Estimating Human pose from depth images using Convolutional Neural Networks

Both Eyes Open

Bård-Kristian Krohg

Thesis submitted for the degree of
Master in Informatics: Robotics and Intelligent Systems
60 credits

Institute for informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2021

Estimating Human pose from depth images using Convolutional Neural Networks

Both Eyes Open

Bård-Kristian Krohg

Estimating Human pose from depth images using Convolutional Neural Networks

http://www.duo.uio.no/

Printed: X-press printing house

Estimating Human pose from depth images using Convolutional Neural Networks

Bård-Kristian Krohg

24th May 2021

Abstract

This work is part of a larger project where we explore bringing robotics into geriatric care. The goal of this project is to create a robotic system that can assist in optimizing the use of caregivers, so they are used where they are needed.

This work will focus on capturing information about the user, anonymization of the data, what data is necessary or ethical to capture, limitations for on-location data processing and what data can be sent for further processing in the cloud, or human analysis.

We will also implement an ethical data-collection suite for the open-source Robotic Operating System, which can be implemented on a wide variety of robots.

Convolutional Neural Networks have been used for solving object recognition in 2D images with great success. This work aims to use the same techniques to extract 3D human pose from depth images in real-time. We will use two multi-staged CNNs, one to encode the location of each joint, and another to encode the association between the joints to do this.

Preface

First, I would like to thank my supervisors Jim Tørresen and Ryo Kurazume, for their support, guidance, and patience during the development of this project. Second, my HR manager Marit Flendstad Kruse, for her assistance in letting me combine work with the writing of this thesis. Last, I would like to thank everyone at the Kurazume-lab for their welcome and help during my stay at Kyushu University, and my friends and family for proofreading and encouragement.

The subtitle, Both Eyes Open, has a double meaning: In this paper, we will explore the world in 3D. The biological way to achieve depth vision is by using two eyes, hence both eyes open. The other way to interpret the subtitle is tied with the small figure on the front page. In Japanese, the saying translated as "Both Eyes Open" refers to the Daruma figure.

When one is working toward a goal, such as completing a thesis, one can purchase a Daruma figure from a temple. When bought, the Daruma has blank eyes - they are closed. The buyer then paints in one eye, asking for the Darumas help in completing their goal. In exchange for the Darumas help, the buyer promises to paint in the other eye. One significant detail about the Daruma is that it is weighted on the bottom. If it should ever falter and fall over, it will right itself back up and continue on its way to completing the goal.

The ability to right yourself up for every setback has been of particular inspiration to me during my work on this thesis. This is why I have placed the figure on the front page; as a personal helper.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As life expectancy increases in Norway, so does the population who needs geriatric care either at home, or in a geriatric facility. Accodring to a communal health sruvey [10], it is projected that the increasing senior population in Oslo will lead to increased pressure on the healthcare services. To mitigate the need for help at home, it is important to encourage health-promoting activities, and uncover what and where preventative measures is needed. The Multimodal Elderly Care System (MECS) is proposed as a solution where data gathered in users homes, guide where and what help is needed from healthcare personel – be it preventative or acute.

The MECS is envisioned as a small, mobile unit which can be introduced to any home. This eliminates the user from the operational loop of the unit, making the data gathering process robust to forgetfulness or physical ability of the user. One of the key information points gathered by the MECS will be the users physical pose. The users physical pose informs the system of the folowing:
Body language, enabling the possibility of smooth Human-Robot Interaction (HRI) when moving around, or determining the intent of the user.
Physical state, informing the MECS wether the user is in need of acute help.
History of natural posture, which could be invaluable information for a physical therapist or doctor to develop personalized training programs preventing muscle degradation.
Activity recognition, helping the MECS decide on what actions to execute. For example, reminding the user to take their prescribed

medication if they forget.

All these key points could be captured by a 2D system, however a 2D representation of a pose would not be robust to different viewing angles. A 3D representation could define the origin of two poses' coordinate system to the same landmark in each pose, making it is easy to compare the two poses by for example the euclidian distance between the poses' corresponding landmarks.

Estimating human pose in 3D, or Motion Capture (mocap) is a well known area of research. However, mocap is expensive, and requires a large amount of physical hardware. The industry standard is to use an elaborate mocap studio that requires multiple expensive cameras, a large area, and specialized software. Therefore the application areas for motion capture are currently mostly limited to research, movie-, and videogame-making. The mocap problem deals with finding a representation of an actor, creature, or object that can be used in animation. This representation is often a *rigged* skeleton with bones that define the movement of the animated character [13]. The movements of the computerized rig is mapped to the movements of an observed actor in the real world and recorded.

