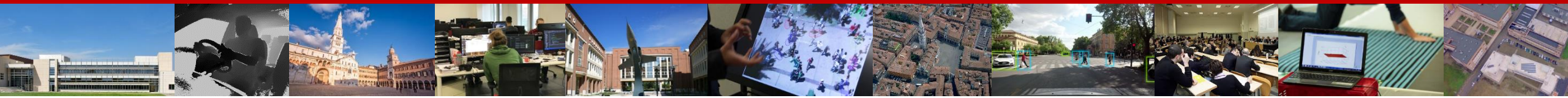
14th IEEE International Conference on
Automatic Face and Gesture Recognition
FG2019

Human Pose Understanding on Depth Maps



Guido Borghi

{name.surname}@unimore.it

University of Modena and Reggio Emilia, Italy

UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

AImage Lab

Centro Interdipartimentale di Ricerca
Softech: ICT per le Imprese

AImage Lab

Agenda

1. Depth Data
   - Depth Maps
   - Depth Devices
2. Depth-based datasets for Human Pose Estimation
   - *ITOP*
   - *Watch-n-Patch*
   - *Watch-R-Patch*
   - …
3. Depth-based methods for Human Pose Estimation
   - Generative Models
   - Discriminative Models
4. Head and Shoulder Pose Estimation on Depth Maps

Depth Data

Why Human Pose on Depth Maps?

- The **majority of the literature** regarding the *Human Pose Estimation* task is focused on **intensity images** (RGB and gray-level), as we have seen in the previous part of the tutorial

- Few works tackle the *Human Pose Estimation* on **different types of images**, like
  - Depth Maps (2.5D images)
  - Infrared Images
  - Thermal Images
  - …

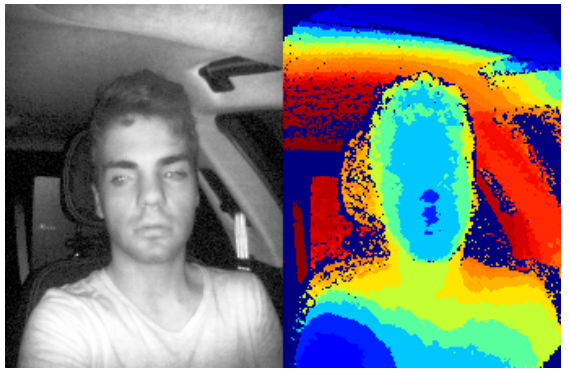
- **It would be interesting to apply Human Pose Estimation on Depth Maps**:
  - Exploit the *intrinsic* **3D information** contained in depth maps
  - **Light invariance**, since many depth sensors are based on infrared
  - The research topic about “depth-based algorithms” is acquiring importance

What is a Depth Map?

- A **depth map** is an image, or an image channel, that **contains information about the distance between two objects**, *e.g.* the acquisition device and a surface into the acquired scene, *i.e.* an object visible from the camera's point of view
- From a **2D perspective**, depth maps are usually coded as a **gray-level image**, *i.e.* a single channel-image with a 0 - 255 range. **However, each sensors has its own type.**
- From a **3D perspective**, depth map is **a projection of a point cloud**, in which every point contains the 3D position in respect to the camera coordinate system
- In the literature, depth maps are also referred as *depth images*, *range images* and *2.5D images*

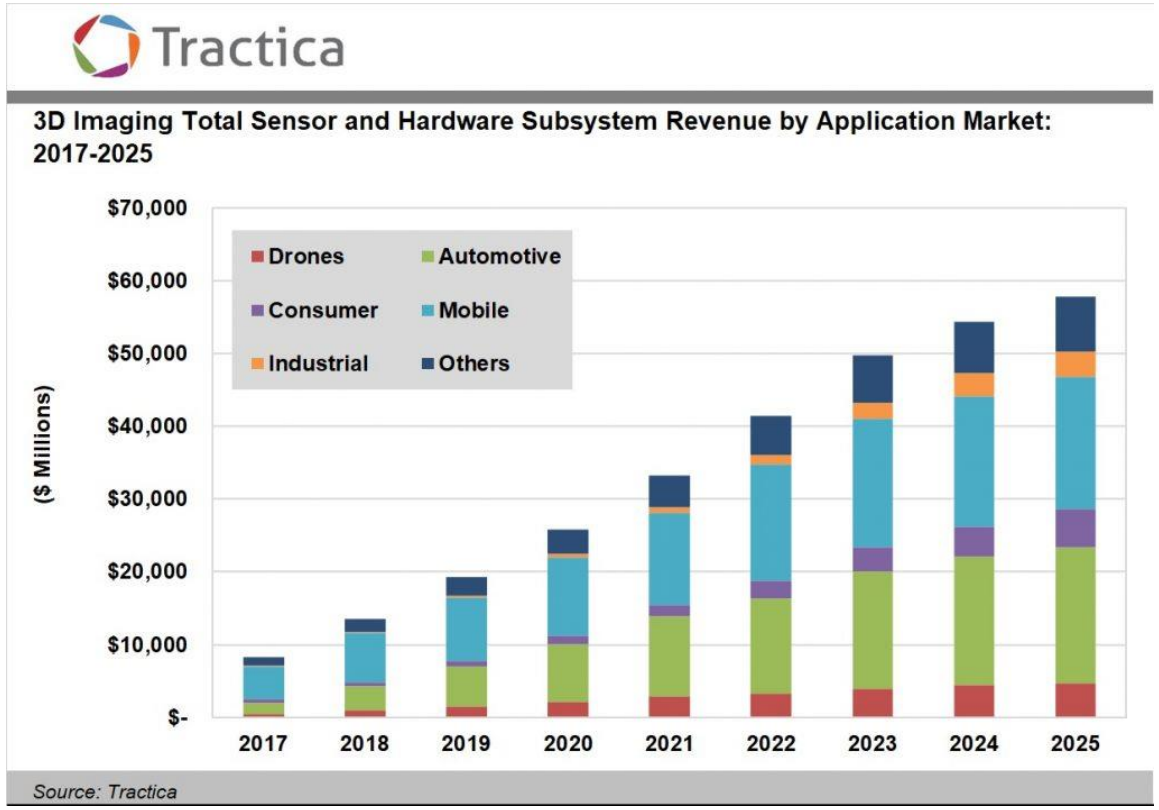
IF Depth Map

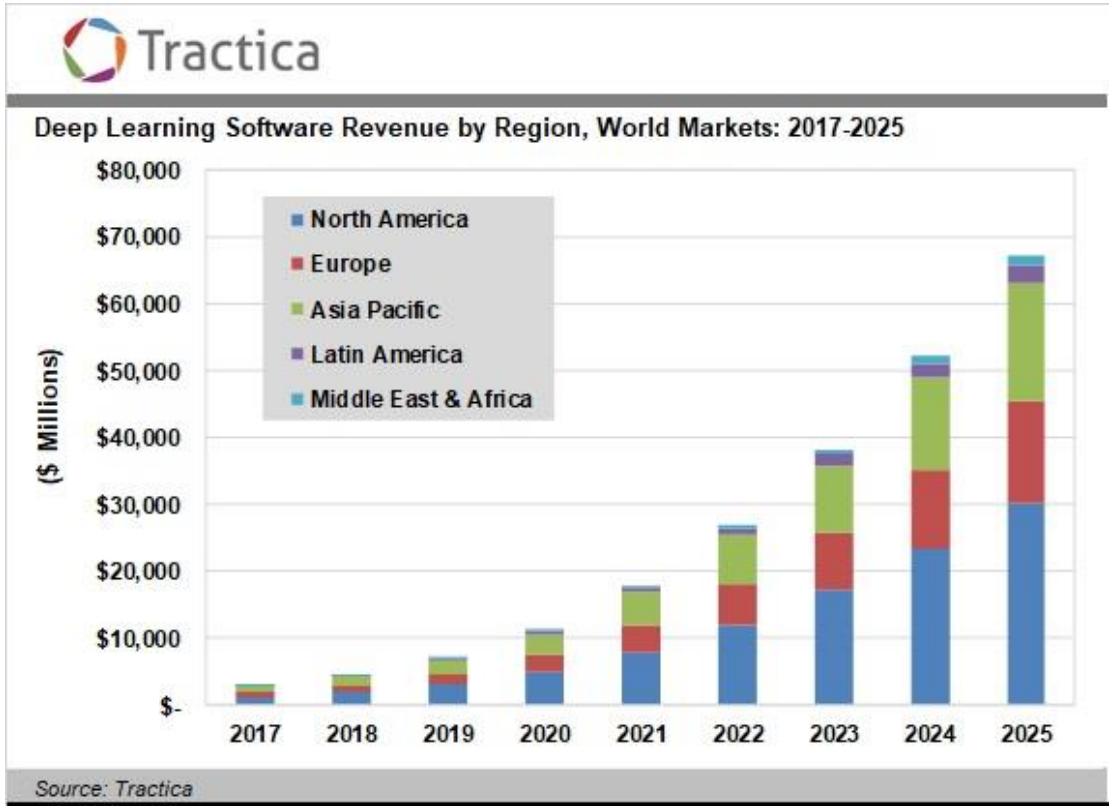
IF Depth Map



Depth Devices

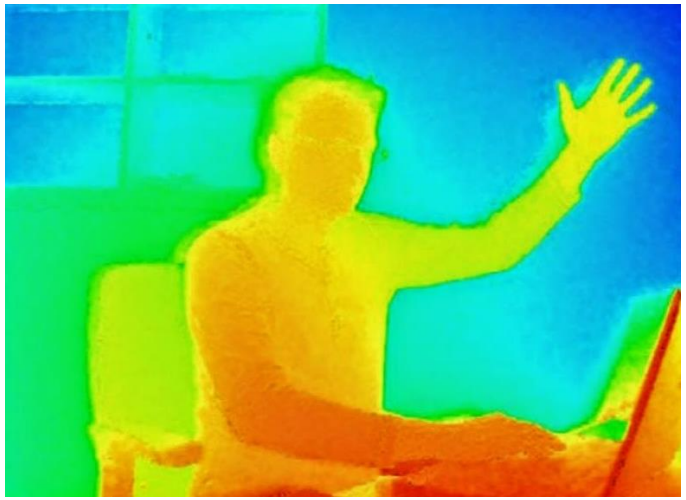
- **Depth Sensors**: devices that are able to provide in output distances
- Recently, a lot of new depth sensors have been introduced in the market
- A new trend is acquiring increasing importance: **Deep Learning** + **Depth Maps**





Depth Devices

- There are **three main technologies**:
  - **Stereo Cameras**
  - **Structured Light** (SL)
  - **Time-of-Flight** (ToF)

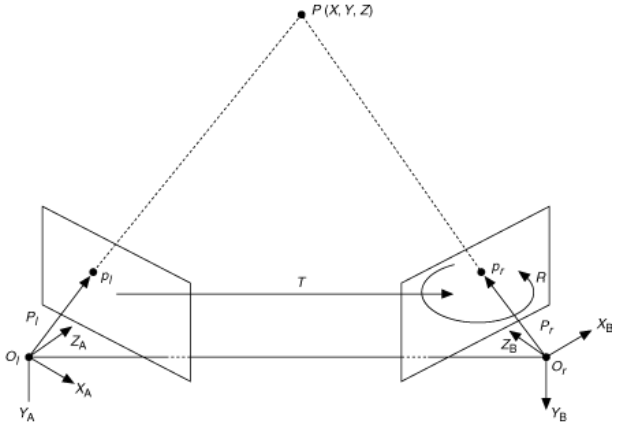


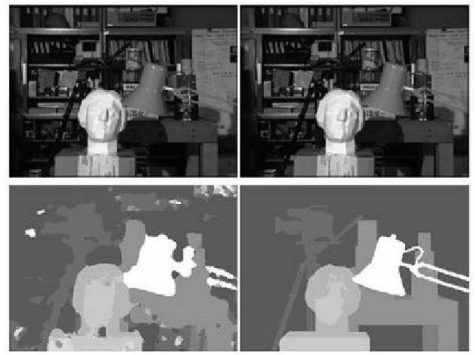
Each type has its **pros** and **cons**.

We briefly analyze these types of sensors in order to better understand the following part of the tutorial.

Depth Devices

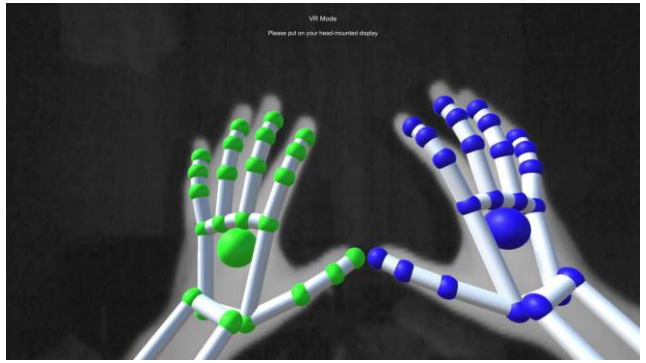
- **Stereo Cameras**:
  - The base concept is similar to what happens in the **human body with the eyes**
  - Two similar cameras are placed in a **fixed distance** on the same plane
  - The **disparity map** of the scene is computed by combining acquired images of these two different intensity (gray-level or RGB) cameras, resolving the so called **correspondence problem**
  - Given a pair of rectified images, it is possible to retrieve the distance of a point in the scene applying a **triangulation method** on a set of corresponding points that lying on *epipolar* lines.





