Automatic analyzing humans in photographs or videos has great potential applications in computer vision, including medical diagnosis, sports, entertainment, movie editing and surveillance, just to name a few. Body, face and hand are the most studied components of humans. Body has many variabilities in shape and clothing along with high degrees of freedom in pose. Face has many muscles causing many visible deformity, beside variable shape and hair style. Hand is a small object, moving fast and has high degrees of freedom. Adding human characteristics to all aforementioned variabilities makes human analysis quite a challenging task.

In this thesis, we developed human segmentation in different modalities. In a first scenario, we segmented human body and hand in depth images using example-based shape warping. We developed a shape descriptor based on shape context and class probabilities of shape regions to extract nearest neighbors. We then considered rigid affine alignment vs. nonrigid iterative shape warping. In a second scenario, we segmented face in RGB images using convolutional neural networks (CNN). We modeled conditional random field with recurrent neural networks. In our model pair-wise kernels are not fixed and learned during training. We trained the network end-to-end using adversarial networks which improved hair segmentation by a high margin.

We also worked on 3D hand pose estimation in depth images. In a generative approach, we fitted a finger model separately for each finger based on our example-based rigid hand segmentation. We minimized an energy function based on overlapping area, depth discrepancy and finger collisions. We also applied linear models in joint trajectory space to refine occluded joints based on visible joints error and invisible joints trajectory smoothness. In a CNN-based approach, we developed a tree-structure network to train specific features for each finger and fused them for global pose consistency. We also formulated physical and appearance constraints as loss functions.

Finally, we developed a number of applications consisting of human soft biometrics measurement and garment retexturing. We also generated some datasets in this thesis consisting of human segmentation, synthetic hand pose, garment retexturing and Italian gestures.

Meysam Madadi

Human Segmentation, Pose Estimation and Applications

Human Segmentation, Pose Estimation and Applications

Meysam Madadi

CVC

Universitat Autònoma de Barcelona

CVC
Centre de Visió per Computador

Human Segmentation, Pose Estimation and Applications

A dissertation submitted by **Meysam Madadi** at Universitat Autònoma de Barcelonato fulfil the degree of **Doctor of Philosophy**.

Bellaterra, September 25, 2017

Co-Director | **Dr. Sergio Escalera**
Department of Mathematics and Informatics
Universitat de Barcelona

Co-Director | **Dr. Jordi Gonzàlez**
Dept. Ciències de la computació & Centre de Visió per Computador
Universitat Autònoma de Barcelona

Co-Director | **Dr. Xavier Baró**
Universitat Oberta de Catalunya

Thesis committee | **Dr. Manuel Jesus Marin-Jimenez**
Departament de Informática y Análisis Numérico
Universidad de Córdoba

Dr. Laura Igual Muñoz
Departament Matemàtiques i Informàtica
Universitat de Barcelona

Dr. David Masip Rodo
Departament Informàtica Multimedia i Telecomunicacions
Universitat Oberta de Catalunya

CVC
Centre de Visió per Computador

This document was typeset by the author using LaTeX 2$_\varepsilon$.

Live as if you were to die tomorrow.
Learn as if you were to live forever.
— Mahatma Gandhi

To Bahar, my family and friends . . .

Acknowledgments

I would like to express my deep and sincere gratitude to my supervisors professors Sergio Escalera, Jordi Gonzàlez and Xavier Baró. I appreciate all their contributions of time, ideas and encouragements during my PhD study. It has been an honor for me to be their PhD student. I am thankful of their patience and support at every stage of this PhD work. I am very grateful for the opportunity that was given to me by group of Human Pose and Behavior Analysis (HUPBA) to carry out this work as well as for the use of facilities.

I must also thank administration staff of CVC for all the help and sympathy provided over the years. I also thank Montse and Gisele for their help. I would like to thank anybody who helped me to finish this career and my friends at CVC. I give a special thank to Generalitat de Catalunya for supporting me in this thesis.

I would like to express my thanks to my family for their discrete but ever so precious moral support. Finally, I would like to give my special thanks to Bahar, my wife, for all her help and patience.

i

Abstract

Automatic analyzing humans in photographs or videos has great potential applications in computer vision, including medical diagnosis, sports, entertainment, movie editing and surveillance, just to name a few. Body, face and hand are the most studied components of humans. Body has many variabilities in shape and clothing along with high degrees of freedom in pose. Face has many muscles causing many visible deformity, beside variable shape and hair style. Hand is a small object, moving fast and has high degrees of freedom. Adding human characteristics to all aforementioned variabilities makes human analysis quite a challenging task.

In this thesis, we developed human segmentation in different modalities. In a first scenario, we segmented human body and hand in depth images using example-based shape warping. We developed a shape descriptor based on shape context and class probabilities of shape regions to extract nearest neighbors. We then considered rigid affine alignment vs. nonrigid iterative shape warping. In a second scenario, we segmented face in RGB images using convolutional neural networks (CNN). We modeled conditional random field with recurrent neural networks. In our model pair-wise kernels are not fixed and learned during training. We trained the network end-to-end using adversarial networks which improved hair segmentation by a high margin.

We also worked on 3D hand pose estimation in depth images. In a generative approach, we fitted a finger model separately for each finger based on our example-based rigid hand segmentation. We minimized an energy function based on overlapping area, depth discrepancy and finger collisions. We also applied linear models in joint trajectory space to refine occluded joints based on visible joints error and invisible joints trajectory smoothness. In a CNN-based approach, we developed a tree-structure network to train specific features for each finger and fused them for global pose consistency. We also formulated physical and appearance constraints as loss functions.

Finally, we developed a number of applications consisting of human soft biometrics measurement and garment retexturing. We also generated some datasets in this thesis consisting of human segmentation, synthetic hand pose, garment retexturing and Italian gestures.

Key words: *depth image, segmentation, pose recovery, CNN, generative model*

Contents

List of Figures

List of Tables

1 Introduction

Computer vision has attracted a lot of interest during last decades because of the growth of captured data, and the increasing demanding from industry for automatic processing, understanding and managing these data. This is while new sensors have been introduced (*e.g.* hand-held camera, smart-phone, Kinect, Google Glass, GoPro), image and video repositories appeared and many applications to process them were required. Therefore, the final goal of computer vision is to solve low level problems like image denoising and resolution enhancement, or high level problems like object/event segmentation/detection/classification and image/video/text generation. Therefore, developing effective and efficient techniques is important, and significant advances in the last decade, thanks to the growth of datasets and computer powers (*e.g.* GPUs), tend to the development of real world applications.

Despite recent advances, still some challenges exist. Feature extraction and image representation are fundamental tasks in computer vision. Correctly describing images with respect to the variability and noise in the data has been vastly studied. Such descriptors must be invariant to some transformations (*e.g.* rotation or translation), illumination and noise (*e.g.* image distortions or missing data), while being able to model inter/intra-class data correlations or even temporal connectivity. This is where machine learning algorithms comes in and play an important role in the success of models. While many hand crafted features were proposed in the literature (like HOG and SIFT or more advanced representations like bag-of-words), recently convolutional neural networks (CNN) achieved a great success by learning simple filters in a complicate feed-forward graph. However, CNNs are not invariant to aforementioned variabilities and such variabilities must be embedded in the data.

Among all captured images and videos in the world, humans are present in most of them and correctly analyzing humans in the data is an important issue in many applications. Face, hand and body are the most studied components of humans. Early works in facial domain tried to solve face detection and localization in the images while face recognition remained a challenging problem till recent years. Psychological face analysis tried to solve problems like emotion and first impression

recovery. In the movie and multimedia industry, facial landmark detection could be used for face alignment and deformation (*e.g.* avatar generation). Face segmentation, as a mid level computer vision problem, used to help some higher order applications like make-up transform. Hand pose recovery and gesture recognition are fundamental tasks in human-computer interaction and robot learning. Recently, the number of publications has been increased in this domain by introducing depth sensors (*e.g.* Kinect) which helped to overcome some traditional challenges like hand self-similarity and illumination changes. Human identification and behavior analysis in surveillance systems, body deformation in movie industry, body gesture recognition in human-computer interaction and body movements in sport analysis are examples of applications where analyzing and tracking human body is necessary. This can not be done without some intermediate tasks like human segmentation, body pose recovery or biometrics extraction.