1.1 Research Goals

This work explores the possibility of using a single depth camera to solve the mocap problem of estimating human pose in 3D. To be viable in a mobile unit with limited processing capability, the resulting method needs to be lightweight and fast enough to process human motions. The resulting method must also be accurate enough for healthcare personel or the MECS itself to extract reliable and useful information about the pose.

The method presented in this thesis will therefore be evaluated using the following goals:

1. Propose a lightweight system for recognizing human pose in 3D based on depth images.

2. Design and develop an architecture which could be deployed to systems with limited hardware, and provide useful information.

3. Evaluate the preformance of the system and find room for improvements for it.

1.2 Contributions

The main contribution of this work is the proposal of using a shallow CNN trained on depth images, in combination with a novel articulation network to define human pose in 3D. The CNNs main task is to propose a set of poses present in the depth image. These poses are refined in the articulation network, which allows for a shallower CNN architecture. This leads then to fewer parameters in the system, which leads to a more lightweight method that should preform faster on limited hardware.

1.3 Thesis Structure

Chapter 2

Background

The classical approach to extract useful information from an image has been to find mathematical definitions for features that describe the information. The features could, for example, be lines, circles, or edges. Circles can be used to find coins, where the diameter denotes the value. Lines can be used to find how many fenceposts are in a fence. Traditional mathematical models include the Hough Transform [6] for detecting lines or circles, the Sobel operator to detect gradients, and in return, edges, or the Gray Level Co-occurrence Matrix for detecting texture features. In common for all these methods is that they are well defined and find precisely *one* type of feature that was previously specified. At a higher level, hand-crafted feature descriptors have been made. They look for a specific set of features in an image to identify unique locations. Some well-known examples are the SIFT [12], SURF [2] and ORB [15] feature descriptors. They have been used successfully in applications such as combining images into panoramas or combining different views to a 3D model, also known as Structure from motion [19].

Hand-crafted features have also been used in machine learning scenarios. As an example, Haar-like features were demonstrated to efficiently find faces in [20]. Here, the machine learning model decided which of a given set of features to use to classify whether the frame contained a face or not. In using both the Sobel operator and using Haar-like features, a filter is passed over the image. In the case of the Sobel operator, this is an actual convolution of the image with the filter, whereas Haar-features are extracted using a sliding window. This is the intuition that is used in Convolutional Neural Networks, discussed next.

2.1 Convolutional Neural Networks

Instead of using hand-crafted features, CNNs define the features they use, when they are trained. This is one of the reasons why CNNs needs fairly large training sets, as well as taking a long time to train. Therefore, a popular approach is to reuse the first few primitive layers (the first layers encountered in a forward-pass) from other models. This is done by first training a neural network for a specific task, for example recognizing a handful of different types of objects in an image. Then, the primitive layers with their parameters are "chopped off" the network. Since these layers only have learned primitive features, these learned features can be input to specialize a different network, which in turn needs less training time, since it has already learned the simple features.

CNNs are widely used in image classification tasks because they look at a collection of spatially connected pixels. This means CNNs are robust to the placement of the object in the frame, and even partial occlusions. The deeper the network is, the more complex features emerge. In addition, the *receptive field*, the patch of the original image that affects the feature, becomes larger. For convolutional layers that is, not pooling layers.

When designing a CNN, it is often helpful to have in mind exactly what one can imagine should be detected in each layer. In 2D images, the first few layers of a deep CNN have been shown to often mimic behavior of the simple handmade feature-extractors such as the ones discussed above. However, such features can be quite different in 3D. Instead of edge-detectors, one can imagine plane-detectors, or other detectors unique to 3D objects. Additionaly, 3D depth images only have a single channel, where as most 2D CNNs have been trained on 3-channel RGB images. [17] argues that it is therefore better to train such networks from scratch.

2.2 Depth Images

In contrast to color, or RGB images, depth images contain information about the distances. Each pixel contains a measure of the distance from the camera x,y plane to the real world point projected through the pixel.

TODO> about loss of accuracy at distance TODO> about reprojection from world to image-coordinates TODO> about occlusion and visual hull still being a thing

2.3 Pose Estimation

Much research has been done in estimating human pose in two dimensions, as quite large datasets have been made such as the MPII, or the Human 3.6M datasets [1, 7].