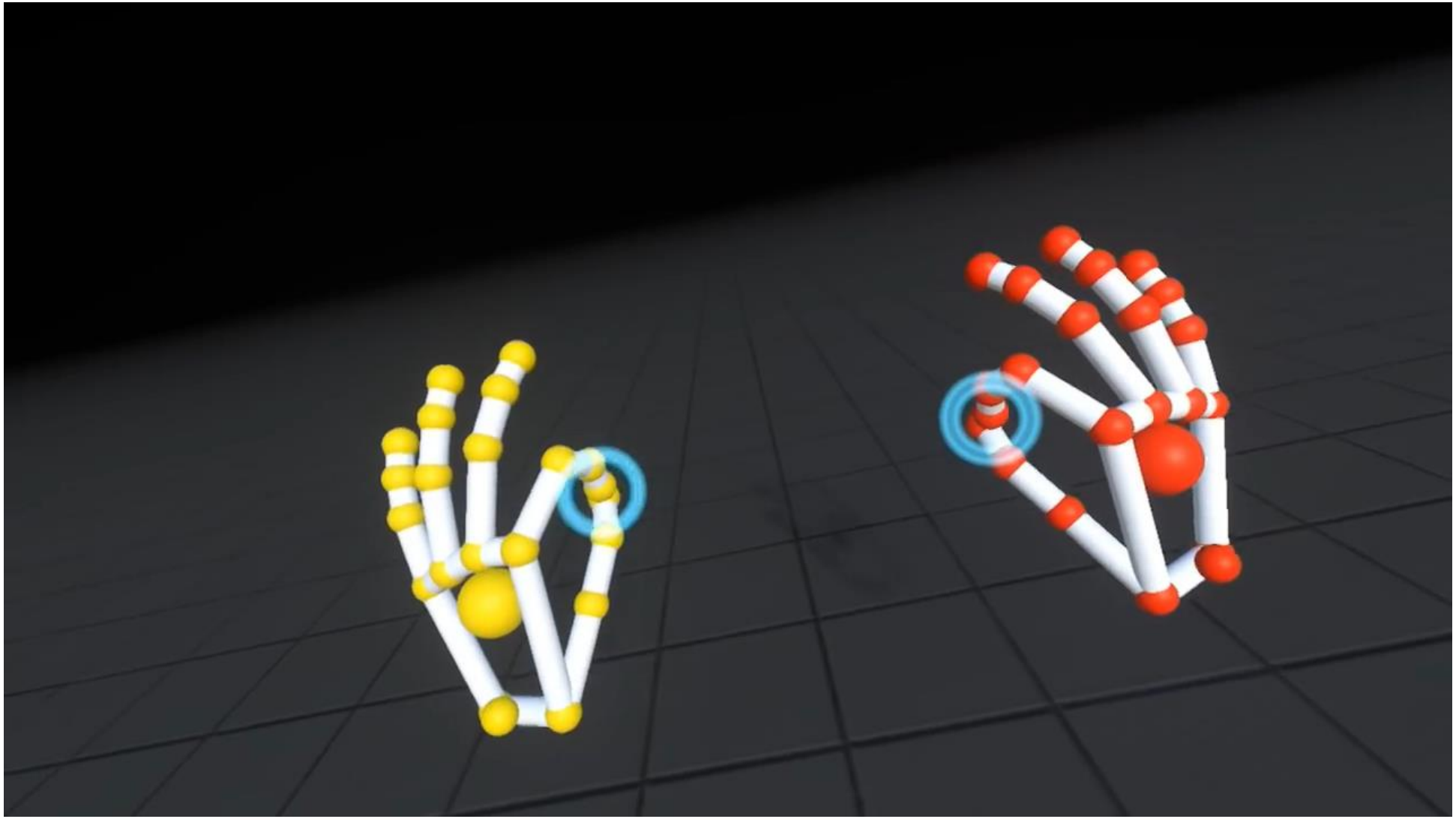
Depth Devices

- Example of Stereo Camera: the *Leap Motion* device
  - **2 infrared cameras** with a spatial resolution of **640x240**
  - **Up to 200 frame per second**
  - Field of view: 135° (**fish-eye lens**)
  - **Small Factor Size**: 7 x 1.2 x 3 mm
  - Only **32g** of weight
  - *SDK* for real time hand tracking (robust and accurate)



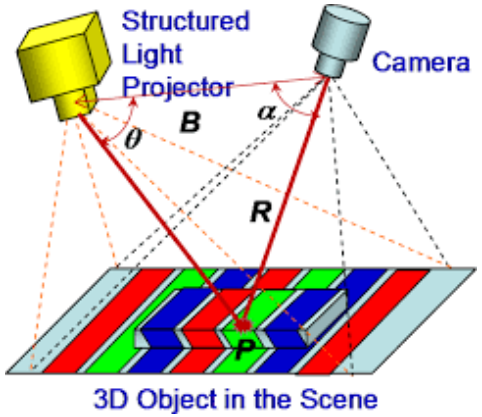
1. https://www.leapmotion.com

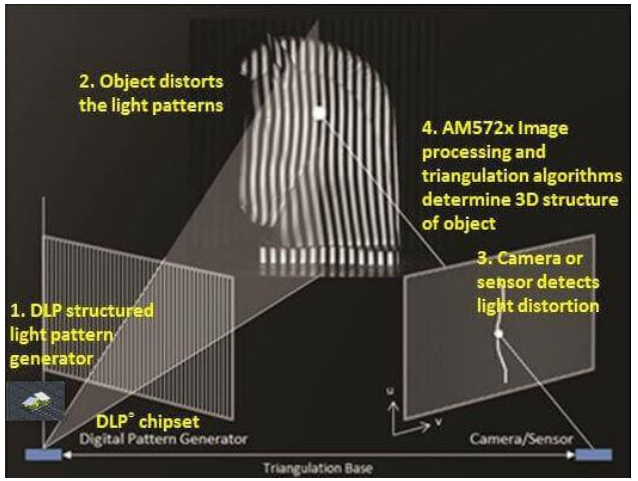
Depth Devices



Depth Devices

- **Structured Light** devices:
  - These scanners project a **specific pattern** inside of the scene
  - The **deformation** in the projected pattern introduced by the objects present inside of the scene is analyzed, through appropriate geometric transformations, **to return for every projected point its 3D position**
  - The hardware of these devices includes a **laser projector** and a **sensor** that is sensitive to the corresponding bandwidth

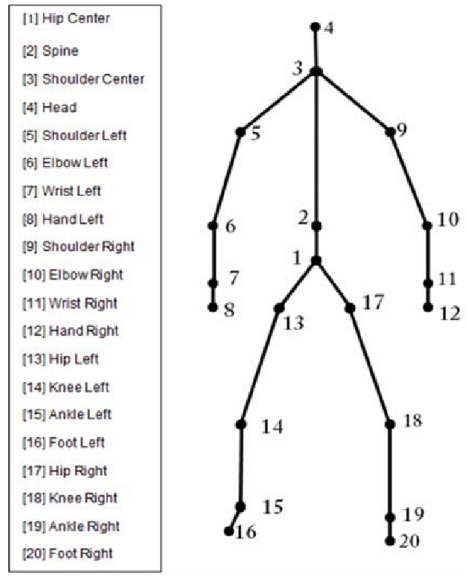


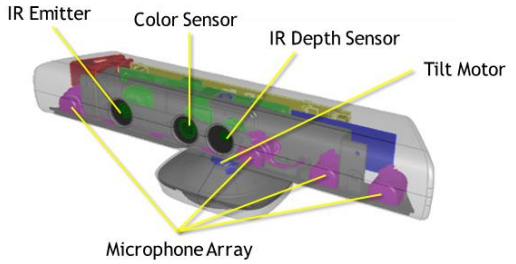


Depth Devices

Example of Structured Light device: the *Microsoft Kinect* (first version)

- **RGB** camera: 640x480 up to 30fps
- CMOS **depth** sensor (320x240)
- **Range**: 0.4 – 4.5/6 m
- **2 full skeleton tracked** (**20 joints**)
- Power consumption: 2.5W
- Field of view: 57° x 43°
- **Tilt motor**

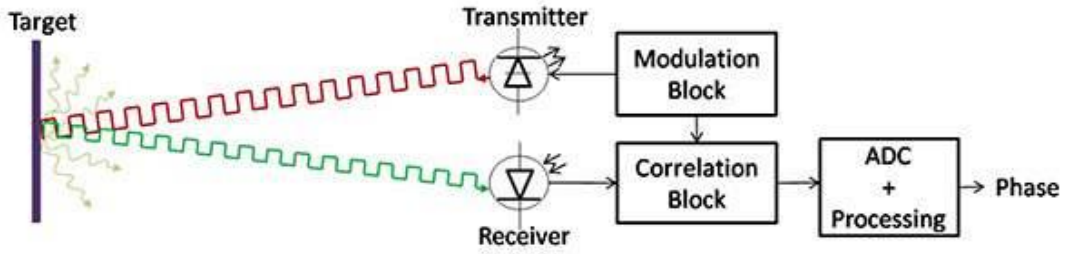




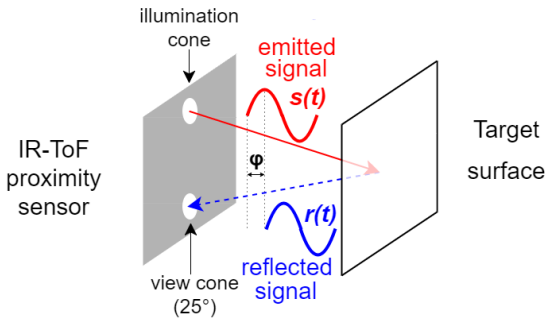
Depth Devices

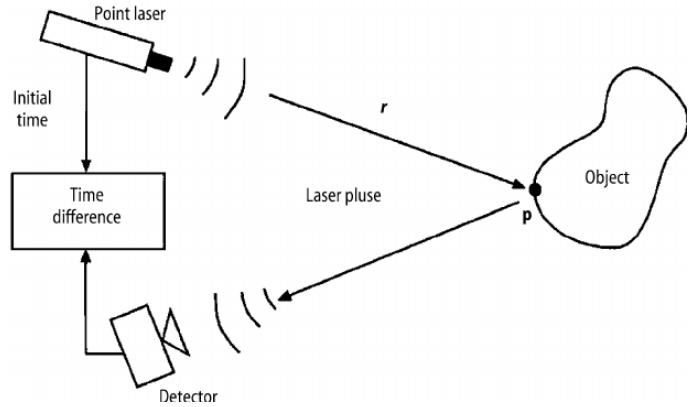
Time-of-Flight devices:

- The distance is computed measuring the **time interval** (the **phase difference**) taken for infrared light to be reflected by the object in the scene



- Also in this case, it is necessary to have a **laser projector** and a **sensor** sensitive to the corresponding bandwidth

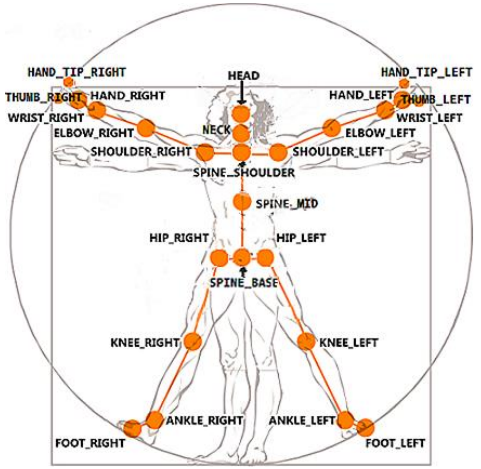




Example of ToF device

Example of ToF device: the *Microsoft Kinect One* (second version):

- **RGB** camera: 1920x1080 up to 30fps
- CMOS **depth** sensor (512x424)
- Range: 0.5 – 5 m
- **6 full skeleton tracked** (**26 joints**)
- Power consumption: 2.5W
- Field of view: 70° x 60°
- **No tilt motor**



Example of ToF

Examples of ToF devices:

- *CamBoard Pico Flexx*
  - **Depth** sensors with a spatial resolution of 224x171
  - Only 68 x 17 x 7.35 mm (**8g**)
  - Up to **45 fps**
  - Range: 0.1 – 4m