Problem statement: In this thesis we address the problem of human segmentation and pose recovery in different modalities and develop a number of applications in both RGB and depth images (captured indoor). Face segments can be defined as hair, face skin and face components like eyes, nose and lips. For hand, segments could be finger phalanxes and palm. Body segments, however, has a broad definition and is application dependent. In one application body limbs can be defined as segments while in another clothes are composing the segments. However, in this thesis we do not cover cloth segmentation. In this study, human segments has a high correlation with human pose. Human pose is determined as joints locations in 2D/3D and bones connect them together. For face, This definition is referred as landmark localization and we do not cover it in this thesis.

In this thesis, we provide different solutions in a single camera single object setup and heavily rely on machine learning techniques. While example/model deformation is used for hand/body segmentation in depth images, CNNs are applied in face segmentation in RGB modality. In 3D hand pose recovery in depth images, we propose two solutions: 1- realistic hand model fitting and temporal pose refinement, and 2- CNN based regression. Finally, we develop some applications based on human segmentation like geometrical soft biometry extraction and garment retexturing.

1.1 Challenges

RGB images: Other than common challenges in RGB images like illumination changes and background clutter, we list some specific challenges in the following.

- In the **face segmentation**, hair segmentation is the most challenging task. Hair has a free-form style with different colors and easily can be confused

with background color. In single object segmentation, background faces near to front face can be easily segmented as frontal face. Also, face can be occluded by some objects like hair, glasses or even hand.

- **Single frame 3D body pose recovery:** Human body has a high degree of freedom and enormous number of poses can be defined by a human being. Human shape variability and different cloths styles and colors add even more difficulty to the problem. Humans can be easily occluded partially by other objects or parts of the body (self-occlusion). In single frame single camera setup, depth information is lost and there are ambiguities in the orientations.

  Usually, the search space is constrained to 20 to 60 joint angles that determine the pose configuration through a skeleton model. Even with this rather coarse representation the search space is very high dimensional. Therefore, efficient optimization and search strategies are needed. A further complication is that only a few regions of parameter space correspond to valid human poses.

  A true generative method should model all the intricacies of the image formation process. Obviously the high appearance variability makes it difficult to model. In one hand, a very detailed model might work best for a specific instance but might not generalize well to other scenes. On the other hand, a very coarse model will not truthfully synthesize the image.

- Generally, **garment retexturing** is a challenging task. Firstly, as a pre-processing, garment must be segmented from background and other objects which is still an open problem, despite recent advances. Secondly, the source and target garments may not have the same texture/pattern or shape and this makes finding correspondences really hard. These problems get harder when we must solve it for a moving subject with different poses and perhaps self-occlusions. In automatic and virtual clothing, different from static garment retexturing, dynamics of the garment must be solved as well.

Depth images: although, in general, depth sensors have helped to cope with some challenges like illumination changes in RGB data, some challenges still remain open. Depth sensors are mainly applicable in indoor scenarios and provide from 0.5 to 4 meters effective depth, suffering from missing data and/or boundary noise.

- **3D hand pose recovery:** hand is a small object which is moving fast, easily getting occluded and has a high self similarity. Lack of texture in depth data along with blurry image (due to fast motion) makes it hard to find correct solution without having temporal information or body state. Hand, similar to body, has a high degree of freedom which needs efficient search strategies.

1.2 Related works

Human pose and segmentation have a wide range of applications and a lot of attention has been paid to these domains during the last decades. In this section we briefly review a number of works and ideas in the state of the art. Generally, we can group any approach into generative and discriminative or a combination of both. Generative approaches aim at modeling all the complications of the image formation process. A truly generative method should model variabilities in the object like human pose, shape, clothing, illumination and even background. In practice, generative methods only model image features with respect to easiness and efficiency of building and synthesizing. Inference in generative methods usually is done by optimizing a defined energy function w.r.t. to the model parameters. The energy measures the consistency between the model and the observations where the hope is that the global optimum corresponds to the true solution. A successful generative model must be able to generalize to different motion patterns and appearance conditions. However, for efficient and feasible evaluation of the energy in huge search space, some initialization must be provided.

Discriminative methods learn a model using training data to map image features to the problem space. In contrast to generative methods, discriminative models are more direct and are therefore easier to implement. Such methods perform very well when proper feature extraction is applied and the distribution of the training is similar to the distribution of the testing. However, these methods often overfit on training data and generalization to instances unseen in the training data is a challenge. Furthermore, obtaining enough training data with accurate groundtruth is not a trivial task. Although each method has its advantages and shortcomings, they can complement each other. Discriminative methods can provide rough idea about the problem and generative methods can provide the fine detail.

1.2.1 Segmentation

Human parsing, segmentation and pose recovery can be treated as synonyms depending on the solution. Pictorial structures model an object with connections between parts and learn appearance and configuration of each part. It is common to keep configuration parameters small for efficient inference, which can contain part location, orientation and scale [47]. Hierarchical models, specifically AND/OR graphs, are kinds of body parts representation used to encode the hierarchical and reconfigurable composition of parts as well as the geometric and compatibility constraints between parts. AND/OR graph models are built upon parts, poselets or parselets features and, generally, structural learning and max-margin framework is used to learn the parameters of the model discriminatively [37, 187, 217].

Energy minimization based on CRF and graph-cut has been actively studied for segmentation problems where an initial unary potential is refined by pair-wise potentials defined among different parts. As an example, while [139] proposed an edge based deformable model and iterative parsing learned by CRF, [64] applied graph cuts to perform a local and spatial optimization as an energy minimization function to improve initial unary potentials. [89] proposed an efficient iterative algorithm for approximate inference in fully-connected CRFs with Gaussian edge potentials, which has been widely adopted for postprocessing outputs of pixel-level segmentation models [9, 66, 124].

Model based shape fitting is a successful solution for both segmentation and pose recovery. Statistical shape models, like PCA-based models, are useful tools to deform a template model and then fit it with shape appearance. Statistical part based models can be combined with graphical models to generate deformable models [219, 220]. Another kind of shape fitting can be applied by exemplar-based non-rigid warping where a template or nearest example is registered with shape [158, 201].

Shotton *et al.*[152] trained random forest (RF) on pixel features extracted from a depth image and showed a successful pixel-level discriminative human segmentation. However, the output segmentation was noisy and researchers proposed spatial connectivities among segments to improve the results [64, 82]. Recently, researchers have been actively working on CNNs, among discriminative approaches, and many advances have been achieved though in the segmentation problem. [107] combined fully-connected layers with (de)convolution layers and enabled denser [9, 66, 124], higher resolution predictions [49], and/or encoder-decoder [208] predictions. [105] generated pairwise terms as class edge potentials through a four-connected graph. Such edge potentials extracted in a multi-objective network along with unary terms were trained using non-structured loss functions.

1.2.2 Pose recovery

Generative models, from pictorial structures to more detailed shape generation like SMPL model [108], have been vastly used in human pose recovery. While pictorial structures and its varities [26, 47, 133] have been used for 2D pose recovery, [6, 13] combined pictorial structures from different view points for 3D pose recovery. Realistic models can be used for multiple purposes. [134] minimized a parametric model using body silhouette and fused video with inertial sensors for accurate motion capture (MoCap). [18] minimized SMPL model parameters based on 2D pose estimation and image appearance. Model fitting has been used for hand pose recovery in depth images as well. While [127] used a simple hand model to fit with appearance, [168] applied a more sophisticated fitting function and model where

the model can be personalized for each person. A set of approaches minimize a function over generated hypotheses. [155] generates a set of 3D hypotheses over Gaussian distributions of body parts based on 2D estimated part locations and finds the best candidate. [213] estimates 2D pose from 3D MoCap data over dense trajectories.