There have mainly been two ways of finding human pose in an image. One is a top-down approach. Here a number of people are first found in the image, based on some different criteria. For example, the algorithm could first look for faces or silhouettes that resemble humans. Then, the human pose is iteratively expanded in the area using different algorithms.

The other approach is to use object recognition to find key features for the whole image. Then the recognized landbarks are combined to build up the people instances.

In [3], the second approach is used. Two CNNs, one for landmark localization, and the other for recombination, are trained to estimate human pose. One of the networks produces a *confidence map* for each joint. Each pixel in the confidence map contains the probability, or confidence, that the pixel is part of a person's joint. The other creates Part Affinity Field (PAF)s, which is a map of vectors pointing in the direction of one of M limbs. [1] These maps are assembled by bipartite matching to create the 2D skeletons observed in the scene.

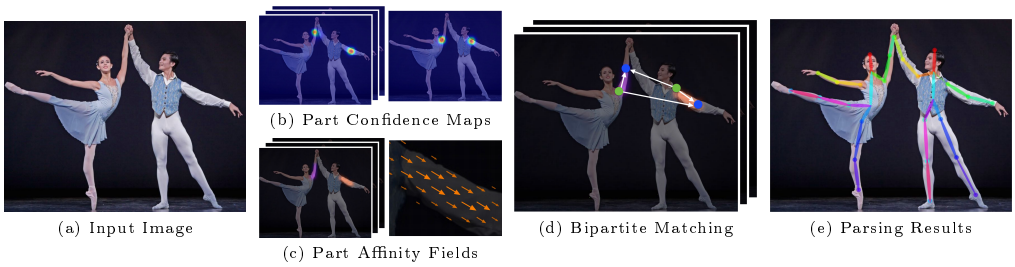


Figure 2.1: The pipeline described in [3]. The input image 2.1a is fed into the two networks, which produce joint detections in confidence maps 2.1b and PAFs 2.1c. Bipartite matching is performed in 2.1d, to determine which detected joints should be connected by a limb. 2.1e shows the finished results.

Using the method described in [3], no information is given for joints that are not accurately detected. If, for example, an extremity is occluded

[1] This work will stick to the convention of using the term *limb* to describe any *connection between any pair of body landmarks*. The body landmarks will be termed *joints*.

together with half of the connecting limb, the extremity will not be part of the output skeleton, even if the joint could be extrapolated from the parts of the limb that is visible. This is also true for undetected joints in the middle of a joint chain. The joint could be extrapolated using the surrounding joints. The problem with joint-extrapolation happens because of the bipartite matching, which does not work if any joint is missing.

TODO> Microsoft pose estimation for kinect using tree-models

2.4 Bipartite Matching

Chapter 3

Datasets

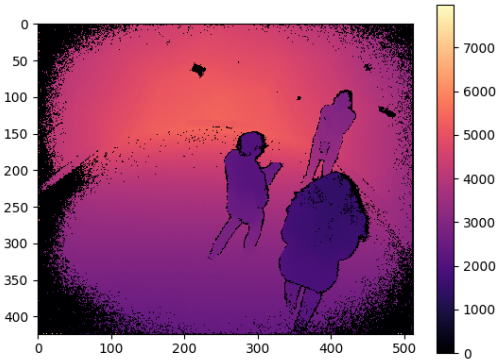
This chapter presents an overview of available datasets suitable for human pose estimation in depth images. The datasets are evaluated according to closeness to real-world conditions, the amount of data and the accuracy of the recorded poses.

3.1 Requirements/Considerations

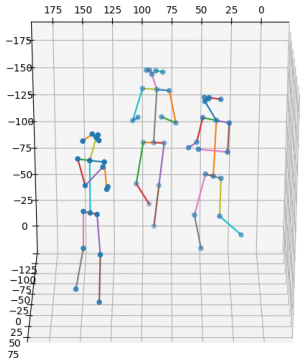
3.2 Existing Datasets

The sequences in the dataset were recorded at approximatley 30 fps. All frames and world coordinate positions at an approximate single timestep is hereby referred to as a *frame*. To create samples, frames from the sequences were extracted at a set of different rates according to the amount of movement in the sequence. For sequences with a lot of movement the rate was set to every 5 seconds, moderate movement every 15 seconds, and for sequences where the subjects stood largely still, samples were extracted every 30 seconds. Each sequence is recorded by 10 different kinects from a height of 1 and 2 meters above ground. This results in a total number of 10x the amount of samples listed in Table 3.1.