- *Pico Zense DCAM710*
  - 69mm x 25mm x 21.5mm
  - **Depth resolution: 640 * 480 @ 30FPS**
  - RGB resolution: 1920 * 1080 @ 30FPS
  - Viewing angle: 69 ° (horizontal) 51 ° (vertical)

1. https://pmdtec.com/picofamily/flexx/
2. https://www.pico-interactive.com/zense

Depth Sensor Comparison

ToF vs Stereo Cameras:

- **Stereo cameras are cheaper and simpler** (just two cameras in a fixed position)
- **In stereo cameras, the computational load of the correspondence problem is significative**: this requires computational intensive algorithms for feature extraction and matching
- **Stereo cameras rely on standard intensity images**, and so are prone to failure in case of low quality of images or an insufficient variation level of textures and colors into the scene
- ToF devices are not affected by these limitations, since they do not depend on color or textures and there is not the correspondence problem
- **With Stereo Cameras the reconstruction error is a quadratic function of the distance**, while ToF sensors, that are based on reflected light, are usually able to increase the illumination energy when necessary

1. *Sarbolandi, H., Lefloch, D., & Kolb, A. "Kinect range sensing: Structured-light versus Time-of-Flight Kinect, CVIU 2015*

Depth Sensor Comparison

ToF vs Structured Light:

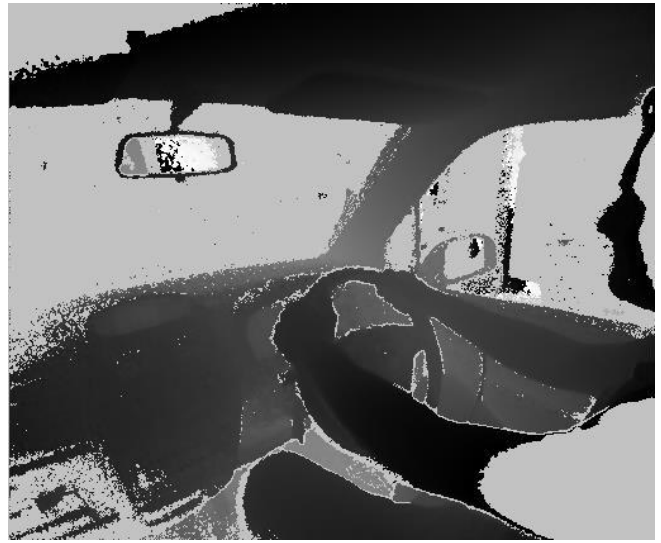
- Both ToF and SL have a high spatial resolution in conjunction with “low” hardware costs
- SL are influenced by external sources of light (and near-infrared light, *e.g.* the sunlight)
- The parallel use of several structured-light or ToF sensors is limited by **interference problems**
- **In SL devices, lower frame rate usually corresponds to undesired blur effects** if the subject does not remain relatively still during the projection sequence
- The majority of the aforementioned problems are only partially present in ToF devices, that are **less influenced by external light sources**, and they can achieve a **higher frame rate** just increasing the size of the laser projector and the related sensor

Depth Maps are useful for…

- Since depth devices are mainly based on the infrared light, depth maps are useful in systems that require:
  - **Light Invariance**: vision-based systems have to be reliable even in presence of light changes
  - For example, this is the case of the **automotive** context, in which the light invariance is needed in case of night, tunnels and bad weather conditions.

The automotive context has some other requirements satisfied by (new) depth devices:

- **Non-invasiveness**: driver's movements and gaze must not be impeded during the driving activity
- **Real Time performance**: monitoring and interaction systems have to quickly detect anomalies and provide a fast feedback



Dept-based Datasets for Human Pose Estimation

Data are important!

- **In deep learning-based methods data are extremely important**
- First works about DL in late ‘70, but they were limited by mathematical and computational issues
- Today, it is a **revolution** in the Computer Vision (but not only) field
- Powerful models, but there are some **limitations** related to:
  - Availability of a **huge amount of training data**
  - **Data must be annotated** (for *supervised* approaches)

  → «*Data is the new oil*» (Clive Humby)
  - **High computational power** needed

  → Nvidia GPU

Data collection and annotation is a key element in developing new AI algorithms

Depth-based dataset Issues

- In general, there is **a lack of depth-based datasets** for computer vision tasks
  - Until a few years ago, depth devices were **expensive** and **difficult** to use
  - Often, datasets are **limited in size** (they do not allow a deep learning approach)
  - **Each sensor codifies the output maps in its own convention**
  - Now, there are a lot of cheap and accurate depth devices
- Only few depth-based datasets have been collected for the *Human Pose Estimation* task
- Often, **body joints are automatically annotated** (usually exploiting the work of Shotton *et al.*)
  - It is a **real time** method
  - It is very simple to use and it is present in the commercial *SDK* of the *Microsoft Kinect*
  - Quite accurate (about 65% of accuracy)
  - Limited to **standing** subjects **facing** the acquisition device (gaming context)
  - For example, is not suitable for the automotive context

Depth-based datasets

- **Invariant-Top View Dataset** (ITOP, ECCV 2016)
  - **100k real-world depth images** from multiple camera viewpoints
    - **Front/side view**
    - **Top view** (camera on the ceiling pointed down to the floor)
  - Two *Asus Xtion PRO* cameras used
  - 20 people
  - 15 actions sequences
  - **Partial manual annotations**:
    - 1° step: Shotton *et al.* on the frontal camera (and then geometricaly transferred)
    - 2° step: iterative ground truth error correction based on *k-nearest neighbors* and center of the mass convergence
    - 3° step: humans manually evaluate, correct and discard noisy frames

1. *Haque, A., Peng, B., Luo, Z., Alahi, A., Yeung, S., & Fei-Fei, L. "Towards viewpoint invariant 3d human pose estimation", ECCV 2016*

Depth-based datasets

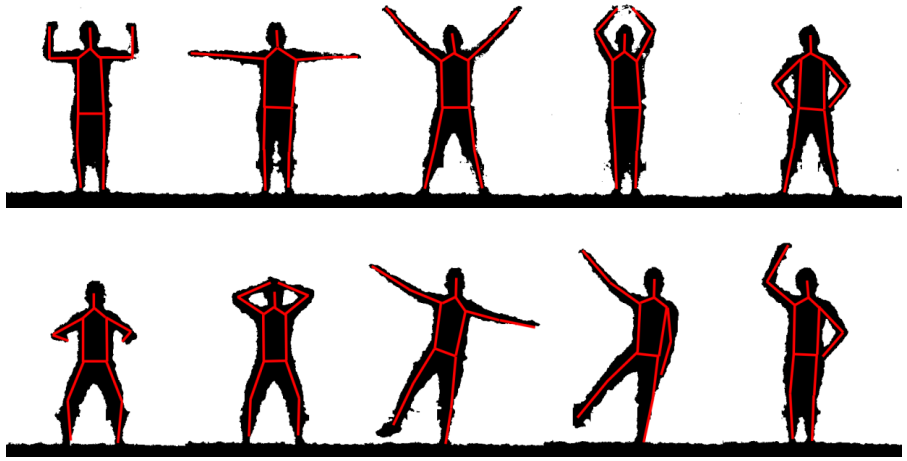
Labels

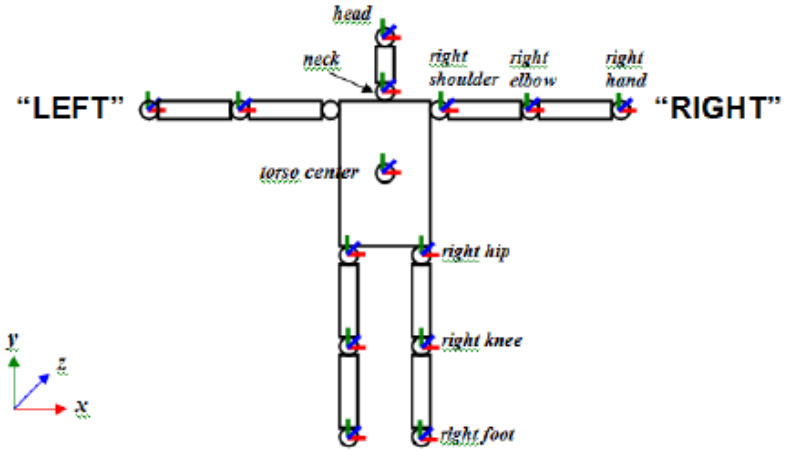
Key	Dimensions	Data Type	Description
id	$(n,)$	uint8	Frame identifier in the form XX_YYYYY where XX is the person's ID number and YYYYY is the frame number.
is_valid	$(n,)$	uint8	Flag corresponding to the result of the human labeling effort. This is a boolean value (represented by an integer) where a one (1) denotes clean, human-approved data. A zero (0) denotes noisy human body part labels. If is_valid is equal to zero, you should not use any of the provided human joint locations for the particular frame.
visible_joints	$(n, 15)$	int16	Binary mask indicating if each human joint is visible or occluded. This is denoted by α in the paper. If $\alpha_j = 1$ then the j^{th} joint is visible (i.e. not occluded). Otherwise, if $\alpha = 0$ then the j^{th} joint is occluded.
image_coordinates	$(n, 15, 2)$	int16	Two-dimensional (x, y) points corresponding to the location of each joint in the depth image or depth map.
real_world_coordinates	$(n, 15, 3)$	float16	Three-dimensional (x, y, z) points corresponding to the location of each joint in real world meters (m).
segmentation	$(n, 240, 320)$	int8	Pixel-wise assignment of body part labels. The background class (i.e. no body part) is denoted by -1.

Depth-based datasets

Depth-based datasets

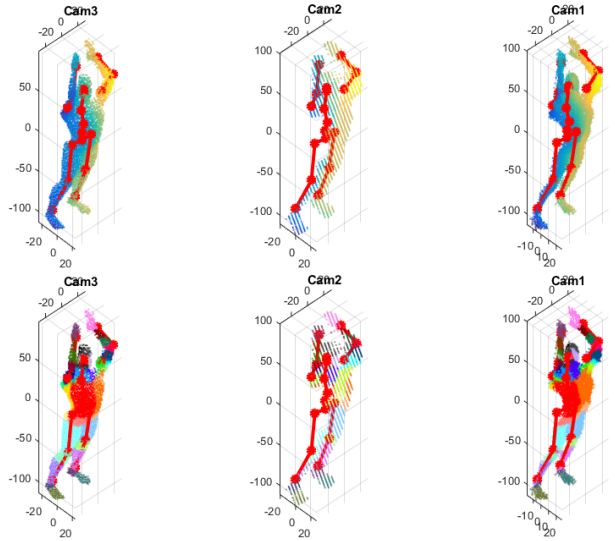
- **Body Pose dataset** (*Universitat Politècnica de Catalunya*, Barcelona)
  - 12 subjects
  - 12 standstill body poses
  - An **articulated body model** is given per frame providing **a set of body articulation positions**
    - For each frame, a body skeleton tracker is applied, taken from the *OpenNI/NITE* library
  - 2 types of sequences (48 recording in total):
    - Basic
    - Advanced





Depth-based datasets

- **UBC3V dataset** (CRV 2016)
  - **Synthetic** dataset
  - The dataset distinguishes the **back-front and left-right** sides of the body
  - The dataset has **three randomly located cameras for each pose**, which makes it suitable for multiview pose estimation settings
  - **3 sub-sets**:
    - **Easy-pose**: limited set of postures
    - **Inter-pose**: all the postures, but with only one subject
    - **Hard-pose**: all the postures and all the 16 subjects
  - *Matlab* tool to handle the dataset