Discriminative approaches can be divided into heat-map based and regression techniques. Heat-map based techniques generate parts likelihood and pose is estimated through generative models like pictorial structures, inverse kinematics [172] or viewpoint fusion [106]. Heat maps are generated through separate classifiers trained on each part or unified classifiers like CNNs. CNNs are able to jointly model heat maps and graphical models [173], cascading models [189] or error feedback loops [21]. Regression techniques estimate pose directly in the output of the regressor and can perform better in case of occlusions [162, 165, 175].

1.2.3 Datasets

Available datasets are growing every day in any domain in computer vision with more complicated and larger data. In this subsection we review a number of datasets with respect to the scope of this thesis.

Semantic segmentation: There is enormous number of datasets in different domains. Urban scene parsing datasets [20, 147] are mainly used in automatic driver assistant systems and contains labels like road, car, building, tree and pedestrian. Indoor/outdoor scene parsing datasets [137, 211] contains more semantic labels under quite different light conditions and environments. Microsoft coco [102] recently is introduced with a large number of labeled images (300K) in more than 70 categories. NYU depth dataset [122] even includes depth images along with RGB data in the video sequences of indoor environment.

More specific to this thesis in human segmentation, Chen *et al.*[25] added part labels to pascal VOC 2010 dataset [120] which contains over 3K images of humans. In the human category 24 semantic parts are defined as hand, foot, lower/upper arm and leg, upper body and head and face parts. Freiburg Sitting people [128] constitutes a small dataset containing 201 images of six subjects in different viewpoints. Recently, [92] fitted SMPL model on LSP dataset [73] based on body joints and silhouette, and provided SMPL-based human segmentation on that dataset where 8515 images are given with part labels and body model. Other than body parts, [100] provides human body labels for clothes segmentation, though semantic labels are categorized based on cloth type. Recently, LIP dataset [50] introduced as cropped person instances from Microsoft coco constituting 50,462 images. Labels are defined as 19 human parts or clothes labels such as dress, socks, pants, gloves, scarf, skirt, and in addition to a background label.

To the best of our knowledge, there is not any dataset for human body segmentation in depth images with enough variability in pose and viewpoint. However, for hand segmentation in depth images, [150] captures RGB and depth images of five subjects applying gestures. To provide ground truth labels, fingers are painted and hand segments are extracted by color segmentation and manual correction. Moreover, recently Freiburg Rendered Handpose Dataset [218] has been introduced as a synthetic dataset with 41,258 training and 2,728 test RGBD images. Segmentation labels are defined as background, body and three per bone finger and palm segments.

Datasets in face segmentation have different semantics in labels. While in part labels dataset [76] segments are defined as face skin, hair and background, in a higher order semantic definition, Helen dataset [93] contain eyebrows, eyes, nose and mouth as well. Parts label dataset contains 2927 face images, a subset of LFW dataset [68]. In this dataset labeling is done manually based on extracted super-pixels. FASSEG dataset [83] is a small dataset containing 70 frontal and 200 multi-view segmented images. Face images are cropped in this dataset. Helen dataset provides labels based on facial landmarks on high resolution images under a broad range of appearance changes like illumination, age, emotion and pose. This dataset consists of 2000 training and 330 test images.

Hand pose recovery: Earlier datasets in this domain are small datasets with mainly front view hand deformation in RGB cameras. Most of recent datasets have been created by the aim of depth cameras. However, a number of datasets are introduced with both RGB and depth images. Datasets in this domain are provided with 3D joint locations. However, Dexter dataset [159] and HandNet dataset [192] just provide fingertips in the groundtruth. Dexter dataset was recorded using multiple RGB and depth cameras in multiple viewpoints. One single subject is performing 7 actions in less than 500 frames. HandNet dataset contains more data (totally 215K frames) recorded from 10 subjects using one single RGBD camera. Moreover, some datasets just assign a posture class to each image as groundtruth such as static or dynamic sign language datasets [7, 11]. Recently, egocentric datasets [10] are introduced with hand-hand and hand-object interactions.

ICVL [166], MSRA [162] and NYU [172] datasets are among the most cited datasets in the state of the art. ICVL has 180K training depth data with small noise in the images and 16 annotated joints. This dataset has a low variability in the pose and over 30% of ground truth annotations have low accuracy. MSRA have been recorded from 9 subjects in 17 pose categories (500 frames per each) and 21 annotated joints. NYU is recorded in three viewpoints simultaneously with three Kinect1 cameras. Training set contains over 72K RGBD frames gathered from one subject. Although, depth images are noisy with missing data, ground truth annotations are almost accurate due to multiple view model fitting. Recently, Big2.2Hand dataset

[203] is introduced with a high number of annotated depth frames (over 2.2M) from different viewpoints including egocentric images and even object interaction. 10 subjects participate in this dataset.

1.3 Contributions

We organize our contributions in three groups: human segmentation consisting of body, hand and face, 3D hand pose recovery, and applications and datasets.

1. **Human segmentation:** In depth images, we use nearest examples to segment body and hand. For this purpose, we applied shape descriptors like histogram of oriented gradients (HOG) and a new descriptor that encodes class probabilities of points into spatial bins. We then used iterative rigid/non-rigid alignment of shape point clouds and classified the points. We obtained accurate results even for small segments and showed benefits of example-based approaches against discriminative shape segmentation.

   In RGB images, we developed face segmentation based on CNNs. We used conditional random fields (CRF) modeled as recurrent neural network (RNN) layers and instead of using fixed pair-wise kernels, we let the network to learn it without training a separate unary potential. We combined the outputs with generative adversarial networks (GAN) and outperformed state of the art specially for hair segmentation by a high margin. We give an initial face segmentation as input to another channel of layers and gain some improvements in the results.

2. **3D human pose recovery:** For hand pose recovery in depth images, we developed two approaches. In both approaches, we showed separating the problem into simpler sub-problems makes it easier and faster to search the space and optimize the solution. More specifically, we separated viewpoint and palm joints optimization from fingers. In the first approach, we used a generative model benefiting from nearest example data which used to segment hand and extract palm joints. The accuracy of segmentation and extracted palm joints make us be able to optimize and fit our model fast and accurate separately for each finger. We also studied the usage of temporal data in occlusion recovery and could slightly improve occluded joints.

   In the second approach, we applied discriminative approaches by using CNNs. We separated fingers in different channels and fused the features at the end. We combined all channels into a tree-structure network which helped a better optimization than single channel networks and made us

8

able to have more finger specific features. We also applied appearance and physical hand constraints as new loss function which helped optimization by providing gradients in the back-propagation.

3. **Applications and datasets:** We developed a number of applications and datasets. In the first application, we provided body soft biometrics measurements based on segmented body. We used an aligned model to fill gaps and occluded areas. Finally, we extracted a set of adjacent points on a plane cutting body surface and measured limbs size by Euclidean distance among the points.

   We also developed a garment retexturing application where a segmented garment is retextured by a given garment image. The source image is in RGB and the target image in RGBD. The surface topology is extracted from depth image using geodesic distance. We used garments boundary correspondences as key points to deform target point cloud through thin plate spline (TPS) and fit on source garment. Finally, each point is assigned a color from source garment.

   In this thesis we have created a number of datasets: 1- human depth images with segments and limb sizes as ground truth, 2- a synthetic dataset of hand depth images with variable pose and viewpoint with segments and poses as ground truth, 3- an Italian gesture dataset with audio and RGBD data, and pose and gestures begin-end as ground truth, 4- a garment dataset with RGBD data and stitched markers as ground truth. Also some contributions have been done in a set of events, "Chalearn: Looking at people" challenges and workshops, and studied gesture recognition on Italian gestures and explainable computer vision on first impressions.

1.4 List of papers

Journal

- **Meysam Madadi**, Egils Avots, Gholamreza Anbarjafari, Sergio Escalera, Xavier Baro, Jordi Gonzalez, From 2D to 3D Geodesic-based Garment Matching: A Virtual Fitting Room Approach, Under revision at IET Computer Vision, 2017.