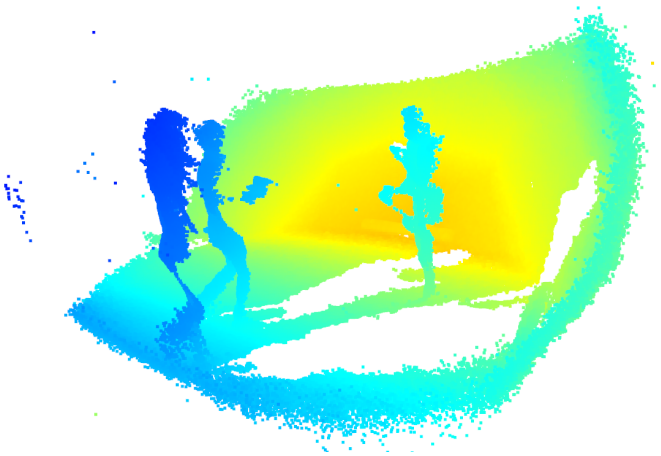
The dataset used in this work was created from source material provided by the Panoptic Studio [8, 9]. Calibration files, predicted ground-truth skeletons and depth images were downloaded using the GNU parallel program [18]. Some problems were experienced, as the Panoptic Studio server was quite unreliable, which resulted in significant delays, as well as corrupted files and in return a smaller dataset than desired.



(a) Depth image from `KINECTNODE6`.



(b) Skeletons defined in the world coordinate system.



(c) The depth image projected to a point cloud.

Figure 3.1: Combination of datapoints for frame 144 in [9]

	Category	Sequence Name	Duration (MM:SS)
Training / Validation	Range of Motion	171204_pose1	17:30
	Range of Motion	171204_pose2	22:30
	Range of Motion	171204_pose3	5:00
	Range of Motion	171204_pose5	15:00
	Range of Motion	171204_pose6	12:50
	Range of Motion	171026_pose1	13:20
	Range of Motion	171026_pose3	4:20
	Social Games	160226_haggling1	8:00
	Social Games	160422_haggling1	8:00
	Social Games	160422_ultimatum1	15:00
	Musical Instruments	160906_band1	1:00
	Musical Instruments	160906_band2	5:00
	Musical Instruments	160906_band3	5:00
	Toddler	160906_ian2	5:00
	Others	170915_office1	3:00
	Others	160906_pizza1	5:00
	Dance	170307_dance5	6:40
Test	Range of Motion	171204_pose4	17:30
	Range of Motion	171026_pose2	9:00
	Social Games	160224_haggling1	5:00
	Toddler	160906_ian1	2:00
	Others	170407_office2	3:00

Table 3.1: Sequences used for training, validation and testing

A set of five complete sequences were set aside as the test set to ensure that the network wouldn't recognize any similar frames or body positions from previous sets.

3.3 Dataset Augmentations

$$K(x,y) = \begin{cases} \frac{2}{\pi\sigma^2}(1 - ((\frac{x}{\sigma})^2 + (\frac{y}{\sigma})^2) & \text{if } |(\frac{x}{\sigma})^2 + (\frac{y}{\sigma})^2| \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Target limb-maps for each frame was created by projecting all instances of a certain type of limb onto an empty matrix of the same dimensions as the depth frame, illustrated in Figure 3.2. A 2D Epanechnikov Kernel with variable σ were then used to calculate the magnitude for each of the 3D vectors pointing in the direction of the limb

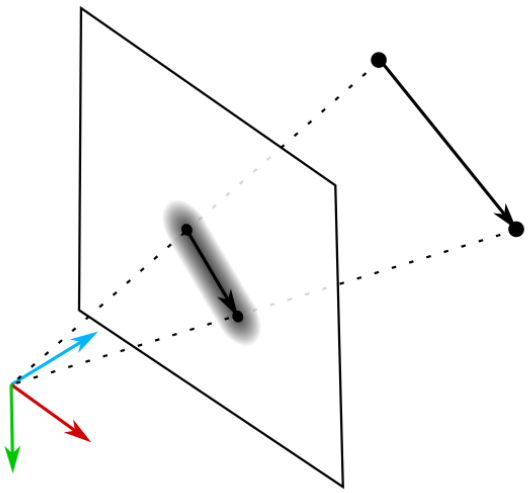


Figure 3.2: A limb is projected onto the image plane

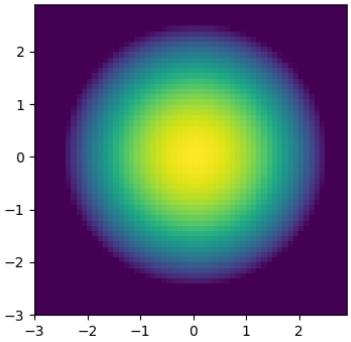


Figure 3.3: A 2D Epanechnikov [4] kernel

in question. The limbs furthest away were calculated first, so the vectors from these would be overwritten by subsequent limbs in case they were behind each other. The σ of the Epanechnikov Kernel was determined by the average depth of a 3×3 kernel around the point. The reason for varying the σ was to simulate that the shell around the limb would appear smaller at a greater distance. The matrix was then thresholded by the magnitude of each vector to generate a cleaner output.