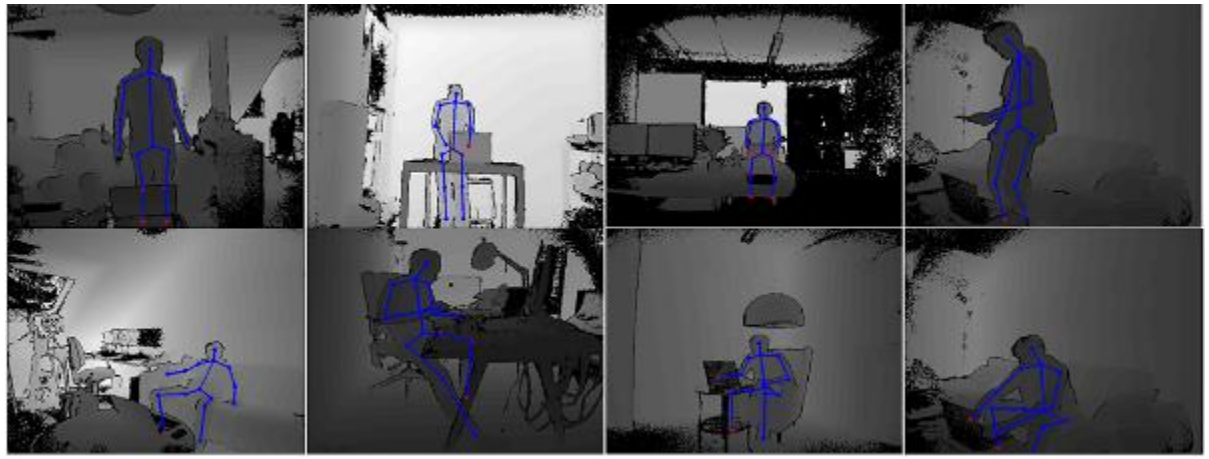


AImage Lab

Depth-based datasets

Depth-based datasets

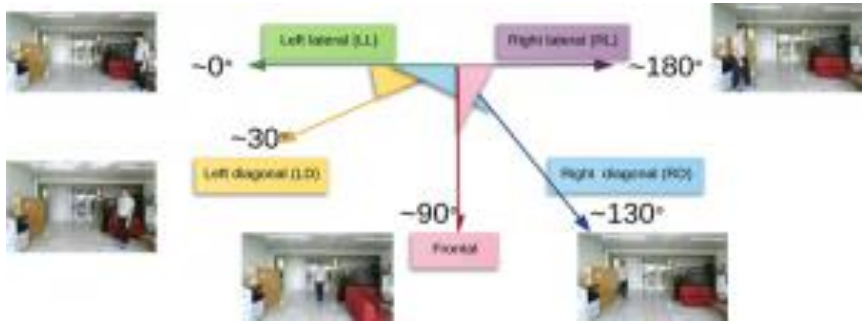
- **Pose Occlusion dataset** (CVPR Workshop 2015)
  - 1000 (800 train + 200 test) depth images
  - *Microsoft Kinect One* used
  - Each image in the dataset consists of a single person **partially occluded** by objects like table, laptop and monitor
  - Each image is annotated with **2D ground-truth** positions of **15 body joints** and their **visibility**



Depth-based datasets

- **Multi-View Kinect skeleton dataset*** (FG 2017)
  - RGB, depth, Infrared acquired by the *Microsoft Kinect One* at 10 fps
  - 22k frames in total
  - 20 walking subjects, 5 different directions
  - Long-term person re-identification using biometrics
  - 2D and 3D body joints (18 keypoints exploiting Shotton *et al.* method)

- **Captury dataset** (ICRA 2018)
  - RGB, depth, Infrared acquired by the *Microsoft Kinect One*
  - 3 subjects
  - Simple actions
  - Only 1.5k frames both in train and test



1. Zimmermann, C., Welschehold, T., Dornhege, C., Burgard, W., & Brox, T. "3d human pose estimation in rgbd images for robotic task learning", ICRA 2018

*originally not created for the Human Pose Estimation task

Depth-based datasets

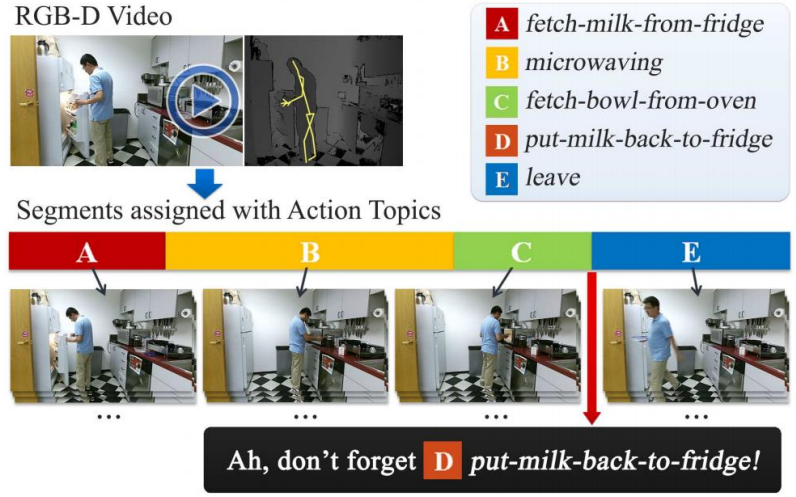
- **CAD-60 dataset***
  - 60 RGB-D videos
  - 4 subjects: two male, two female, one left-handed
  - 5 different environments: office, kitchen, bedroom, bathroom, and living room
  - 12 different activities with **tracked skeletons** (15 joints, based on Shotton *et al.*)

- **CAD-120 dataset***
  - 120 RGB-D videos of long daily activities
  - 4 subjects: *two male, two female, one left-handed*
  - 10 high-level activities + 10 sub-activity labels
  - **Tracked skeletons** (15 joints, based on Shotton *et al.*)

**originally not created for the Human Pose Estimation task*

Depth-based datasets

- **Watch-n-Patch (CVPR 2015)***
- RGB-D dataset acquired with the *Microsoft Kinect One* (second version)
  - **RGB**: Full HD resolution (1920x1080)
  - **Depth**: 512x424
  - Recordings of 7 people
  - 21 different kinds of actions
    - Each recording contains a single subject performing multiple actions in one room chosen between 8 offices and 5 kitchens.
  - 458 videos, corresponding to 230 minutes and 78k frames
  - **Annotations:**
    - **Actions**
    - **Body Joints** (Shotton *et al.*)

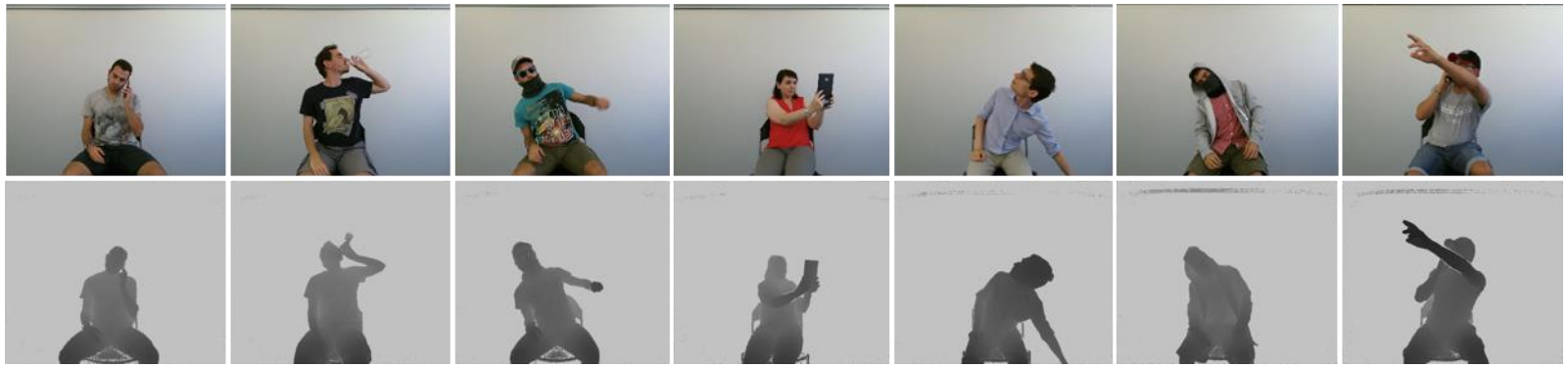


1. Wu, C., Zhang, J., Savarese, S., & Saxena, A. "Watch-n-patch: Unsupervised understanding of actions and relations", CVPR 2015

*originally not created for the Human Pose Estimation task

Depth-based datasets

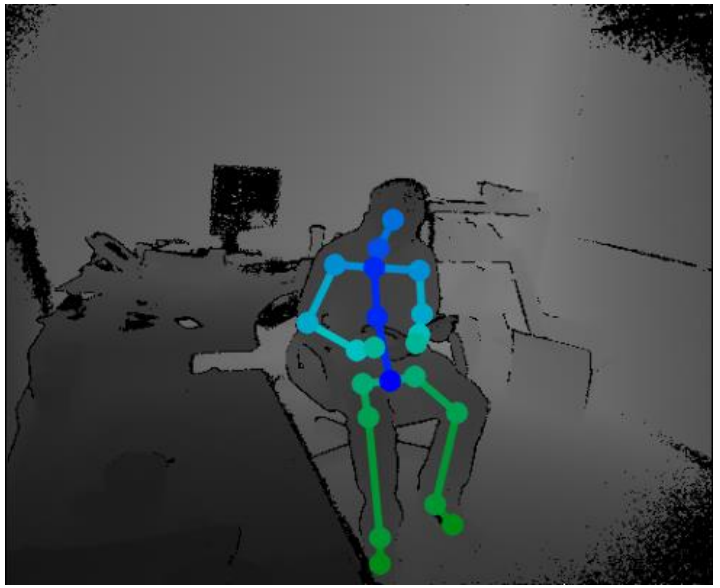
- **Pandora**
  - Goal: *Head and Shoulder Pose Estimation* task
  - **250k** images (**Full HD RGB** and **depth**) of the upper body
  - **22 subjects**
  - Annotations: **head** and **shoulder** angles (*yaw*, *pitch* and *roll*)
    - **Head**: ±70° roll, ±100° pitch and ±125° yaw
    - **Shoulder**: ±70° roll, ±60° pitch and ±60° yaw
  - Challenging **camouflage** (glasses, scarves, caps…)



1. *Borghi, G., Venturelli, M., Vezzani, R., & Cucchiara, R. " Poseidon: Face-from-depth for driver pose estimation", CVPR 2017*

Depth-based datasets

- **Watch-R-Patch*** (currently under submission)
  - **Refined annotated version** of the *Watch-n-Patch* (*Watch-R(efined)-Patch*) dataset
  - 20 (train) + 20 (test) sequences **manually** annotated in body joint positions
    - Sequences equally split between office and kitchen
    - Annotation every 1 or 3 frames
  - The total number of annotated frames is about 3k
    - **Train**: 1135
    - **Validation**: 766
    - **Test**: 1428

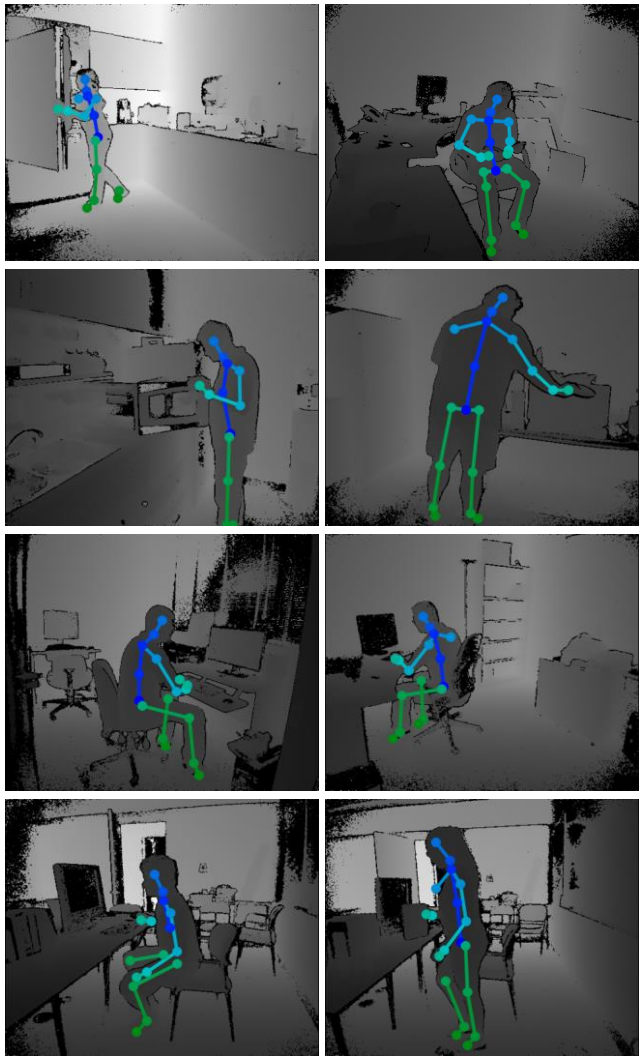


**originally not created for the Human Pose Estimation task*

Depth-based datasets

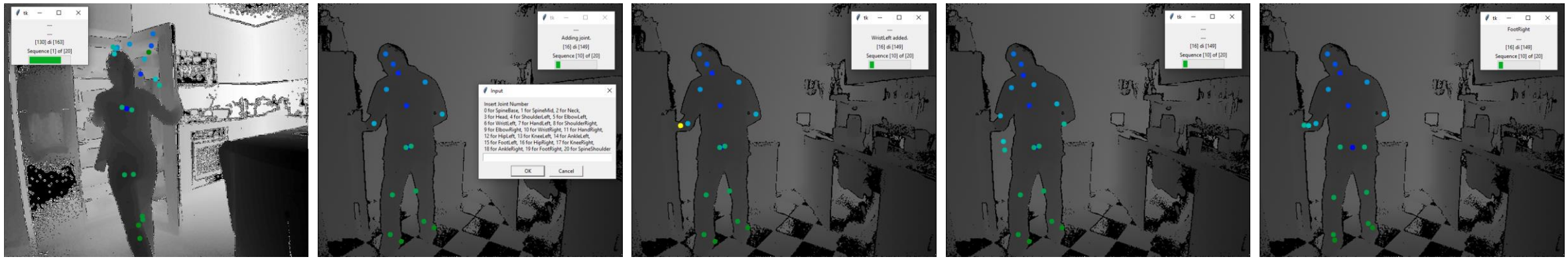
Table 1. Statistics of the *Watch-R-Patch* dataset.