- **Meysam Madadi**, Sergio Escalera, Jordi Gonzàlez, F. Xavier Roca, Felipe Lumbreras, Multi-part body segmentation based on depth maps for soft biometry analysis, Pattern Recognition Letters 56 (2015), pp. 14–21

Conference proceedings

9

- **Meysam Madadi**, Sergio Escalera, Alex Carruesco Llorens, Carlos Andujar, Xavier Baro, Jordi Gonzalez, Occlusion aware hand pose recovery from sequences of depth images, 12th IEEE Conference on Automatic Face and Gesture Recognition (FG), 2017

- Hugo Jair Escalante, Isabelle Guyon, Sergio Escalera, Julio Jacques Jr., **Meysam Madadi**, Xavier Baro, Stephane Ayache, Evelyne Viegas, Yagmur Gucluturk, Umut Guclu, Marcel van Gerven, Rob van Lier. Design of an Explainable Machine Learning Challenge for Video Interviews. Proceedings of the The 2017 International Joint Conference on Neural Networks (IJCNN 2017), IEEE, 2017

- Sergio Escalera, Xavier Baro, Jordi Gonzalez, Miguel A. Bautista, **Meysam Madadi**, Miguel Reyes, Víctor Ponce, Hugo J. Escalante, Jamie Shotton, Isabelle Guyon, ChaLearn Looking at People Challenge 2014: Dataset and Results, ChaLearn Looking at People, European Conference on Computer Vision, 2014.

ArXiv

- **Meysam Madadi**, Sergio Escalera, Xavier Baro, Jordi Gonzalez, End-to-end Global to Local CNN Learning for Hand Pose Recovery in Depth data, arXiv: 1705.09606, 2017.

- Umut Guclu, Yagmur Gucluturk, **Meysam Madadi**, Sergio Escalera, Xavier Baro, Jordi Gonzalez, et al. End-to-end semantic face segmentation with conditional random fields as convolutional, recurrent and adversarial networks. Under revision at PAMI, 2017.

1.5   Outline

The organization of the thesis is as follows. In Chap. 2 we explain human segmentation by exemplar-based body and hand segmentation, and CNN-based face segmentation. 3D hand pose recovery is covered in Chap. 3 based on generative models and CNNs. In Chap. 4 we explain two applications and datasets we created during this thesis. Finally, conclusions and future work discussion are presented in Chap. 5.

2 Human Segmentation

2.1 Exemplar-based body segmentation in depth images

Human body segmentation is a mid level computer vision problem and used in many high level applications like biomedical image processing, clothes classification, clothes retexturing and virtual try-on, and biometrics extraction and surveillance, just to name a few. Additionally, utilization of body segmentation techniques could potentially improve the performance in several other computer vision tasks such as pose recovery or gesture recognition. Body segmentation is mainly a challenging task due to the variabilities in body pose, shape, clothes and environment. A body segmentation approach must be able to detect body boundaries and mask, and segment body into parts. Segmenting body from background can be applied as pre-processing (*e.g.* using background subtraction, depth thresholding, classification or energy-based approaches) or along with body parts. In this section, we assume a subject is segmented from the background beforehand. Fig. 2.1 shows segmentation models we used for body segmentation using depth images, consisting of hand as well.

Depth images, as the outputs of depth sensors (*e.g.* Kinect and Intel Realscene), are a kind of image with the distance of each visible point in the scene to the camera, and thus invariant to the color changes. Currently, most of depth sensors are mainly based on time of flight (ToF) for depth computation and applied indoors with a limited field of view (FoV) and range. Kinect is one of the first sensors with a broader FoV. Although, its first version was using light coding with an infrared projector and performed poorly with many missing data in IR interference, its second version has been replaced by ToF technology and kept a high FoV, resulting to a higher quality image. Regardless of generated image quality and level of captured details, a common problem with ToF cameras is generation of noise in objects boundaries. In some applications we use morphological operators to filter such noise.

Among segmentation approaches, [152] proposed a random forest based ap-
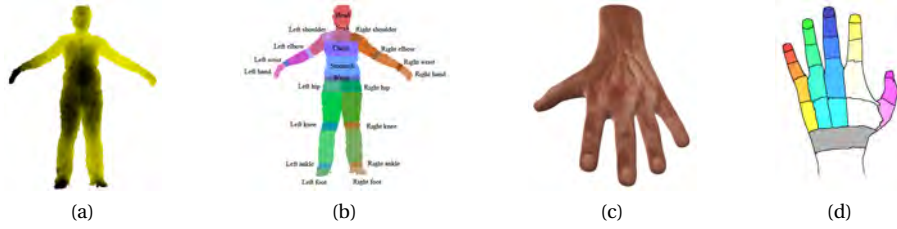
Figure 2.1: Body and hand models. Typical body depth image, hand model and defined segments.

proach to learn pixel labels from depth offsets, achieving robust segmentation results. This method has become one of the standard techniques for segmentation in depth data. However, this approach requires a huge dataset of real and synthetic labeled images as well as an expensive training procedure. Researchers have used different modifications of random forest such as hough forests [195] and hierarchical decision forests [82], or post processing energy minimization to improve segmentation accuracy. For instance, [65] applied graph cuts to perform a local and spatial optimization of random forest output probabilities.

Recently, CNNs have been actively used for the segmentation problem in general. However, there has been relatively fewer semantic body segmentation models relying on CNNs and depth images. [123] modeled structure learning in two sequential networks, one for pixel label and the other for context learning. They also modeled the prediction structure into the loss function using local prediction consistency and global segment variances. [128] refined fully convolutional network in [107] by adding more up-convolution layers and generating spatial dropout in transition layers coming from earlier pooling layers. [186] proposed a general architecture for jointly object and parts segmentation in multiple objects datasets. They grouped similar semantic parts among the objects and applied a two-stream network for object and part potentials which is followed by a fully connected CRF to explore long-range context.

Generative and realistic model fitting can be handled easier in depth than RGB images because of easier measuring of object surface geometries (*e.g.* surface normals) and shape topologies. Simple geometric models represent shape by connection of simple geometric objects (such as cylinders and spheres) in the kinematic tree [79, 127], while more sophisticated models represent shape point displacements by training parameters in reduced linear space (*e.g.* using PCA) [75, 108]. [219] used a part-based model and separated part parameters for part shape and

body pose. They stitched parts and fitted this model on 3D human data using graphical models. A number of works used examples, retrieved from a dataset [150] or previous frame estimation [127], to reduce search space and avoid model drifts. Authors in [201] used examples to register with shape points without keeping any special constraint on the connection and dependency of points in the example model. Therefore, the example/model fitting remained as a registration problem.

In this section, we consider exemplar-based segmentation in depth images using rigid and non-rigid 3D shape alignment and provide results for body and hand data. For this task, we develop a number of shape descriptors based on histograms of oriented gradients (HOG) [2] and shape context [15, 91] to extract example models. Rigid alignment does not change source shape topology and it can be defined as an affine transformation consisting of rotation, translation and scale. When there are large numbers of examples with a uniform distribution of shape viewpoints and poses, rigid alignment with nearest example is practical and efficient. However, in small datasets, similar examples to the shape may not exist, and thus, non-rigid alignment is used. To study the effect of small number of exemplars, we group data into clusters and keep cluster centers as exemplars. The number of clusters is defined using a Gaussian mixture in an EM algorithm. With such an optimization, we are able to accurately cluster training data in a problem-dependent way without the need of prefixing clustering parameters.

2.1.1 Non-rigid 3D shape alignment

Retrieve nearest exemplar

The Histogram of Oriented Gradients (HOG) descriptor has been studied vastly in the domain of human detection and pose recognition. Here, the key idea is to use HOG as the human body descriptor in depth images, where the gradients of the depth image are the derivatives of the body hull surface. We apply HOG feature vector on the whole body to retrieve nearest exemplar. For non-rigid shape alignment, we tried to keep the smallest possible number of exemplars in the training data. Therefore, we use a problem-dependent clustering strategy to group HOG feature vectors of training data and keep each cluster center as possible exemplar.