3.4 Batch Loading

The data were loaded in batches...

Chapter 4

Human 3D pose from depth images

To train any network using supervised learning, large amounts of training data is needed. One of the goals for the MECS project is to do Human Activity Recognition (HAR), with the purpose of tracking a user from day to day, and look for patterns that could lead to worsening or more dangerous living conditions. The project also aims to be able to recognize the activity from any viewpoint, as the robot should be mobile to stay out of the way of the user as much as possible. A 2D approach to activity recognition will lack robustness or require training data from many different angles to complete this goal. A 3D approach will on the other hand provide robustness by not being dependent on the viewing angle.

This thesis focuses on describing a method for extracting human pose in 3D. By applying methods normally used on 3-channel (RGB) images to depth images, it is demonstrated that

As in [3], two networks are used to create the PAFs and the confidence maps for the joints. However, instead of training on 3-channel RGB images, we will use a single channel depth image to discover the body landmarks/joints. However, since depth images are single channel, and thus have less information than the RGB images, we propose using a shallower network. This also means we have to do the first step of feature extraction which was already done in a However, since the depth images are less detailed than normal RGB images, some landmarks might be harder to detect: eyes, nose, or placing the joint on an outstretched limb.

This was considered when preparing the training data.

TODO> about not needing a full 3D body mesh, and privacy concerns

4.1 Architecture

TODO> about shallowness of network

The architecture of this project is *recurrent* in that it repeats itself for a number of iterations. As with the architecture in [3], we have to have a first step which produces the first outputs we can use in later steps. However, this step is not illustrated in Fig.4.1, since the first step will be identical to the next steps, except it will not have the additional inputs produced by the outputs of the previous step.

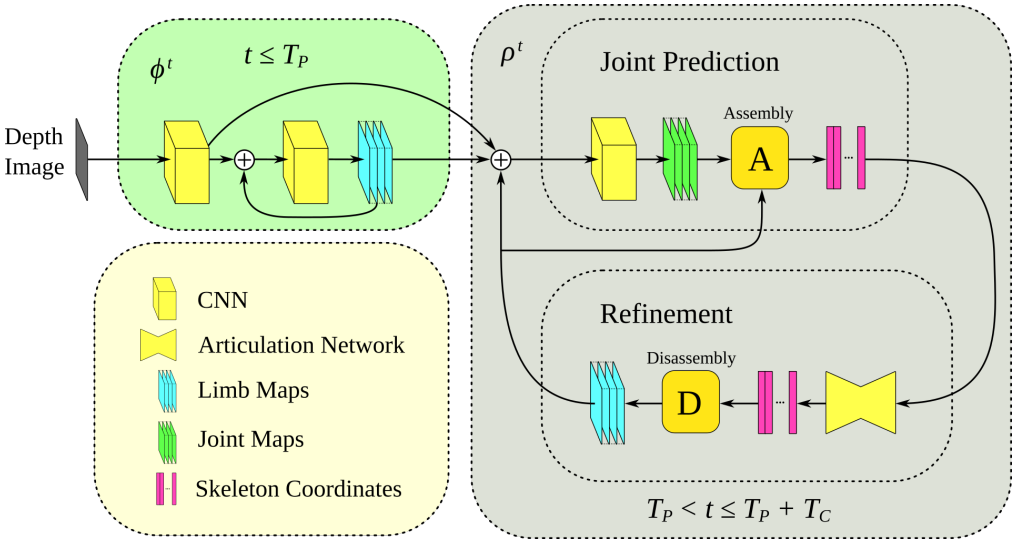


Figure 4.1: Main architecture, as in [3] two recurrent stages, ϕ^t and ρ^t are used to iteratively refine the limb and joint locations.

4.1.1 Depth feature extraction

For each pixel in the current layer of the CNN, we collect information from a filter-sized portion of the previous layer. This means that deeper layers look at a larger and larger portion of the input layer. This is useful for detecting connections between large-scale structures. This also means that after a certain depth, there may not be any more useful information.

