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 2020

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

4th May 2020

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 physical personnel, 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

Motion capture is expensive. However, being able to estimate human pose could have many application areas: analyzing movements, activity recognition, importing natural motions to Viritual Reality (VR), and improving Human Robot Interaction (HRI). The industry standard is to use an elaborate motion-capturing studio that requires multiple expensive cameras, a large area, and specialized software. Therefore the application areas for motion capture are currently mostly limited to staged research, movie-, and video game-making.

The Motion capture problem sets out to find a representation of an actor, that can be used in animation. This representation is often a *rigged* skeleton with bones that define the movement of the animated character [9].

This work explores both human pose detection in the depth domain, making the methods used here applicable to a wide range of sensors, and it explores a novel articulation network that refines the detection of the human pose.

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 [5] 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 [8], SURF [2] and ORB [11] 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 [13].

Hand-crafted features have also been used in machine learning scenarios. As an example, Haar-like features were demonstrated to efficiently find faces in [14]. 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. 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*.

CNNs are widely used in image classification tasks because they look at a col-

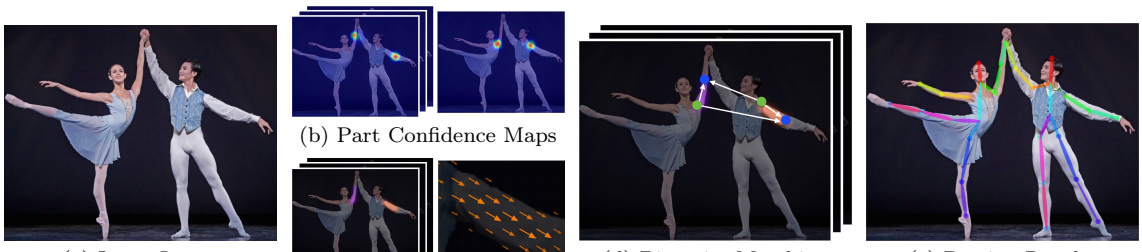
lection of spatially connected pixels. This means CNNs are robust to the placement of the object in the frame, and even partial occlusions.

2.2 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, 6].

There have been two ways of finding human pose in an image. One is a top-down approach, where a number of persons first are found in the image, and advanced algorithms have refined their pose after finding key features. The other is to find the key features for the whole image, then build up the instances of people.

In [3], two *cnn*s are used 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.



(a) Input Image (b) Part Confidence Maps (c) Part Affinity Fields (d) Bipartite Matching (e) Parsing Results

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 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

[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*.

happens because of the bipartite matching, which does not work if any joint is missing.

Chapter 3

Human 3D pose from depth images

To train any network using supervised learning, we need large amounts of training data. One of the goals for the Multimodal Elderly Care System (MECS) project is to do Human Activity Recognition (HAR), so one can track the user from day to day and look for patterns that could lead to worsening living conditions. We also want to be able to recognize the activity from any viewpoint, and this is where a 2D approach will lack robustness. This is because any HAR model trained solely on 2D data will only be able to recognize the activity from the views it has seen the activity being performed. A 3D approach will provide us with robustness for the viewing angle.

We implemented an algorithm for extracting human pose in 3D. Applying methods used on 3-channel (RGB) images to depth images, we show that the same methods can be used to extract objects in 2d images, can be used to extract objects in-depth images as well, when it comes to human pose.

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.

3.1 Architecture

When we are creating a neural network, it is often helpful to have in mind *what* we want to detect in each layer. It is, therefore, segregated into a couple of different

steps to make it easier to follow along.

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.3.1, 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.