To cope with the problem of determining the exact number of clusters, we estimate the optimum number of clusters by combining the EM and k-means algorithms as proposed in [97]: let $X = \{x_1, ..., x_N\}, x_i \in R^d$ be a given data set. An iterative algorithm starts from M_{min} to M_{max}, $M \in \{1, ..., N\}$, and at each iteration EM algorithm is initialized with the clusters of X obtained from k-means ($k = M$); then the parameters of the mixture model and the posterior probabilities of the

members of X are computed. At the end of each iteration the mutual relationship between every two mixtures is measured as:

$$\psi(i,j) = p(i,j)\log_2 \frac{p(i,j)}{p(i)p(j)}, i = 1,...,j, \tag{2.1}$$

$$p(i) = \frac{1}{N}\sum_{n=1}^{N} p(i|x_n) \tag{2.2}$$

$$p(i,j) = \frac{1}{N}\sum_{n=1}^{N} p(i|x_n)p(j|x_n) \tag{2.3}$$

where $p(i)$ is the probability of the mixture i and $p(i,j)$ is the joint probability of i and j mixtures. For any composition of $i, j \in \{1,...,M\}$, if $\psi(i,j) > 0$, then i and j mixtures are considered statistically dependent so the process finishes and $M-1$ is returned as the most suitable number of mixtures.

One of the limitations of such an approach is when there is no 'meaningful' mixture, i.e. when the number of training poses is low or when the data does not follow normal distributions. In the case of straggly or scarce data, algorithm goes to reach M_{max} where each data is assumed as a cluster itself. When the data distributions are not normal, we can tune M_{min} and M_{max} to solve this problem. After estimating the optimal M, the EM is trained and the labels and feature vectors of each model are kept. EM algorithm shows better cluster results than k-means, besides we can keep the parameters of EM and retrieve them later in order to predict the cluster of a new feature vector applying the posterior probabilities of the feature vector and the mixtures. This is useful to retrieve the closest model to test point cloud in an efficient way at test step.

Point matching

Once an exemplar retrieved, a set of points are sampled from shapes and reconstructed into 3D coordinates. Then, alignment is performed based on an iterative process in which the selected points are matched between the two clouds and refined at each iteration. Those detected points which do not match are considered as outliers and discarded in the next iteration. The process is repeated until a certain estimation error or a maximum number of iterations is reached. Fig. 2.2 shows the flow diagram of this approach. In the following we review this alignment process in detail.
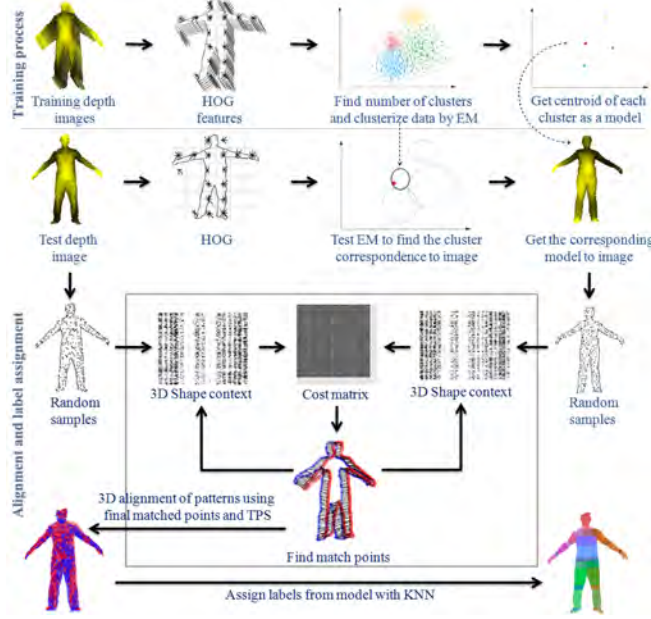
Figure 2.2: Process diagram.

Assume $P_{N \times 3}$ is the matrix of selected x, y, z points in the depth image coordinates of the model, $t_{N \times 1}$ is the matrix of gradient angles and $g_{N \times 2}$ is the matrix of normalized gradients. Then, the point cloud for the model is defined as:

$$P_{tan} = P + \alpha \left[g \circ \left[\cos(t) \,|\, \sin(t) \right] | 0 \right]_{N \times 3}, \tag{2.4}$$

where α is a static constant and $A \circ B$ is the Hadamard product of A and B. We use P_{tan} later to compute new gradient angles matrix of the model after converting it to real world coordinates.

Subsequently, we employ the basic shape context of [15] extended to 3D data. We propose to use exponential space for the radius of nested spheres (n_r) as:

$$r_i = \frac{10^{i/n_r} - 1}{9} \times max(\{||x - y|| \,|\, x, y \in \omega\}), i = 1, ..., n_r, \tag{2.5}$$

where ω is the set of inlier points (vs. outliers). This space partitioning forces shape context to be more sensitive to near samples to the sphere bin than farther ones. After computing the shape context histograms for all selected points, the

15

best matched points between all pairs of points on the model and the input test pattern are found. Such a matching process minimizes the overall matching cost, for which a cost table performs matching based on the histogram similarity and appearance similarity of the points. As in [15], we use χ^2 test to find the histogram similarity cost and the gradient angular difference polarity to find the appearance cost between pairs of (p_i, q_j) points of the two point clouds. So the cost function is defined as:

$$C(p_i, q_j) = \frac{1}{2}\left((1-\alpha) \sum_{k=1}^{\frac{n_r n_\theta^2}{2}} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} + \alpha(1 - \cos(t_i - t_j)) \right), \qquad (2.6)$$

where n_θ must be an even number, $h_i(k)$ and $h_j(k)$ denote the k-th bin of the histogram, and t_i and t_j are the gradient angles at p_i and q_j, respectively. The appearance cost acts as a penalty function causing smooth alignments on the surfaces, while the α coefficient controls this smoothing factor.

Redundant, "dummy" points are also added to the cost table with a constant cost to control the sensitivity of the shape context to noise as in [15]. Therefore, points that do not match any other with a lower value than this dummy cost will be considered as outliers. In hard assignments, each point matches exactly one point in the cost table so that the overall cost is minimum. This task, commonly referred to as Linear Assignment Problem (LAP), can be solved using [74].

Transformations

Sample points are aligned after matching to generate new coordinates and gradient angles which will be used in next iteration to refine the final matched points. This alignment task is done by generating an interpolation matrix using the best matches of random samples and thin plate spline (see [15, 19]) in the form of:

$$T = \begin{bmatrix} K_{N \times N} & [1|P_{N \times 3}] \\ [1|P_{N \times 3}]^\top & 0 \end{bmatrix}^{-1}_{(N+4)^2} \begin{bmatrix} Q_{N \times 3} \\ 0 \end{bmatrix}_{(N+4) \times 3}, \qquad (2.7)$$

where K is a kernel matrix, P is the best model matching points matrix, and Q is the best test pattern matching points matrix. K is computed as:

$$K_{ij} = \begin{cases} ||P_i - P_j|| \log(||P_i - P_j||) & \text{if } i \neq j, \\ \lambda & \text{if } i = j, \lambda > 0, \end{cases} \qquad (2.8)$$
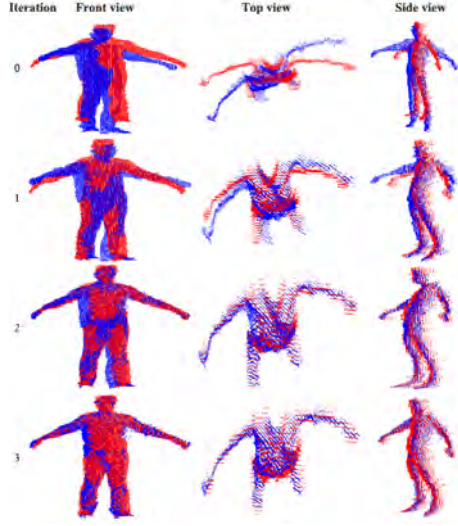
Figure 2.3: Iterative alignment process shows how points get closer at each step.

where $\lambda = \alpha^2 \lambda_0$ is a regularization parameter used to smooth the interpolation. The term α is defined as:

$$\alpha = \frac{1}{N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} ||P_i - P_j||, \tag{2.9}$$

and λ_0 is a scaling factor. We change 0 values of $\left\| P_i - P_j \right\|$ in the indices matrix K to 1 to avoid generating $-\infty$ in logarithm. Then, model sample points are mapped to their interpolated locations in the test pattern using the interpolation matrix T and the same procedure specified in equation 2.7 in the form $[K|1|O]T$, where O is the point cloud model and $K_{ij} = ||O_i - P_j|| \log(||O_i - P_j||)$. These new mapped points are sent to the next iteration along with new gradient angles. New gradient angles are updated in a procedure as follows: image coordinates of the mapped model samples are computed, P_{tan} points computed in previous sections are mapped using the matching points and the alignment procedure, and their image coordinates are estimated; finally the angles extracted from the differences of the coordinates of 2D mapped model and P_{tan} sample points are returned as new gradient angles. Figure 2.3 illustrates an alignment example within the described iterative process.