In our experiments, we will try different depths for feature extraction.

Some of these features might be desirable as inputs for later layers in object classification.

This network is built from the ground up. Therefore we want some layers to, for example, detect edges and one for detecting slanting gradients or connected surfaces. For limbs, we might want to find surfaces that are shaped like tubes or oblong spheroids.

4.1.2 3D object detection

In this part of the network, we borrow some of the architecture described in [3]. The purpose is to find joints and limbs and put them into PAFs or joint confidence maps.

4.1.3 Articulation network

The articulation network is stacked on top of the part-detection network and its main role is to refine the limb lengths and angles between each joint. Each of the detected persons are passed through the articulation network, which leads to a bit more complexity and runtime for the network based on the number of people. However, since the network has so comparatively few inputs, and is quite shallow, preformance is not expected to suffer notably.

The coordinates and confidences for each joint (if not detected, confidence is 0) and the mean PAF vector for each limb is the input to the network.

The network will try to find out what the positions of joints with low confidences, or no detections, should be. It is hypothesized that this network will learn things like symmetry (left and right limbs should have the same length), proportionality (limbs should be proportional to each other), possible articulations, and natural poses.

For joints that isn't detected or where the confidence is very low, the network will input some standard coordinates for that joint, scaled by the limbs we already have the strongest confidences for. The standard scaling/coordinates are hard-coded, based on the human anatomy surveys in [14]. The exception is eyes and ears, which is set to the best guess. In addition, the depth coordinate for each joint is set to 0. Numbering and a visualization of the skeleton can be seen in Figure 4.2.

The architecture is visualized as a simple, fully connected neural

network. Though, it might be enough to connect the neurons responsible for connected limbs. A neuron in the second layer connected to the foot, knee and one of the hip-joints does not need to be connected to the inputs from a hand or shoulder. Subsequent hidden layers can however, be fully connected.

If we had some input describing the direction of the camera, and the visual hull containing the undetected points, this network may perform better, because the position of the limb would be constrained to the visual hull. However, that would require a fundamental change to the network, which is not done in this work.

This network will also be trained separately, as all it needs is poses and random confidences as inputs.

4.2 Training data preparation

Skeleton models

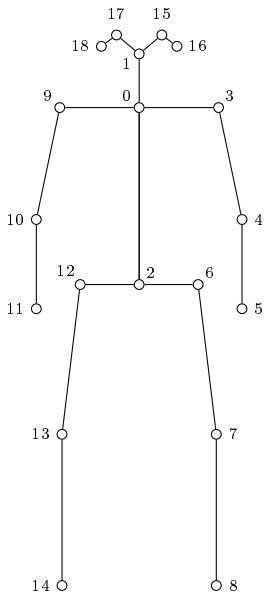


Figure 4.2: Numbering for detected landmarks/keypoint markers.

ID	Description	Std.Coord.
0	Neck	(0.00, 2.34)
1	Nose	(0.00, 3.05)
2	Middle hip	(0.00, 0.00)
3	Left shoulder	(1.05, 2.34)
4	Left elbow	(1.36, 0.86)
5	Left wrist	(1.36, -0.32)
6	Left hip	(0.78, 0.00)
7	Left knee	(1.02, -1.98)
8	Left ankle	(1.02, -3.98)
9	Right shoulder	(-1.05, 2.34)
10	Right elbow	(-1.36, 0.86)
11	Right wrist	(-1.36, -0.32)
12	Right hip	(-0.78, 0.00)
13	Right knee	(-1.02, -1.98)
14	Right ankle	(-1.02, -3.98)
15	Left eye	(0.30, 3.30)
16	Left ear	(0.50, 3.15)
17	Right eye	(-0.30, 3.30)
18	Right ear	(-0.50, 3.15)

Table 4.1: Numberings, names/descriptions and standard coordinates for recognized landmarks

4.3 3D encoding

The 3D scene is stored in a 2D depth map. That means that there is no need for a bounding box that creates constraints on how far away any detections can happen. The drawback being that any objects placed behind each other in the scene will suffer from being occluded.

Because of the 2D space, occurrences of distance information is limited to 1 per pixel in the 2D space. So, if two objects we want to represent are in line with each other (one occludes the other), there is no way to retain information about/distances to the two objects in the 2D space. This leads to the question of how two inline (along the z-axis) joints can be represented for a 3D part affinity field. Even though for this use case, the only scenario is that two of the same limb is co-linear (i.e., two persons are standing right behind each other). This means that when parts are connected into skeletons, there will be a chance for a limb to be reused between two people.