Split	Sequences		Frames	Annotated frames	Modified joints (%)	mAP
	Kitchen	Office				
Train	data_02-28-33 data_03-22-44 data_03-38-20 data_03-42-37 data_03-46-49 data_03-50-38 data_04-07-17 data_04-17-37 data_04-31-11 data_04-34-13	data_01-50-09 data_03-28-59 data_04-02-43 data_04-31-13 data_04-41-55 data_04-47-41 data_04-56-00 data_05-31-10 data_05-34-47 data_12-03-57	3385	1135	0.757	0.574
Val	data_01-52-55 data_03-53-06 data_04-52-02	data_02-32-08 data_02-50-20 data_03-25-32	995	766	0.643	0.600
Test	data_02-10-35 data_03-45-21 data_04-13-06 data_04-27-09 data_04-51-42 data_05-04-12 data_12-07-43	data_03-04-16 data_03-05-15 data_03-21-23 data_03-35-07 data_03-58-25 data_04-30-36 data_11-11-59	2213	1428	0.555	0.610
Overall	-	-	6593	3329	0.644	0.595



Annotation Tool

- *Watch-R-Patch* dataset is manually annotated through an annotation tool
- We develop a system that shows the **original body joints on top of the acquired depth images**
- The user is able to **move the incorrect joints** in the proper positions using the mouse in a *drag-and-drop* fashion
- Once every incorrect joint has been placed in the correct location, the user can save the new annotation and move to the next frame



The annotation tool is publicly released: https://github.com/aimagelab/human-pose-annotation-tool

Depth-based Methods for Human Pose Estimation

Depth-based Methods for HPE

- These methods generally rely on:
  - **Generative models**
    They estimate the pose by finding correspondences between the pre-defined body model and the input 3D point cloud. A body template is required.
    - ICP algorithm
    - Template Fitting (Gaussian mixture models)

  - **Discriminative models**
    They directly estimate the position of the body joints.
    Viewpoint-invariant features are needed.
    - Classificaton of pixels into body parts (Shotton *et al.* with Random Forests)
    - Direct regression of the coordinates of body joints
    - CNN and RNN for classification or regression

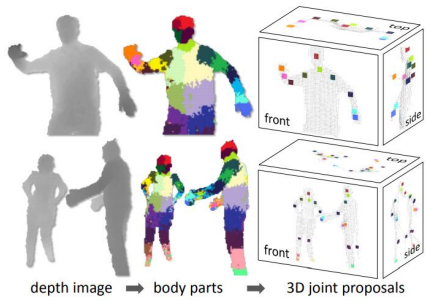
Depth-based Methods for HPE

Time

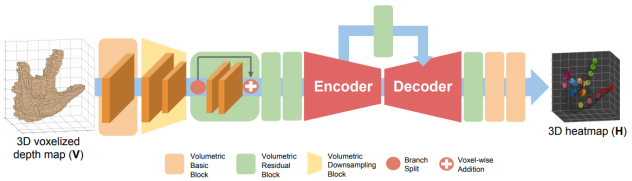
Watch-n-Patch *(CVPR2015)*

ITOP *(ECCV 2016)*

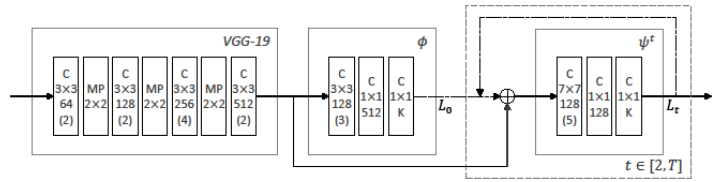
Shotton *et al.* *(CVPR 2011)*



PoseNet *(CVPR 2017)*

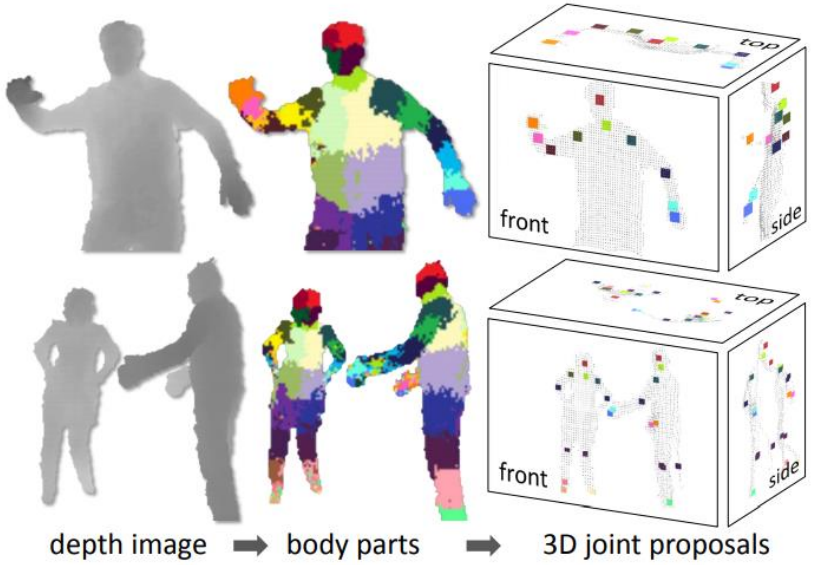


New research trend *(2019)*



Human Pose Estimation on Depth Maps

- **Real-Time Human Pose Recognition in Parts from Single Depth Images** (Shotton *et al.*, CVPR 2011)
  - *Milestone* for the HPE on Depth Maps
  - Embedded in the commercial SDK of the *Microsoft Kinect* (both versions)
  - Real time performance: the system runs at 200 fps on consumer hardware
  - No temporal information exploited
  - Key elements:
    - **Task**: Intermediate body parts representation
    - **Features**: 3D translation invariant depth image features
    - **Classifier**: *Randomized Decision Forests*
    - Joint Position proposals
  - The dataset is not publicly released



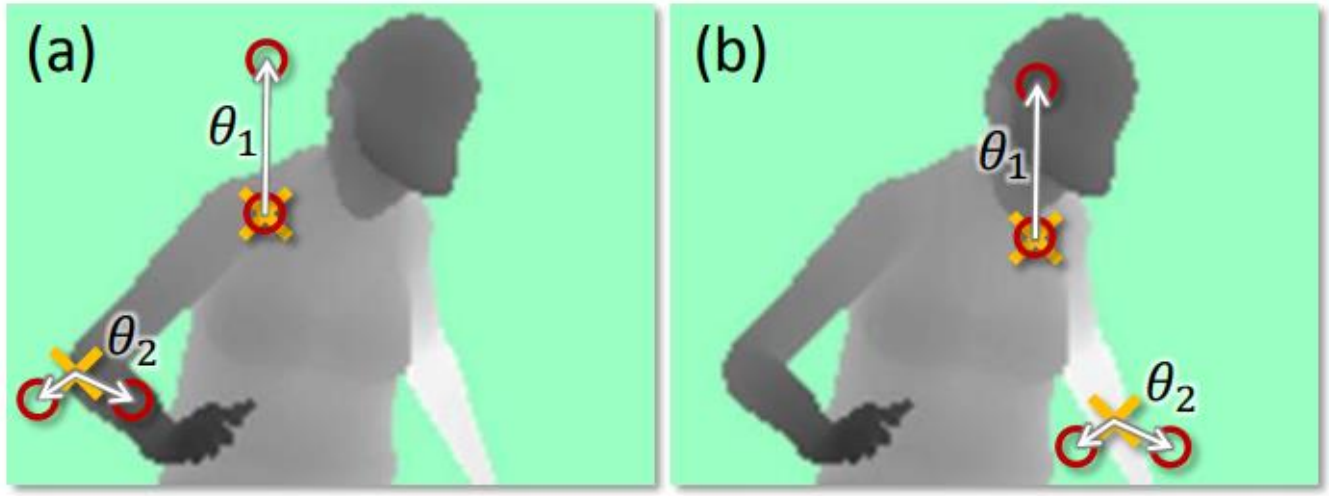
1. Shotton, J., Fitzgibbon, A. W., Cook, M., Sharp, T., Finocchio, M., Moore, R., ... & Blake, A. "Real-time human pose recognition in parts from single depth images" CVPR 2011

Human Pose Estimation on Depth Maps

- **Intermediate body part representation**:
  - Several localized body part are defined, that densely cover the whole body
  - These parts are related to skeletal joints of interest
  - 31 body parts. These parts should be sufficiently small to accurately localize body joints, but not too numerous as to waste capacity of the classifier
  - HPE task becomes a classification problem: a classifier is needed

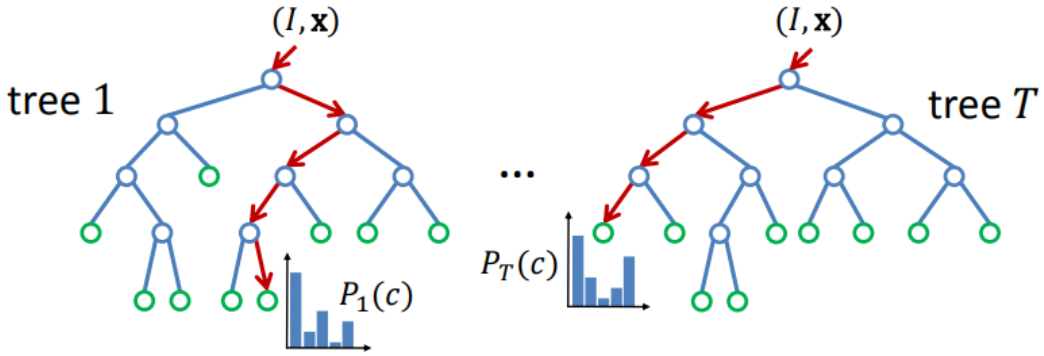
Human Pose Estimation on Depth Maps

- **Depth image features**:
  - Features are based on a **comparison**: $f_\theta(I, \mathbf{x}) = d_I\left(\mathbf{x} + \frac{\mathbf{u}}{d_I(\mathbf{x})}\right) - d_I\left(\mathbf{x} + \frac{\mathbf{v}}{d_I(\mathbf{x})}\right)$
  - $d_I(x + u, v)$ is the depth value for the two **offsets**
  - $1/d_I(x)$ is a **normalization parameter** for the translation invariance
  - These features provides only a weak signal about which part of the body the pixel belongs to, so they are combined with a strong classification mechanism



Human Pose Estimation on Depth Maps

- **Randomized Decision Forests**:
  - A forest is an ensamble of *T* decision trees
  - Each tree is a set of splits and leaf nodes:
    - **Split**: feature f_θ + threshold τ
    - **Leaf**: learned distribution $P_t(c|I,x)$