17

Label assignment

In order to cope with self-occlusions, we maintain a complete point cloud for each sample model. Once alignment has been done, the complete model is transformed to the test point cloud using the matching points of the former described approach through TPS. This warped complete model is used to complete occluded parts and assign labels. We can easily estimate the label of each point using its nearest neighbor pixel label after alignment of the point clouds by applying the matching points and the transformation procedure described above. Unfortunately, assignments of labels from 3D nearest neighbors cause problems in the case of imperfect alignments and broken segments. To minimize this issue, as well as noise points, we proposed to train SVM directly on 3D coordinates of warped model using a linear kernel and predict test points labels from it. SVM makes a bound around points and tune the assignments.

Experiments and results

To evaluate our method, we have created a dataset manually labeled containing 1155 frames (Chap. 4). We used a 10-fold cross validation over all 1155 frames to generate the results. The segmentation error per frame is the proportion of mislabeled pixels in relation to the total number of pixels. Then, the overall error is averaged.

We have used a block of 9×6 including 2×2 cells and 8 orientation bins for HOG. Figure 2.4 illustrates some clusters contents and shows how the poses and bodies are clusterized together. We achieved the best number of clusters at 90 in a range of [15..100] clusters. Although the results show the robustness of our approach to the HOG parameters, we found that the parameter values give the best alignment results: $n_\theta = 8$, $n_r = 15$, 500 random samples, dummy cost 0.1, 35% of samples as additional dummy points, appearance cost weight 0.15, and 4 iterations. We set the parameter λ_0 to 1000 for first iteration to have an affine transformation and br^{k-1} for next iterations where $b = 0.9$, $r = 1.7$, and k is the iteration number.

In order to compare our approach, we have considered the random forest (RF) pixel labeling approach as defined in [152]. In particular, the RF implementation computes the weights of each body part label as:

$$W_l = 0.5 + \frac{P(l)}{\sqrt{\sum_{i=1}^{N} P(i)^2}}, \tag{2.10}$$

where $P(l)$ is the probability of the label l and is equal to the averaged proportion of the number of pixels with label l compare to the total number of pixels in the body for some random images. In essence, this weight adjusts the probability of small
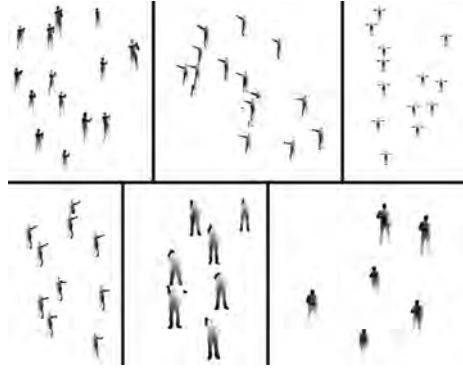
Figure 2.4: Some typical clusters are shown in this image among HOG and EM. The number of clusters is estimated for every combination of parameters. We employed randomly 90% of data to train 10% to test.

vs. large segments. The whole approach is implemented in C++ using the OpenCV library, and the computational time for the complete soft biometric estimation is around 50 seconds: less than 1s for nearest model finding, 2s for point sampling, 14s for alignment and 33s for label assignment.

2.1.2 Segmentation results

Given the results shown in Figure 2.5, we fixed the number of cluster to be 90 and progressively increasing the number of training images in order to test our multi-part segmentation methodology. The results in Figure 2.5 illustrate a low sensitivity of our approach to the number of training data: human body segmentation accuracy is improved between 10% and 20% compared to RF. On the other hand, RF trend shows that segmentation errors remain stable and is not able to improve for higher amounts of training data. We plot the qualitative results of segmentation in Figure 2.6, where the influence of alignment in the segmentation is shown. Segmentation errors occur in the areas for which alignment is not perfect, so SVM has incorrect labels. Another source of error comes from inconsistencies of manual labeling for different samples. The results in the image shows how the approach copes with the self occlusions.

The similarity of the test pattern to the estimated model plays also an important role. In this case, higher number of random samples will generate better alignment and segmentation result, and higher complexity instead. Using perfect alignment parameters implies a low number of iterations, whereas a high number of iterations
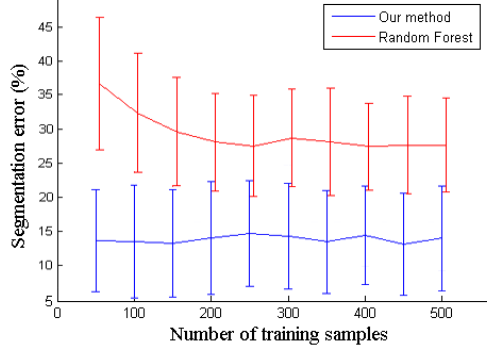
Figure 2.5: In our approach, segmentation error remains stable in different amount of training data in comparison to RF. Average error percentage for our method and RF at the best case is 13.55 ± 7.01 and 32.26 ± 10.03, respectively.

will reduce the accuracy dramatically in some cases. We observed that the affine behavior of λ_0 at first iteration generated quite better transformation results. We also implemented soft-assignment vs. hard-assignment which is faster but no difference in accuracy was found.

2.1.3   Rigid 3D shape alignment

Several state-of-the-art works [82, 150, 195] use Random forest (RF) to extract viewpoint or nearest neighbors from the deeper branches of the trees trained on a particular dataset. Such methodology can be seen as stochastic shape extraction and leads to some irrelevant nearest shape recovery. Besides that, this approach is not efficient for large scale datasets. On the other hand, common statistical shape descriptors try to find a correlation among the components composing the shape and grouping them into bins. For rigid shape alignment, we developed a 3D shape descriptor and retrieved a number of nearest shapes.

In order to evaluate our method on highly-variable poses and viewpoints, as well as temporal analysis, we created a rich synthetic hand dataset mimicking the features of commodity depth cameras (Chap. 4) in which 3D joint locations and pixel labels are available. We illustrate some properties of the hand model used to create this dataset in Fig. 3.1. We created a hand model with 25 semantic segments used as low-level pixel labels in the dataset. At a higher level of semantics, we segmented the hand by assigning each pixel a label from the set $L = \{l_1, ..., l_6\}$, where L represents fingers and the palm.

Figure 2.6: Qualitative results. First column shows the nearest model found, second column is the test sample, third column is the warped model after registration, and the next two columns belong to our approach segmentation and RF respectively. Black points correspond to segment lines used for soft biometrics measurements (described in Chap. 4).

Nearest shapes extraction

In the proposed descriptor, we train a classifier to segment a hand into a set $S = \{s_i\}_{i=1}^{25}$ with 25 classes defined in the dataset and group probability responses of the classifier into log-polar bins. Therefore we first select a fixed random number of pixels from the hand and estimate each class response for each pixel applying the trained classifier. For aggregating the responses into bins, we reconstruct a point cloud of selected pixels and divide XYZ axes into three axis pairs XY, XZ and YZ. Thus, we map the point cloud to front, top and side views and apply measurements separately on each view.

