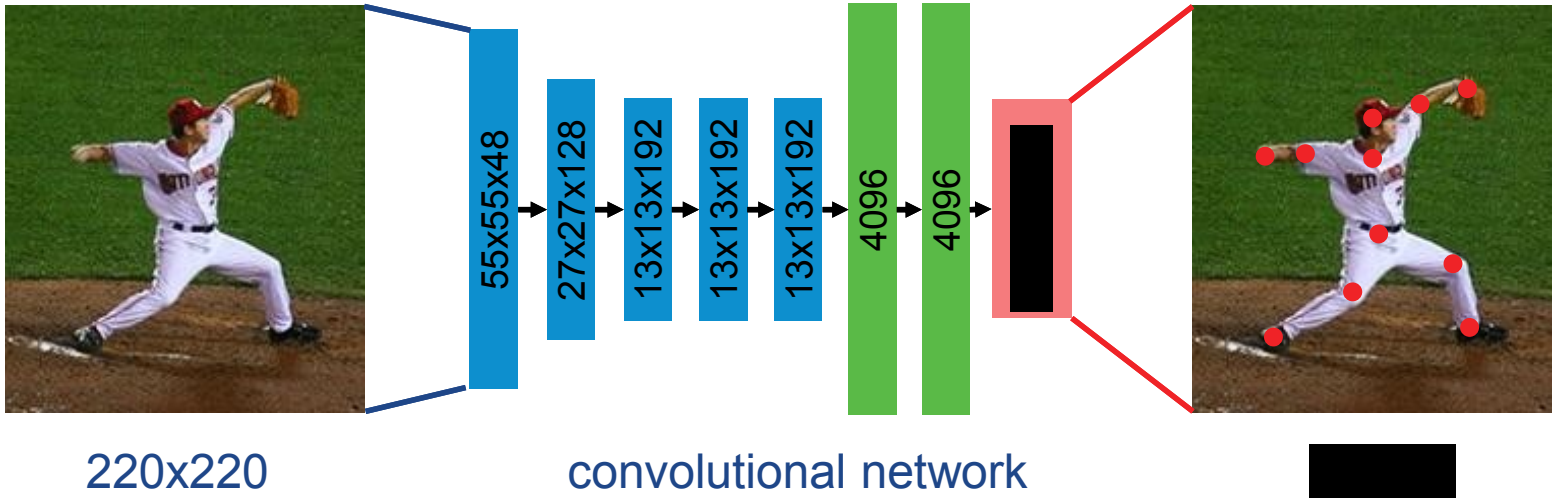


Face Alignment Summary

- Provides stable 2D landmark positions on face
- Can handle partial occlusions, landmarks are correctly correlated
- Enables automatic placement of objects on face
  - Earrings
  - Sunglasses
- Enables face augmentation without depth camera
- Very fast, current (unoptimized) implementation runs easily at 50fps on smartphones/tablets (500 fps on laptop)
- Try it yourself at our booth

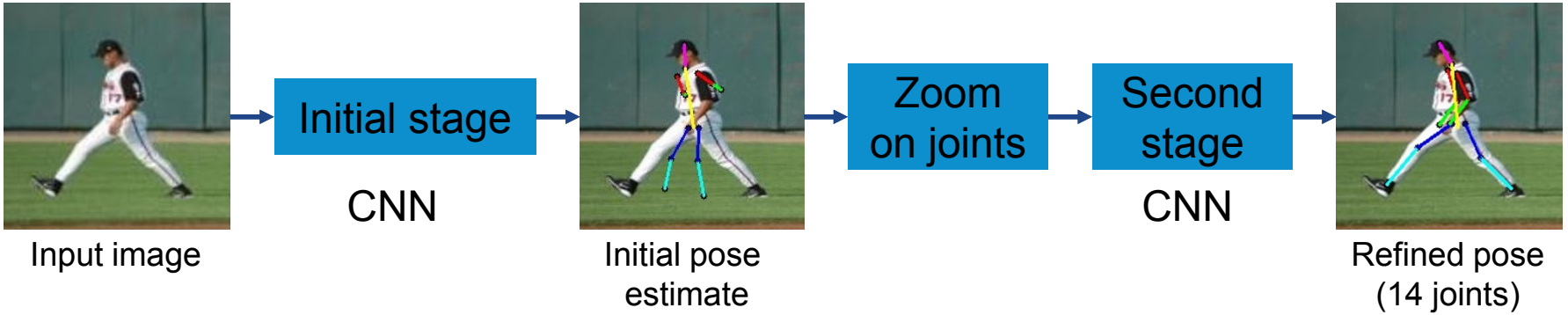
Pose Estimation using Convolution Neural Networks (CNNs)

- Estimate 2D positions of 14 full body joints
- For whole body augmentations (e.g., cloth)
- Can we use convolutional neural networks for this? [Toshev et al., CVPR '14; Jia et al., arXiv '14; …]



Cascade of Convolutional Neural Networks

- Initial stage takes downsampled image as input
- Second stage takes zoomed-in region around predicted joint positions



Pose Estimation using Convolution Neural Networks (CNNs)

- Implementation based on Caffe library from Berkeley
- Dataset:
  - LSP dataset (Leeds Sports Pose Dataset)
  - 11000 training, 1000 testing images
- Training:
  - Nvidia Quadro K4000 – 3GB Memory, 784 cores
  - Time: approximately 4 days
  - 180.000 iterations
- Promising results on test data
- Application to smartphones is difficult
  - Large memory requirements
  - High computational demands

Conclusion

- Take home message from this talk:
  Machine Learning is a key enabler for our technology!
- This Talk: Deep Learning for Virtual Shopping
  - Face detection, reconstruction for virtual-try on applications
  - Head pose estimation in 3D using random forests
  - Face alignment using random forests (localize 2D landmarks on the face)
  - Full body human pose estimation using CNNs
- We‘re looking for full time employees and summer interns (MSc and PhD)
  - Machine Learning
  - Computer Vision
- Visit us at our booth and see our live demos!

Metaio
Phone (EMEA): +49-89-5480-198-0
Phone (US): +1-415-814-3376
info@metaio.com
www.metaio.com

facebook.com/metaio

@twitt_AR
twitter.com/#!/twitt_AR

augmentedblog.wordpress.com

www.youtube.com/user/metaioAR

metaio