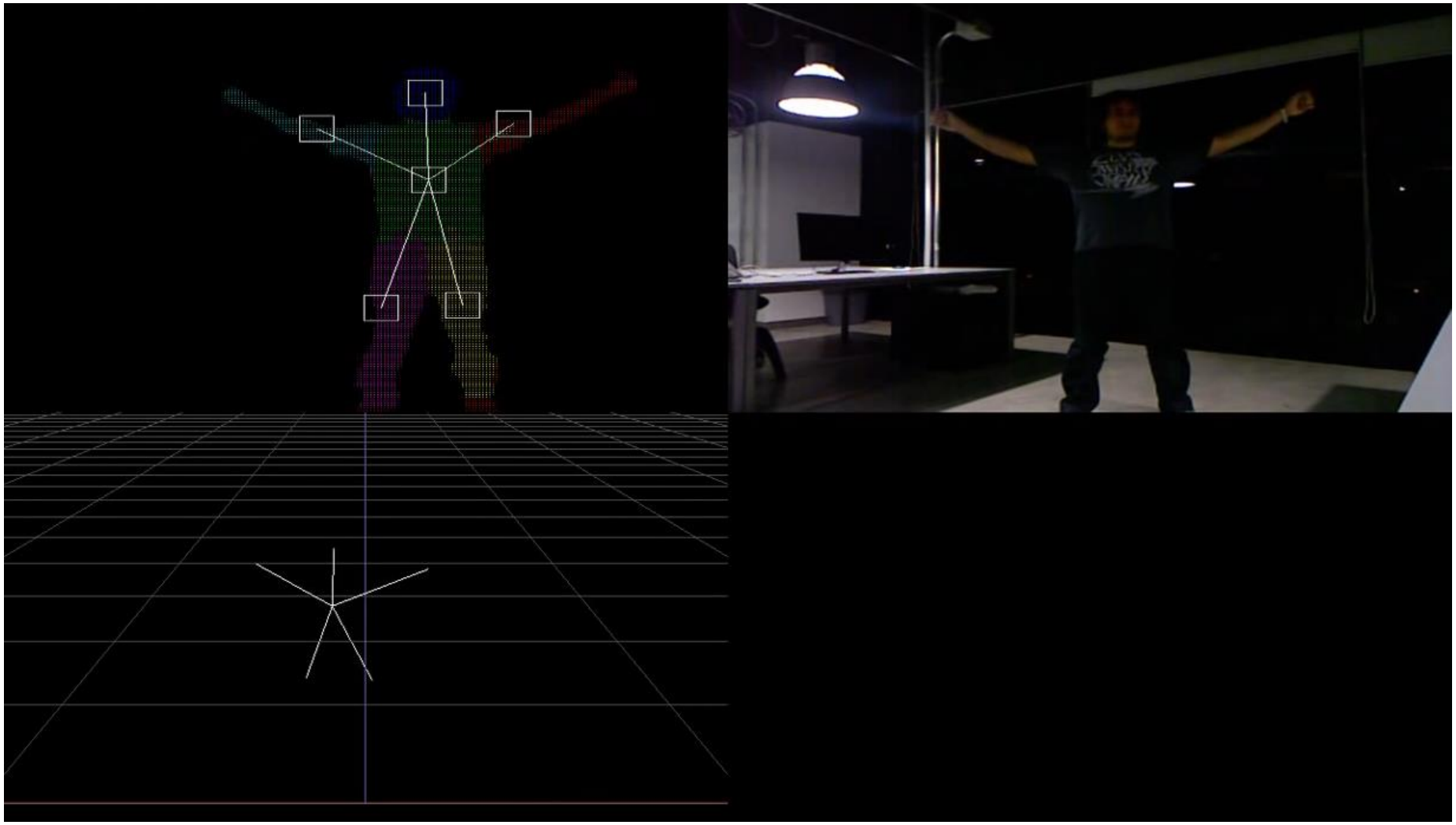
  - The distributions are averaged together for all trees in the forest to give the final classification

$$P(c|I,\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|I,\mathbf{x})$$

- **Joint body proposals to define the final body joints**:
  - At this point, the pixels are divided in different groups = classes (with outliers)
  - For each group, a «center» (the body joint) is defined
  - The **mean shift** algorithm with a **weighted Gaussian kernel** to accumulate 3D global centers

Human Pose Estimation on Depth Maps

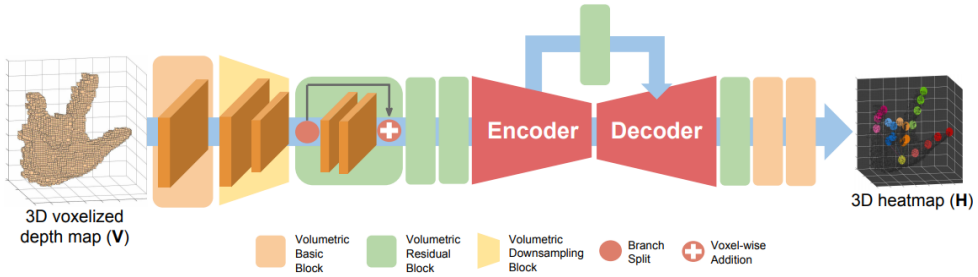
Demo video of the method of Shotton *et al.*



1. *Credits to Alfonso Ros (https://www.youtube.com/watch?v=gwovIigpkTo)*

Human Pose Estimation on Depth Maps

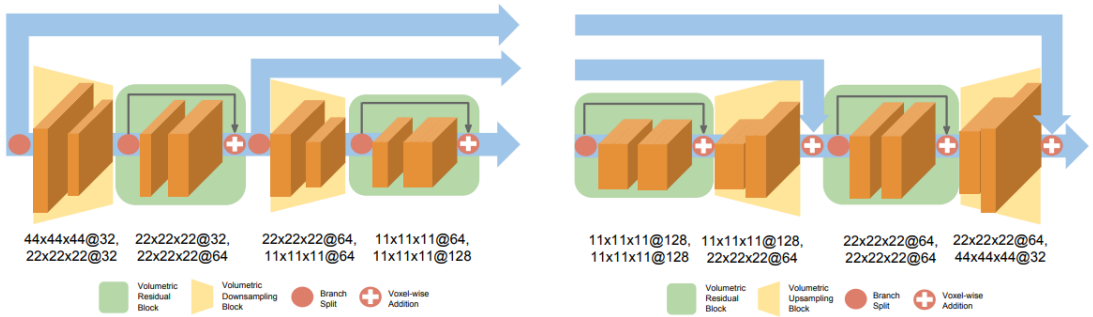
- **V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation from a Single Depth Map** (Moon *et al.,* CVPR 2018)
  - This method is primarly created for the *Hand* Pose Estimation task
  - Then, it is applied also on the *Human* Pose Estimation task
  - Initial considerations:
    - Depth-maps are often used as 2D images as input for CNNs
    - In depth maps there are **perspective distortion issues**
    - 3D coordinates from 2D images is a highly non-linear mapping → problems in learning procedures
  - They propose to use a **3D CNN on voxel**
  - The overall architecture of the *V2V-PoseNet*:



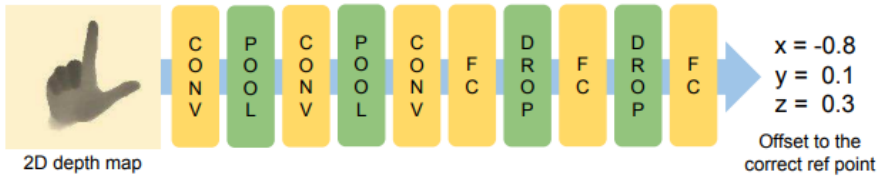
1. *Moon, G., Yong Chang, J., & Mu Lee, K. "V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map", CVPR 2018*

Human Pose Estimation on Depth Maps

- The *encoder-decoder* architecture is the following



- **Voxelization**: it is the conversion of the 2D depth map in 3D volumetric forms.
  The continuous space is discretized, a cubic 3D space is created around a reference point
- How to find the reference point?
  - Using **ground truth positions**, but it is not too realistic (in real-world apps there is not the GT)
  - Computing the **center-of-mass**, but there are problems with occlusions
    - The cubic box can contain only a part of the target object
  - They propose a specific **2D CNN to refine the center-of-mass** previously computed

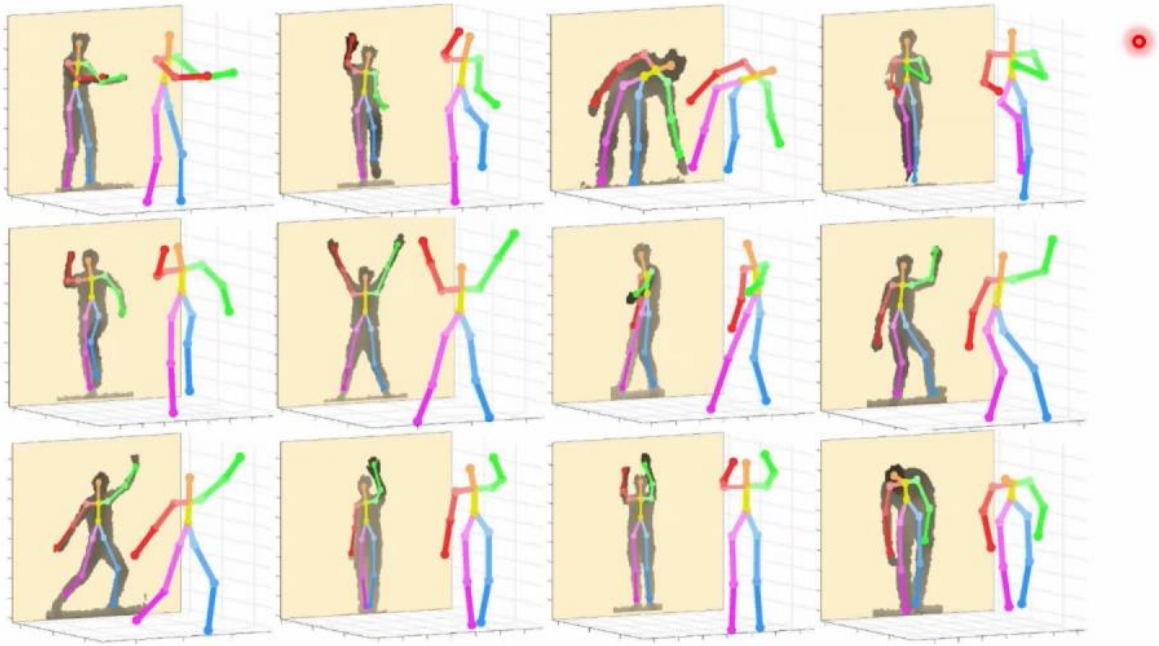


Human Pose Estimation on Depth Maps

- **Voxel values are set to 1** if the voxel is occupied by any depth point, **0 otherwise**
- The **Mean Square Error** function is adopted as loss function (between 3D joint heatmaps, the mean of the gaussian peak is positioned at the ground truth joint location)

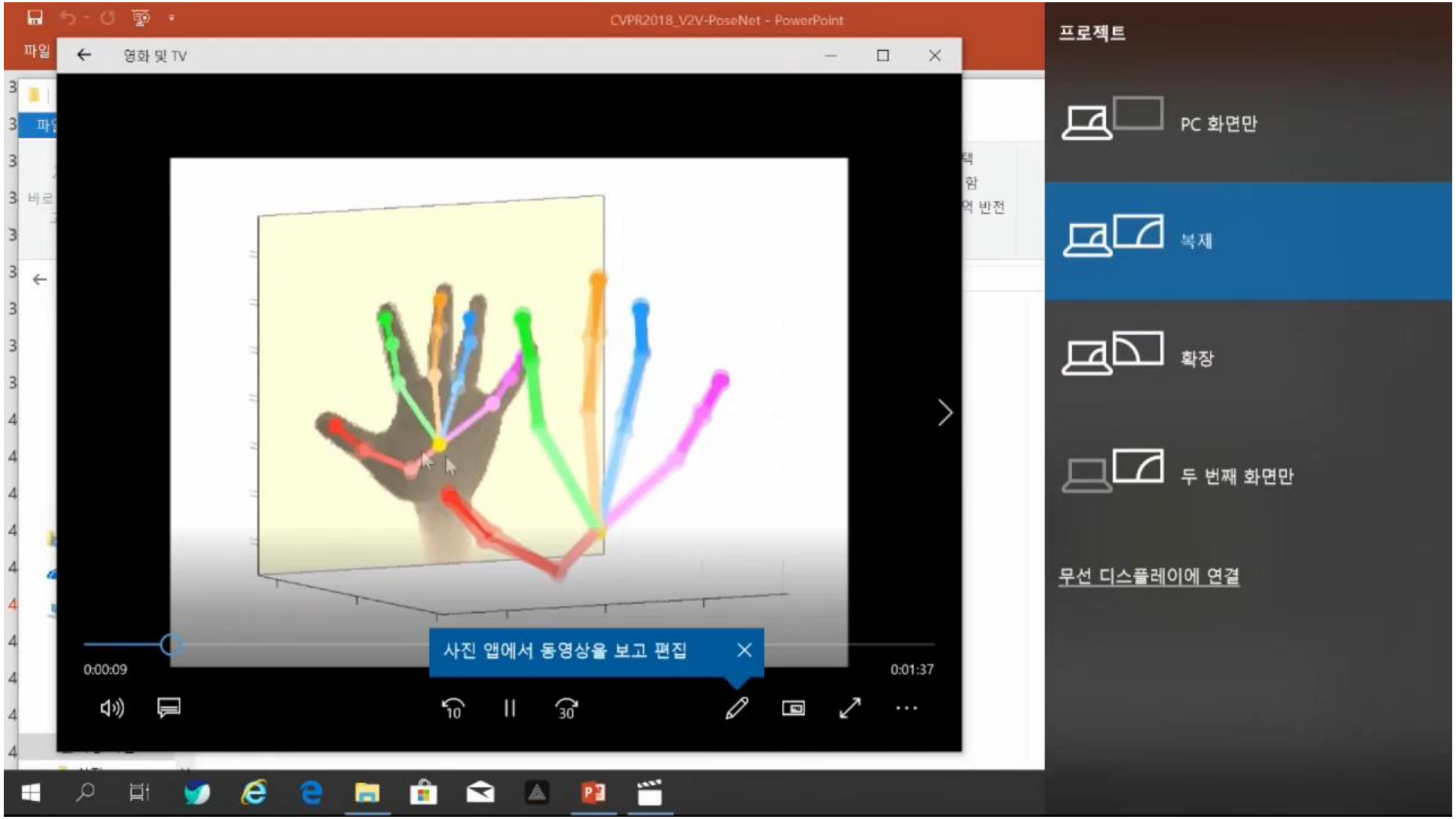
Qualitative Results

- ITOP dataset: Front View



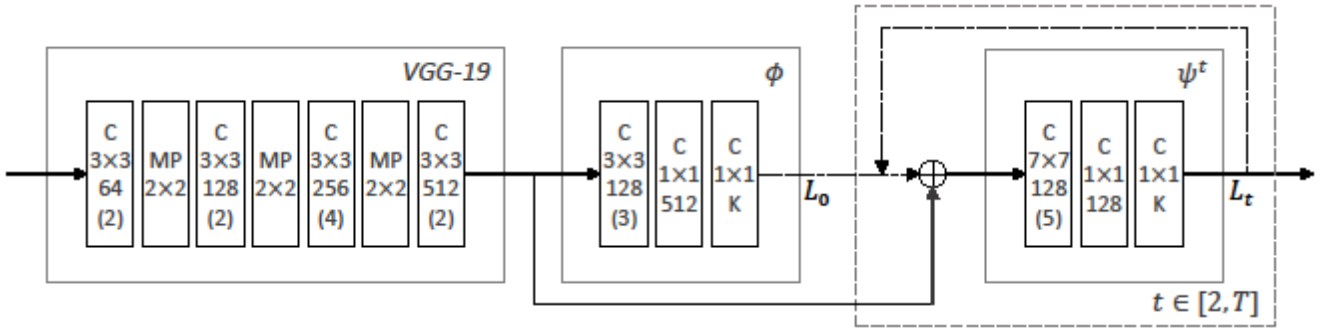
ComputerVisionLab Seoul National University