We compute the log-polar binning based on shape context [15]. Let $q = \frac{1}{N} \sum_{i=1}^{N} P_i$ be the center of the point cloud where N is the number of points and let $P_i \in \mathbb{R}^3$ denote the i-th point in world coordinates. We set q as the center of the log-polar

coordinate system. Then histograms of different views (front view for instance) are computed as:

$$H_{xy}(k,c) = \sum_{i=1}^{N} \{R_{ic} | (P_i^{xy} - q^{xy}) \in bin_{xy}(k)\}, \tag{2.11}$$

where R_{ic} denotes probability responses of the i-th point and c-th class predicted by the classifier, and k is the bin number. Finally histograms at each view are concatenated and normalized. Applying such descriptor we discriminate both spatial and class dependencies of different shape points into bins, being fast to compute, invariant to slight rotations of the hand and robust against boundary noise due to the random selection of points. We show an illustration of our descriptor in Fig. 3.2. We set 8 angle and 5 radius bins as the log-polar binning parameters of the shape descriptor. Finally, for a fast extraction of the K nearest shapes, a kd-tree is trained based on the extracted features. We apply the work of Shotton et al. [153] as our segmentation classifier and train 14 trees with depth 20 using 100 random features, 150K random samples with a subset of 500 randomly selected pixels per frame and 1000 uniformly distributed offsets with a scaling factor of 120mm. To fit data in memory we train each tree with 23% of random data.

Palm and finger segmentation

Nearest shapes can vary in shape and pose and need to be aligned to each other beforehand. We use palm joints of nearest shapes to align them through Procrustes analysis. This provides a uniform and smooth distribution of palm points in the point cloud of the nearest shapes. Given this point cloud with their corresponding labels l_i, we find an affine transformation A with scaling factor s to hand point cloud P by applying iterative closest point matching (ICP) [207]. For a faster convergence, we modify ICP process to find closest points from group of points with the same label. Pixel labels of test frame were estimated by RF beforehand. Then, we get the palm joints by transforming the nearest shape joints given A and s.

Although our trained RF could segment the hand, it is not reliable under some situations, especially for distinguishing fingers (See Fig. 2.7 for some samples). Quadratic discriminant analysis provides a proper way to assign each point in the point cloud in query a label from aligned point cloud of nearest shapes efficiently.

Results

By incorporating QDA for segmenting hand into the set L, we could improve RF segmentation performance. Since each segment l_i has a number of sub-segments from the set S, for a given pixel P belonging to segment l_i, we discard those proba-
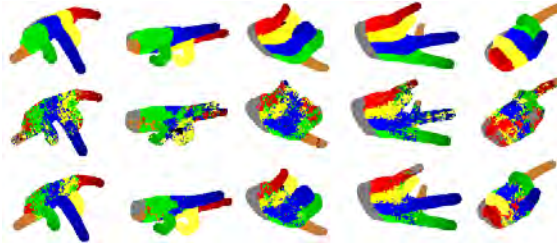
Figure 2.7: Qualitative RF performance. Rows from top to bottom: ground truth, RF, and improved RF results. We improve RF segmentation performance by around 20%.

bilities (given by RF) not belonging to l_i, and consider the index of the maximum probability as the final estimated label for that pixel. Fig. 2.7 illustrates some qualitative results of RF segmentation performance and its improvement in a number of frames.

2.2 CNN based face segmentation

Semantic segmentation is a very important topic in computer vision because of its many applications in object recognition, image annotation, image coding, scene understanding and biomedical image processing. One specific field of semantic segmentation is face segmentation in which the task is to correctly assign labels of face regions such as nose, mouth, eye, hair, etc. to each pixel in a face image. Face segmentation techniques are frequently used in security systems and in the field of human computer interaction, mainly in order to facilitate the problems of face detection and recognition [3, 112, 117, 135], and emotion/expression recognition [29, 58, 160]. Further specialized entertainment oriented applications of face segmentation include style transfer [41], virtual make-up application [104], virtual face-swapping [87] and 3D performance capturing [144]. Additionally, utilization of face segmentation techniques could potentially improve the performance in several other computer vision tasks involving the processing of face images such as apparent personality prediction [53] and face hallucination [54].

Semantic segmentation of faces is a difficult problem because of the large number of variable conditions that need to be considered, especially when applied to face pictures taken in uncontrolled environments. These conditions include variations in facial expression, skin color, lighting, image quality, pose, hair texture and style, as well as the presence of varying amounts of background clutter and

Table 2.1: **An overview of the differences and the similarities between the variants of our model, and the recent semantic segmentation models that they are based on.** ψ_u and ψ_p denote learned instead of fixed unary potential and pairwise potential of a conditional random field, respectively.

| | adversarial training | conditional random field | | | dilated conv. |
		(ψ_u)	(ψ_p)	(end-to-end)	
Yu and Koltun (2015) [202]	×	—	—	—	✓
Liu et al. (2015) [105]	×	✓	✓	×	×
Zheng et al. (2015) [209]	×	✓	×	✓	×
Luc et al. (2016) [109]	✓	—	—	—	✓
Cnn (Ours)	×	—	—	—	✓
CnnGan (Ours)	✓	—	—	—	✓
CnnRnn (Ours)	×	✓	✓	✓	✓
CnnRnnGan (Ours)	✓	✓	✓	✓	✓

occlusions. Furthermore, despite extensive studies in face segmentation, correctly classifying hair pixels still remains a particularly challenging task [184], mainly due to the inherent properties of hair such as color similarity to background, non-rigidity and non-unique shape.

Recently, there has been a sizable number of advances in semantic segmentation. In the context of semantic face segmentation, [105] showed that formulating the unary potential and the pairwise potential of a conditional random field (CRF) over a four-connected graph as a convolutional neural network (CNN) resulted in state-of-the-art accuracy on the Part Labels dataset [76, 94] and the Helen dataset [93, 158]. While this model was not end-to-end trainable and relied on graph cuts, it learned both the the unary potential and the pairwise potential of the CRF. In the context of semantic image segmentation, [209] showed that formulating the iterative update equation of a CRF over a fully-connected graph [89] as a recurrent neural network (RNN) resulted in state-of-the-art accuracy on Pascal VOC 2012 dataset [43]. While this model did not learn the pairwise potential of the CRF and relied on fixed Gaussian kernels, it was end-to-end trainable.

Furthermore, [202] showed that the results of convolutional semantic segmentation models can be improved by using dilated kernels instead of regular kernels, which increase receptive field size without decreasing receptive field resolution.

Similarly, [109] showed that results of convolutional semantic segmentation models can be improved by using an adversarial loss function in addition to a segmentation loss function, which enforces higher-order consistencies without

explicitly taking into account any higher-order potentials.

Here, our goal is to formulate a model for semantic face segmentation by combining the respective strengths of the aforementioned models. That is, the model should learn both the unary potential and the pairwise potential of a CRF over a four-connected graph like [105], and be end-to-end trainable like [209] while aggregating multiscale contextual information like [202], and detecting and correcting higher-order inconsistencies like [109]. Table 2.1 shows an overview of the differences and similarities between our proposed model (i.e., CnnRnnGan), its variants (i.e., Cnn, CnnGan and CnnRnn), and the recent semantic segmentation models that they are based on.

The contributions of our work are the following:

1. We introduce an end-to-end trainable convolutional and recurrent neural network formulation of a conditional random field over a four-connected graph for face segmentation, which is augmented with dilated convolutions and adversarial training.

2. We exploit the structured nature of faces by conditioning the model on face landmarks, and/or training multiple models for different face landmarks and aggregating their outputs.

3. We achieve state-of-the-art results on the Part Labels dataset and the Helen dataset.

2.2.1   Related Work

Semantic segmentation has been widely studied in computer vision in a wide spectrum of domains. For a comprehensive review of classical approaches for semantic segmentation, we refer the reader to [216]. In this section, we review recent work on semantic segmentation in general and semantic face segmentation in particular.