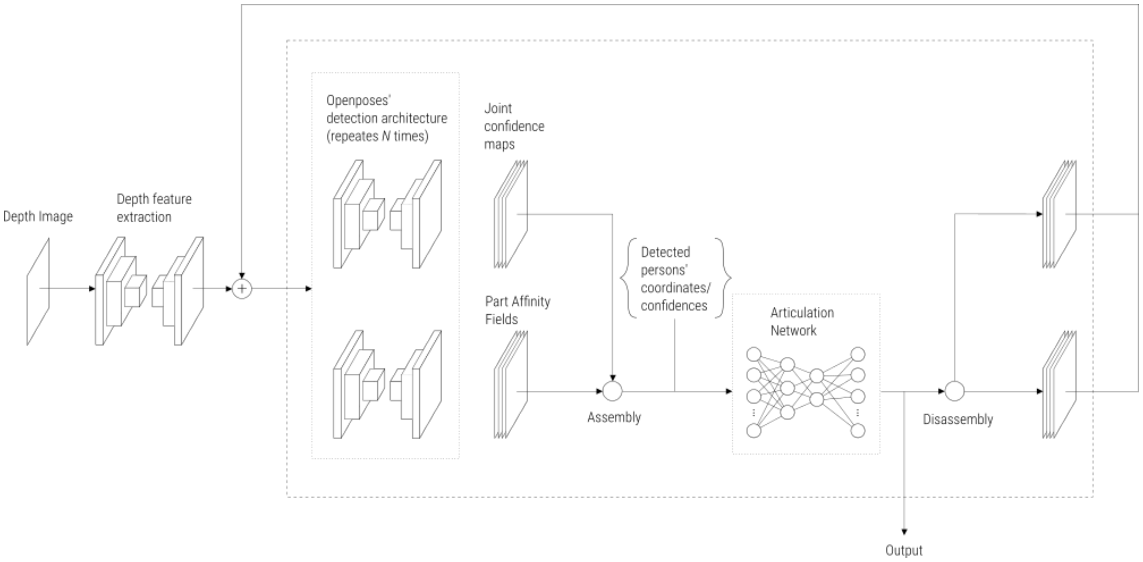


Figure 3.1: Main architecture. CNNs are illustrated as hourglass networks, but this is not necessarily representative of the final architecture. A depth image is fed to a depth-feature extraction network (3.1.1), which in consecutive steps are combined with output PAFs and joint confidence maps before this is fed to the 3D object detection networks (3.1.2). These are run for a small number of iterations to get some estimate for the PAFs and confidence maps, before we move on to the assembly step. For each of the skeletons detected (with a ceiling for maximum detected skeletons) the skeletons with the strongest evidence are assembled. In the case that a body landmark is not found, we use preconceived coordinates from a standard defined skeleton. The coordinate frame of the skeleton in the articulation step till be the root joint (middle hip) with the z-axis pointing up, and the y-axis orthogonal to the line through the two side hip joints. The articulation network (3.1.3) tries to scale and articulate the skeleton based on detected evidence. In the disassembly step, we place all the skeletons back into the camera coordinate frames, and update the PAFs and confidence maps with confidences and placements from the articulation network, and use them in the following steps. This is repeated a small number of times (3-4).

3.1.1 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.

3.1.2 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.

3.1.3 Articulation network

Input: Coordinates and confidences for each joint (if not detected, confidence is 0) and the mean PAF vector for each limb. Here, we get some coordinates for different joints, along with the confidences for those coordinates. The network will try to find out what it thinks the 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 we have not 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 [10]. 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 3.2.

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. 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.

3.2 Training data preparation

Skeleton models

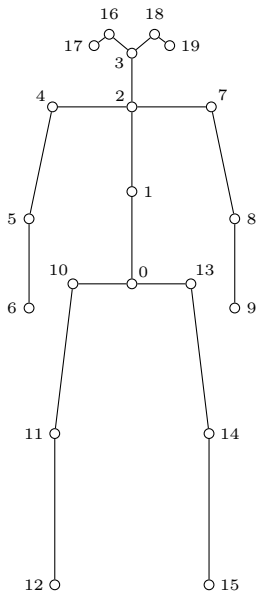


Figure 3.2: Numbering for detected landmarks/keypoint markers.

ID	Description	Std.Coord.
0	Middle hip	(0.00, 0.00)
1	Middle back	(0.00, 1.22)
2	Neck	(0.00, 2.34)
3	Nose	(0.00, 3.05)
4	Right shoulder	(-1.05, 2.34)
5	Right elbow	(-1.36, 0.86)
6	Right wrist	(-1.36, -0.32)
7	Left shoulder	(1.05, 2.34)
8	Left elbow	(1.36, 0.86)
9	Left wrist	(1.36, -0.32)
10	Right hip	(-0.78, 0.00)
11	Right knee	(-1.02, -1.98)
12	Right ankle	(-1.02, -3.98)
13	Left hip	(0.78, 0.00)
14	Left knee	(1.02, -1.98)
15	Left ankle	(1.02, -3.98)
16	Right eye	(-0.30, 3.30)
17	Right ear	(-0.50, 3.15)
18	Left eye	(0.30, 3.30)
19	Left ear	(0.50, 3.15)

Table 3.1: Numberings, names/descriptions and standard coordinates for recognized landmarks

Chapter 4

Experiments

Chapter 5

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

Chapter 6

Conclusions

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.

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.

[1] The Microsoft Kinect V2 sensor uses a wavelength of 827-850nm, according to [4, Chapter 4.1]

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 [12].

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 [7].

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.

PAF Part Affinity Field.

ToF Time-of-Flight.

VR Viritual Reality.

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] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.