Human Pose Estimation on Depth Maps



Human Pose Estimation on Depth Maps

- **Manual Annotations on Depth Maps for Human Pose Estimation** (*under submission*)



- Taking inspiration from [1], the first part of the architecture is composed of a **VGG-like feature extraction block** which comprises the first 10 layers of VGG-19
- Then, there are two layers that **gradually reduce the number of feature maps** to the desired value
- A convolutional block ϕ produces an initial coarse prediction of human body joints analyzing the image features extracted by the previous block only
- **Multi-stage architecture**: a convolutional block is repeated T-1 times in order to gradually refine the body joint prediction. This block analyzes the concatenation of the features extracted by the feature extraction module and the output of the previous stage, refining the earlier prediction

1. *Zhe Cao and Tomas Simon and Shih-En Wei and Yaser Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields", CVPR 2017*

Human Pose Estimation on Depth Maps

- **Training procedure**:
  - Objective function: $L^t = \sum_{k=1}^{K} \alpha_k \cdot \sum_{\mathbf{p}} \|\mathbf{P}_k^t(\mathbf{p}) - \mathbf{H}_k(\mathbf{p})\|_2^2$
  - K is the number of considered joints
  - α_k is a binary mask with value 0 is the annotation of joint k is missing
  - $P \in \mathcal{R}^2$ is the spatial location
  - $P_k^t(p)$ and $H_k(p)$ are the predicted and ground truth heatmaps

$$\mathbf{H}_k(\mathbf{p}) = e^{-\|\mathbf{p}-\mathbf{x}_k\|_2^2 \cdot \sigma^{-2}}$$

  - $x_k \in \mathcal{R}^2$is the location of joint k, σ is the parameter to control the Gaussian spread
  - Applying the supervision at every stage of the network mitigates the *vanishing gradient* problem

Human Pose Estimation on Depth Maps

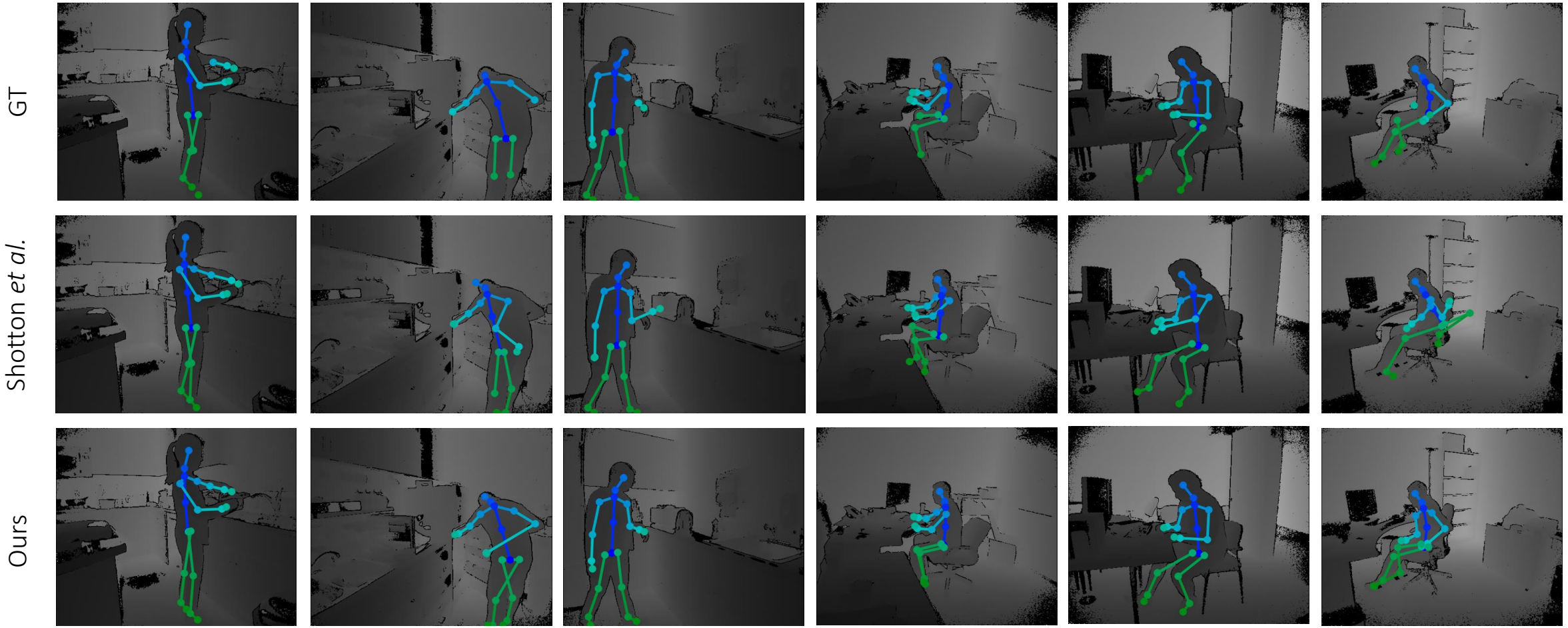
- **Evaluation procedure** (on the *Watch-R-Patch* dataset):
  - COCO keypoints challenge: *mean Average Precision* (mAP)
  - The mAP is defined as the 10 average precision computed with different OKS
  - *Object Keypoint Similarity* (OKS) thresholds:

$$\mathrm{OKS} = \frac{\sum_i^K [\delta(v_i > 0) \cdot \exp \frac{-d_i^2}{2s^2 k_i^2}]}{\sum_i^K [\delta(v_i > 0)]}$$

    - δ_i is the Euclidean distance between keypoints
    - s is the area containing all the keypoints, $k_i = 2\sigma_i$
    - v_i is the visibility flag (0: keypoint not labeled, 1: keypoint labeled)

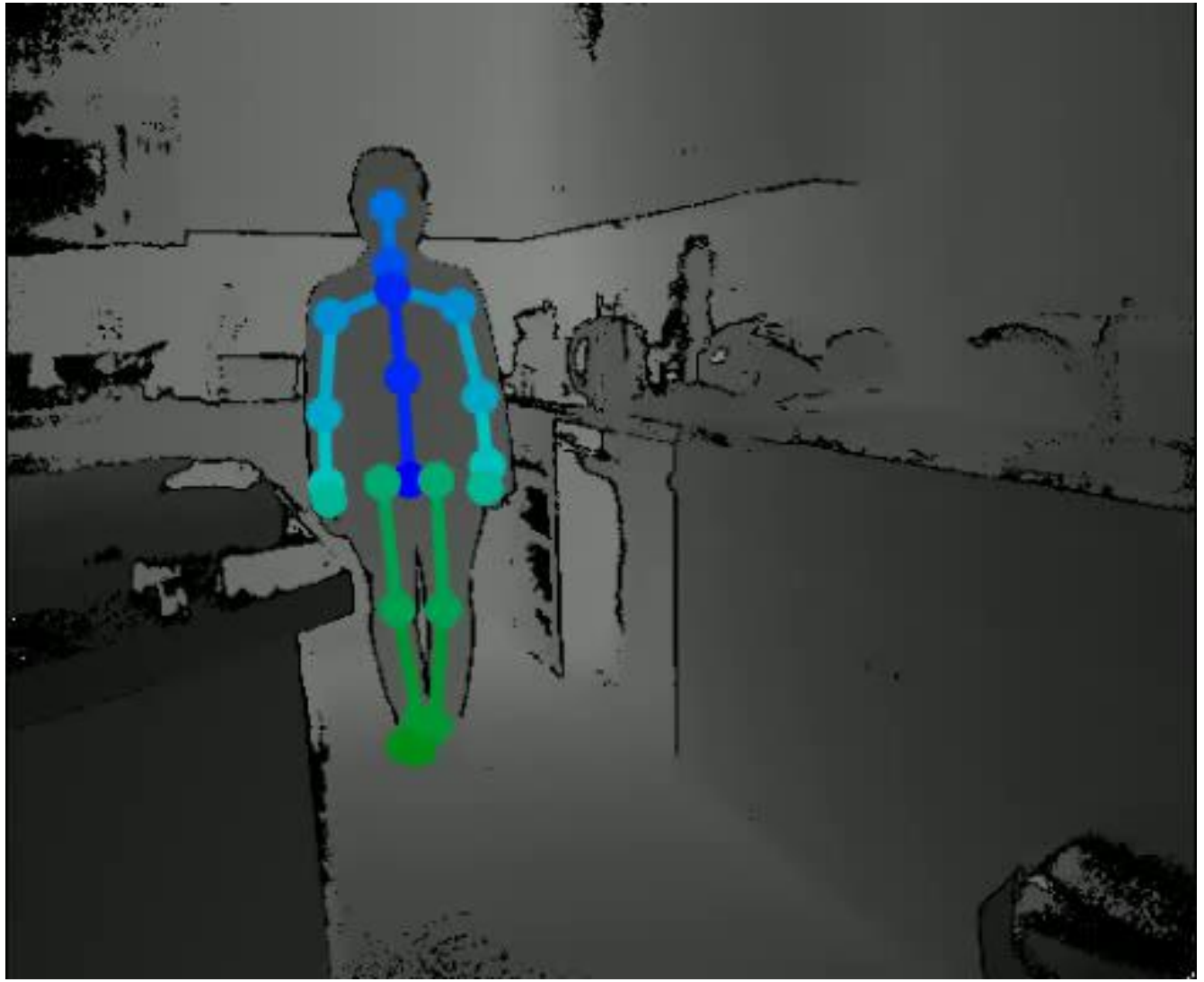
	Shotton *et al.* [13]	Ours$_{orig}$	Ours$_{last}$	Ours$_{blk}$	**Ours**
AP$^{\mathrm{OKS=0.50}}$	0.669	0.845	0.834	0.894	**0.901**
AP$^{\mathrm{OKS=0.75}}$	0.618	0.763	0.758	0.837	**0.839**
mAP	0.610	0.729	0.726	0.792	**0.797**