TODO> explain targets for the CNN (limb- and joint-maps)

Chapter 5

Experiments

5.1 Results

Chapter 6

Conclusions

Chapter 7

Future Work

Finding more accurate pose from a temporal algorithm which takes input from a series of depth images, from a single viewpoint.

Finding more accurate pose from multiple viewpoints.

Heart/respiration rate monitoring using frequency search in changing rgb and depth-pixel values for automatically selected RoIs.

Mood detection on facial expressions.

Human activity recognition using 3D pose provided by the method proposed in this paper.

Train the network over a larger dataset in unstructured environments and with multiple people present.

Train an accompanying network that takes a sequence of estimated limb positions and their probability as input, and trying to refine the estimation based on earlier detection. This could also be done through a Kalman filter.

This should all accumulate in an LSTM network for predicting diseases. – requires dataset aquired over possibly years, dispersed over many users, and their daily activities, as possible. Other factors that should be taken into consideration is environmental factors such as humidity, temprature and weather. (As they may be risk factors for certain conditions such as heatstroke or depression.) With such a diverse dataset we could possibly do PCA to determine certain risk factors for different diseases.

Train and test network on the Human 3.6M dataset using TOF data

Appendices

A Depth sensors

Since depth sensors are widely used in different robotics applications for tasks such as SLAM, odometry and object detection, we selected this as our main source of information for monitoring the user. There are mainly three different technologies to choose from: *Structured light*, *Time-of-Flight (ToF)* and *Stereo vision*.

A.1 Stereo vision

uses two cameras that are observing parts of the same scene. In commercial packages the cameras are usually calibrated, so we have measurements to put into the camera matrix as well as the rotation and translation between the two camera matrices.

However, to get a 3D structure, we need to find common feature points between the two cameras. To do this, we can use various feature descriptors such as ORB, SWIFT and SURF. When good matches has been found between the images, we measure the disparity between the points, and triangulate the distance. The depth measurements for the rest of the image are then calculated by matching pixels close to the found featurepoints.

Since this is an optically based technology, it will work well in well-lit scenes that contain many unique featurepoints. If we operate in an homogenous environment with few, or similar textures it will be difficult to find featurepoints to map the environment. An example of this could be on the seabed or inside buildings with limited light conditions, for example during a blackout.

A.2 Structured light

uses a projected pattern of light points onto the scene which is registered by a calibrated camera. Usually, the projected light pattern and camera operate in the infrared part of the electromagnetic spectrum[1]. This means that in locations where one can expect a lot of IR radiation, this technology will not work very well. Since the IR radiation from the sun usually is much stronger than the one emitted from the projector on the sensor, this technology will not work well outside in well-lit conditions. It will however work inside and in conditions where no external light source are provided.

[1] The Microsoft Kinect V2 sensor uses a wavelength of 827-850nm, according to [5, Chapter 4.1]

In addition, since the light is structured and the sensor is calibrated, we can skip the step where we find common featurepoints to triangulate the distance which we have to do in the stereo vision case.

A.3 Time-of-Flight

cameras uses the known constant of c to calculate distances in the image, by measuring the time a light-pulse emitted from the camera uses to be reflected onto the camera sensor. For example, Microsofts Kinect v2 uses a specialized ToF-pixel array in conjunction with a timing generator and modulated laser diodes to obtain per-pixel depth images [16].

As with structured light sensors, this is susceptible to interferrence from external light sources, or specular surfaces, and has limited range because of light fall-off. However, since the distance calculations are timing based, we can obtain framerates up to 30 fps in the Kinect v2 sensor [11].

B Robotic Operating System

In order to make the system easier to use and available to as many platforms as possible, it was decided to create it for the Robotic Operating System (ROS). ROS is a collection of libraries and a runtime environment making communication with different modules and programs on the robot possible.

C Classifiers

write a bit about what classifiers are, and how we use them to find the different keypoints in the image.

Glossary

CNN A type of neural network that uses convolved filters to create spacial recognition more robust.

visual hull The 3D geometric volume occluded by a foreground object. The visual hull of an object is the 3D geometric volume produced behind the object when .

Abbreviations

HAR Human Activity Recognition.

HRI Human-Robot Interaction.

MECS Multimodal Elderly Care System.

mocap Motion Capture.

PAF Part Affinity Field.

ToF Time-of-Flight.

Symbols

c Speed of Light.