Results



AImage Lab

Human Pose Estimation on Depth Maps

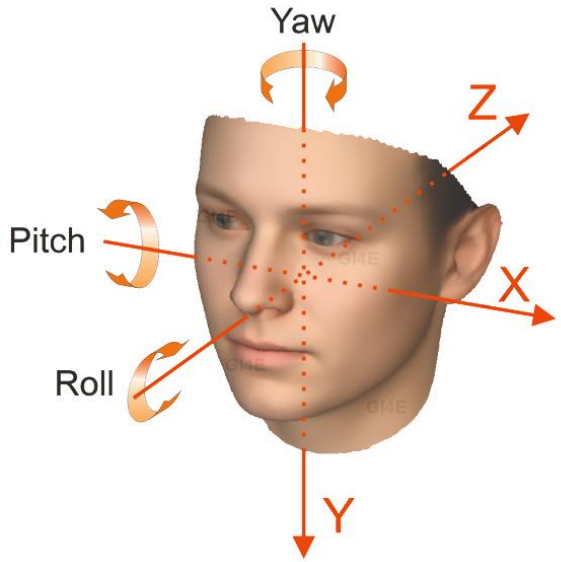


Depth-based Methods for Head (and Shoulder) Pose Estimation

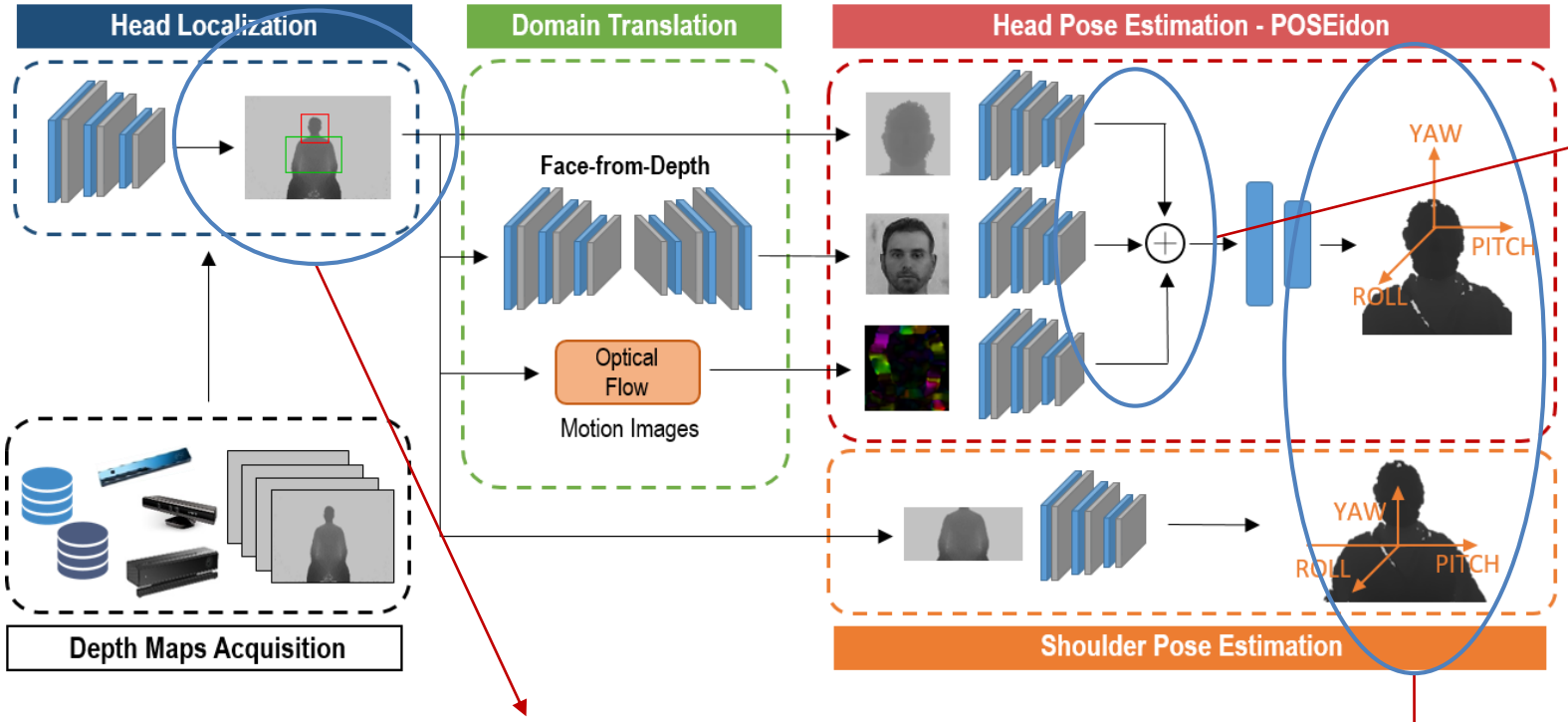
Head and Shoulder Pose Estimation

It is the ability to infer the orientation of the person's head (shoulders) relative to the view of a camera.

The orientation is expressed through 3D angles: *yaw*, *pitch* and *roll*



Head and Shoulder Pose Estimation



Training Procedure

- Double-step procedure
  - 1st: each individual network is trained
  - 2nd: the last fc layer is removed, networks are then merged through a *conv* and *concat* operations:

$$y^{cat} = [x^a | x^b], \qquad d^y = d_a^x + d_b^x$$

$$y^{cnv} = y^{cat} * k + \beta, d^y = \frac{(d_a^x + d_b^x)}{2}$$

x^a , x^b: feature maps

d_a^x , d_b^x: feature channel

- Loss function:

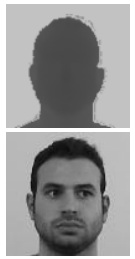
$$L = \sum_{i=1}^{3} |w_i \cdot (y_i - f(x_i))|_2^2$$

$$\boldsymbol{w} = [0{,}2, 0{,}35, 0{,}45]$$

Head crop formula

$$w, h = \frac{f_{x,y} \cdot R}{D}$$

- $f_{x,y}$ are the horizontal and vertical focal lengths
- R is the average value representing the width of a face (200mm)
- D is the distance between the acquisition device and the head in the scene.



Final Outputs

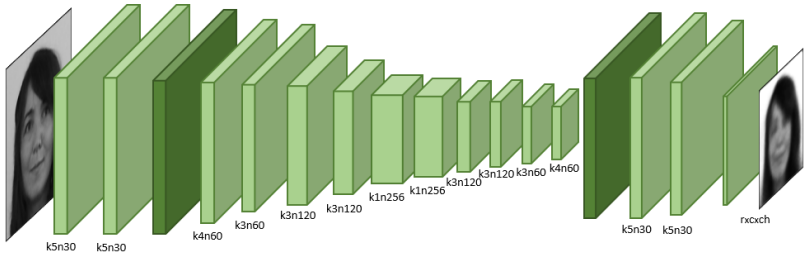
- 3D angles *yaw*, *pitch* and *roll* for head and shoulders

1. Borghi, G., Venturelli, M., Vezzani, R., & Cucchiara, R. "Poseidon: Face-from-depth for driver pose estimation", CVPR 2017

Face Translation: *Face-from-Depth*

- The main idea is to **add knowledge** (the *generated* faces) at training and testing time, **to improve the performance**
- We propose (two versions) of a **new** neural network called *Face-from-Depth*:

FfD [v1]: CNN (CVPR 17)



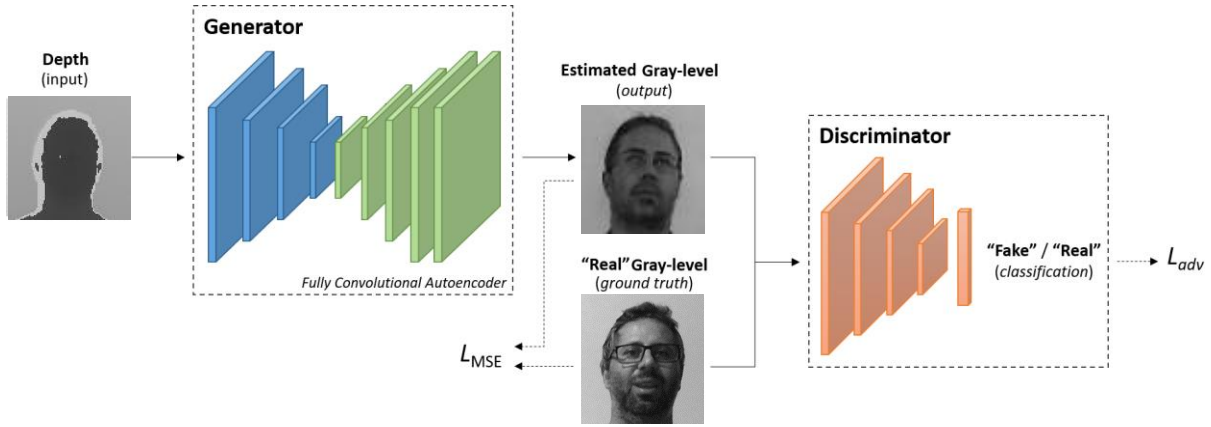
- Elements from **autoencoder** and **CNN**
- **Weighted Loss**:

$$L = \frac{1}{R\,C}\sum_{i}^{R}\sum_{j}^{C}(|y_{ij} - y'_{ij}|_2^2 \cdot w_{ij}^{\mathcal{N}})$$

$$\mathcal{N}:\ \mu = \left[\frac{R}{2}, \frac{C}{2}\right]^T \quad \sum = \mathbb{I} \cdot \left[\left(\frac{R}{\alpha}\right)^2, \left(\frac{C}{\beta}\right)^2\right]^T \qquad \alpha = 3.5, \beta = 2.5$$

- *Bi-variate Gaussian* prior mask to highlight the central area

FfD [v2]: *conditional GAN* (PAMI 18)



- **Min-max game** (Generator - Discriminator):

$$\min_{\theta_d}\max_{\theta_g}\mathbb{E}_{x\sim p_{gray}(x)}[\log(G(x))] + \mathbb{E}_{y\sim p_{dpt}(y)}[\log(1 - G(D(y)))]$$

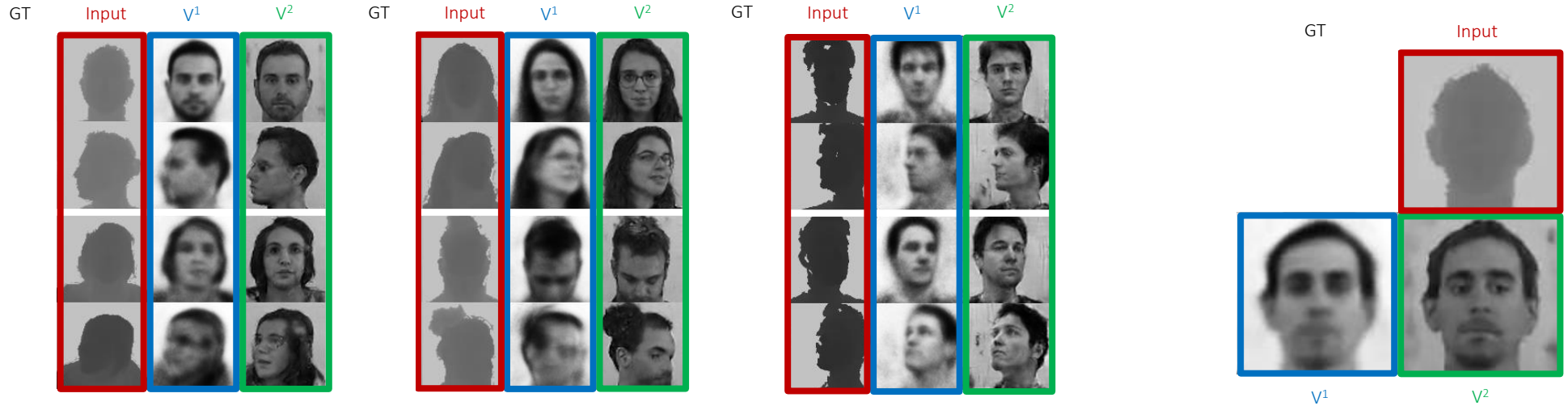
- **2 loss functions**:

$$L_{MSE}(s^g, s^d) = \frac{1}{N}\sum_{i=1}^{N}|G(s_i^g) - s_i^d|_2^2$$

$$L_{adv}(y, t) = -\frac{1}{N}\sum_{i=1}^{N}[t_i \log y_i + (1 - t_i)\log(1 - y_i)]$$

Face Translation: *Face-from-Depth*

- **Visual** and **numerical** comparisons between FfD [v1] and FfD [v2]:



Pandora dataset

Biwi dataset

Dataset	Method	Norm ↓		Difference ↓		RMSE ↓			Threshold ↑		
		L_1	L_2	Abs	Squared	linear	log	scale-inv	1.25	2.5	3.75
Biwi	FfD[1]	33.35	2586	0.454	24.07	40.55	**0.489**	**0.445**	0.507	**0.806**	**0.878**
	FfD	**24.44**	**2230**	**0.388**	**19.81**	**35.50**	0.653	0.610	**0.615**	0.764	0.840
Pandora	FfD[1]	41.36	3226	0.705	46.00	50.77	0.603	**0.485**	0.263	0.725	0.819
	pix2pix[2]	19.37	1909	**0.468**	24.07	30.80	0.568	0.539	0.583	0.722	0.813
	AVSS[3]	23.93	2226	0.629	34.49	35.46	0.658	0.579	0.541	0.675	0.764
	FfD + U-Net	23.75	2123	0.653	34.96	33.89	0.639	0.553	0.555	0.689	0.775
	FfD	**18.21**	**1808**	0.469	**22.90**	**28.90**	**0.556**	0.501	**0.605**	**0.743**	**0.828**

1. *G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara, "Poseidon: Face-from-depth for driver pose estimation", CVPR 2017*
2. *P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks", CVPR 2017*
3. *M. Fabbri, S. Calderara, and R. Cucchiara, "Generative adversarial models for people attribute recognition in surveillance", AVSS 2017*

POSEidon system

- Demo video about POSEidon system in a real in-cabin environment
- Acquisition device: *Microsoft Kinect One*