Bibliography

[1] Mykhaylo Andriluka et al. '2D Human Pose Estimation: New Benchmark and State of the Art Analysis'. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2014.

[2] H. Bay, T. Tuytelaars and L. Van Gool. 'SURF: Speeded Up Robust Features'. In: *Computer Vision – ECCV 2006*. Ed. by A. Leonardis, H. Bischof and A. Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417. ISBN: 978-3-540-33833-8.

[3] Zhe Cao et al. 'Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields'. In: *CVPR*. 2017.

[4] V. A. Epanechnikov. 'Non-Parametric Estimation of a Multivariate Probability Density'. In: *Theory of Probability & Its Applications* 14.1 (1969), pp. 153–158. DOI: 10.1137/1114019. eprint: https://doi.org/10.1137/1114019. URL: https://doi.org/10.1137/1114019.

[5] Silvio Giancola, Matteo Valenti and Remo Sala. *A Survey on 3D Cameras: Metrological Comparison of Time-of-Flight, Structured-Light and Active Stereoscopy Technologies*. Springer International Publishing, 2018. DOI: 10.1007/978-3-319-91761-0. URL: http://dx.doi.org/10.1007/978-3-319-91761-0.

[6] P. V. C. Hough. 'Method and means for recognizing complex patterns'. US 3 069 654A. 1960. URL: https://patents.google.com/patent/US3069654A/en.

[7] Catalin Ionescu et al. 'Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.7 (July 2014), pp. 1325–1339.

[8] Hanbyul Joo et al. 'Panoptic Studio: A Massively Multiview System for Social Interaction Capture'. In: 2017.

[9] Hanbyul Joo et al. 'Panoptic Studio: A Massively Multiview System for Social Motion Capture'. In: *ICCV*. 2015.

[10] Helseetaten Oslo Kommune. *Oslohelsa – Kortversjonen, Oversikt over helsetilstand og påvirkningsfaktorer*. June 2016. URL: https://www.oslo.kommune.no/getfile.php/13139280/Innhold/Politikk%20og%20administrasjon/Statistikk/Oslohelsa_kortversjon.pdf.

[11] Elise Lachat et al. 'Assessment and Calibration of a RGB-D Camera (Kinect v2 Sensor) Towards a Potential Use for Close-Range 3D Modeling'. In: *Remote Sensing* 7.10 (Oct. 2015), pp. 13070–13097. ISSN: 2072-4292. DOI: 10.3390/rs71013070. URL: http://dx.doi.org/10.3390/rs71013070.

[12] D. G. Lowe. 'Object recognition from local scale-invariant features'. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. 1999, 1150–1157 vol.2.

[13] N. Magnenat-Thalmann, R. Laperrière and D. Thalmann. 'Joint-Dependent Local Deformations for Hand Animation and Object Grasping'. In: *Proceedings on Graphics Interface '88*. Edmonton, Alberta, Canada: Canadian Information Processing Society, 1989, pp. 26–33.

[14] Drillis R. and Contini R. *Body Segment Parameters*. 1966. URL: http://edge.rit.edu/edge/P13032/public/FinalDocuments/Detailed%20Analysis/Anthropometric%20Data/Drillis%20%26%20Contini.pdf.

[15] E. Rublee et al. 'ORB: An efficient alternative to SIFT or SURF'. In: *2011 International Conference on Computer Vision*. 2011, pp. 2564–2571.

[16] John Sell and Pat O'Connor. 'XBOX One Silicon'. Hot Chips 25. Aug. 2013. URL: http://www.hotchips.org/wp-content/uploads/hc_archives/hc25/HC25.10-SoC1-epub/HC25.26.121-fixed-%20XB1%2020130826gnn.pdf.

[17] Xinhang Song, Luis Herranz and Shuqiang Jiang. *Depth CNNs for RGB-D scene recognition: learning from scratch better than transferring from RGB-CNNs*. 2018. arXiv: 1801.06797 [cs.CV].

[18] O. Tange. 'GNU Parallel - The Command-Line Power Tool'. In: *;login: The USENIX Magazine* 36.1 (Feb. 2011), pp. 42–47. DOI: http://dx.doi.org/10.5281/zenodo.16303. URL: http://www.gnu.org/s/parallel.

[19] S. Ullman and S. Brenner. 'The interpretation of structure from motion'. In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 203.1153 (1979), pp. 405–426. DOI: 10.1098/rspb.1979.0006. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rspb.1979.0006. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspb.1979.0006.

[20] P. Viola and M. Jones. 'Rapid object detection using a boosted cascade of simple features'. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001, pp. I–I.