The most recent state-of-the-art semantic segmentation models almost exclusively rely on convolutional neural networks. In contrast to earlier approaches where recognition architectures were directly used for semantic segmentation [46], current approaches utilize architectures that are carefully adapted for the task at hand. [107] proposed the first such approach, where the fully-connected layers of popular architectures such as AlexNet [90], VGGNet [156] and GoogLeNet [163] were replaced with (de)convolution layers and combined with earlier layers to enable dense and high resolution predictions. Since then, this approach has been continuously improved by the introduction of more sophisticated architectures, which
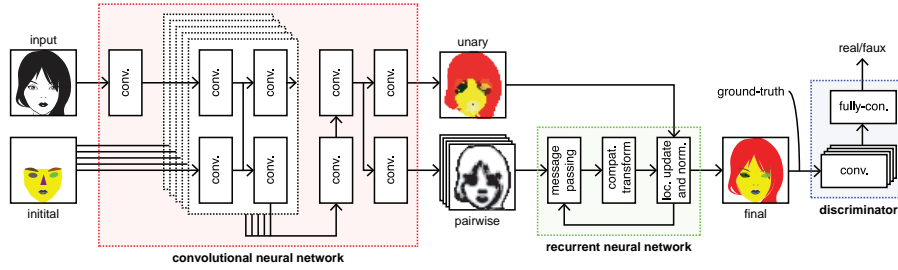
Figure 2.8: **Our proposed end-to-end semantic face segmentation model.** The conditional random field is formulated as a composition of two neural networks: i) A convolutional neural network, which nonlinearly transforms an input face and its initial segmentation to the unary potential and the pairwise kernels of the conditional random field. ii) A recurrent neural network, which transforms the unary potential and the pairwise kernels of the conditional random field to the final segmentation of the input face. In the training phase, a discriminator and the conditional random field play a two-player minimax game, in which the objective of the discriminator is distinguishing ground-truth segmentations from final segmentations, and the objective of the conditional random field is fooling the discriminator.

enabled denser [9, 66, 124], higher resolution predictions [49], and/or encoder-decoder [208] predictions. In particular, [202] proposed dilated convolutions for dense prediction, where contextual information could be aggregated by multiscale levels without loss of neither resolution nor coverage. This idea has been extended by [23] to enable a larger field of view through spatial pyramid pooling. Such approaches enjoy the benefits of dense and high resolution predictions without the burden of extra parameters.

At the same time, conditional random fields have been used in semantic segmentation for postprocessing outputs of region-level or pixel-level semantic segmentation models. While the relatively small number of outputs of region-based semantic segmentation models could be postprocessed by CRFs with dense pairwise connectivity [76], the relatively large number of outputs of pixel-level semantic segmentation models could only be postprocessed by CRFs with sparse pairwise connectivity [105]. In a seminal work, [89] proposed an efficient iterative algorithm for approximate inference in fully-connected CRFs with Gaussian edge potentials, which has been widely adopted for postprocessing outputs of pixel-level segmentation models [9, 66, 124]. [209] formulated this algorithm as a recurrent neural network, which is trained along with a pixel-level segmentation model instead of

postprocessing it. This formulation is reminiscent of the pixel-level semantic segmentation model in [132], whose outputs were iteratively refined with a recurrent convolutional neural network.

Recently, generative adversarial networks (GANs) [51] have received particular attention in computer vision [24, 35, 138]. The idea behind GANs is training a discriminator and a generator by letting them play a two-player minimax game. In this game, the objective of the discriminator is distinguishing samples that are drawn from the data distribution from samples that are drawn from the model distribution, and the objective of the generator is fooling the discriminator. While GANs have been proposed for estimating generative models via an adversarial process, they have been widely adopted for other tasks such as inpainting [131], style transfer [98] and super-resolution [95] as loss functions. In particular, [109] estimated a semantic segmentation model via an adversarial process by training a discriminator for distinguishing ground-truths from outputs of the semantic segmentation model and the semantic segmentation model for fooling the discriminator. They showed that this process leads to improved results on the Stanford Background dataset [52] and the PASCAL VOC 2012 dataset.

There has been relatively fewer semantic face segmentation models that rely on convolutional neural networks. Most earlier models were based on CRFs [76], hand designed features [188] and exemplars [158]. Kae et al. [76] modeled global part dependencies using a restricted Boltzmann machine to have an overall realistic shape while local shape details were modeled through a CRF, whereas Smith et al. [158] used exemplar-based non-rigid warping for face segmentation. Despite the progress in the models, hair segmentation is still the most challenging part due to its color and style variability. Earlier works include attempts of modeling hair, skin and background color [148, 197], mixture of hair styles [96] or MRF/CRF labeling [67]. As a specialized hair segmentation, Wang et al. [185] applied co-occurrence probabilities of face components identified by a Markov random field. The final segmentations were constrained in a tree-structured model built over part co-occurrences.

Among CNN-based face segmentation methods, [111] segmented faces based on a hierarchical part detection process, where the face was detected as the root of the hierarchy and the smallest components of the face were detected at the bottom of such hierarchy. Then, an autoencoder network transformed those detected components into label maps. As a result of using hierarchies, partially occluded faces could be easily handled. Recently, [214] applied the part detection idea by training one network for each part and mapping the segmentation result to the original image. [105] generated pairwise terms as class edge potentials through a four-connected graph. Such edge potentials extracted in a multi-objective network along with unary terms were trained using non-structured loss functions, and

27
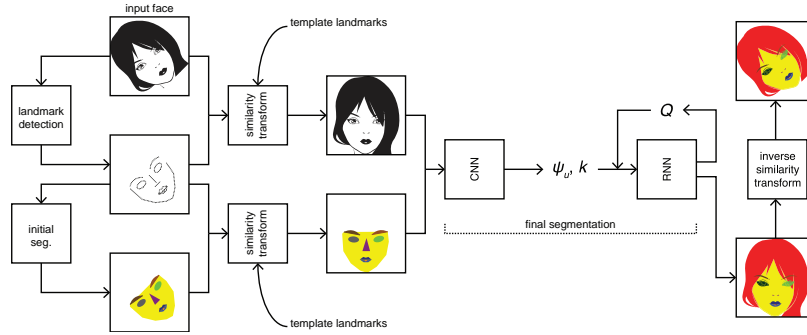
Figure 2.9: **Our semantic face segmentation pipeline.** First, 68 landmarks of the
the input face are detected. An initial segmentation of the input face is obtained by
filling the regions that are formed by connecting the landmarks around background,
face skin, left eyebrow, right eyebrow, left eye, right eye, nose, upper lip, inner
mouth and lower lip. A similarity transformation from the landmarks of the input
face to the landmarks of a template face is estimated. The input face and the
initial segmentation of the input face are warped to the template face by using
the similarity transformation, and resized to 500 pixels × 500 pixels. The final
segmentation of the face is obtained by using our model. Optionally, the final
segmentation of the image is warped back from the template face by using the
inverse of the similarity transformation.

provided prior knowledge to the network by including inaccurate segmentations
as an additional network input. This study showed the benefits of including prior
knowledge for improving face segmentation.

2.2.2   End-to-end semantic face segmentation

For end-to-end semantic face segmentation, we formulate a conditional random
field as a composition of a convolutional neural network and a recurrent neural
network (Section 2.2.2). The convolutional neural network is used for obtaining
the unary potential and the pairwise kernels of the conditional random field as a
function of an input face and its initial segmentation (Section 2.2.2). The recur-
rent neural network is used for obtaining the label compatibility function and a
mean field approximation of the Gibbs distribution of the conditional random field
as a function of the unary potential and the pairwise kernels of the conditional
random field (Section 2.2.2). In the training phase, a discriminator and the condi-
tional random field play a two-player minimax game, in which the objective of the

discriminator is distinguishing ground-truth segmentations from final segmentations, and the objective of the conditional random field is fooling the discriminator (Section 2.2.2). Fig. 2.8 illustrates main components of our model.

Prior to entering the model, an input face is preprocessed as follows: A template face is obtained by averaging the faces in the training set. Sixty-eight landmarks of the template face and the input face are detected by using the dlib implementation [84] of an ensemble of regression trees [78].[1] An initial segmentation of the input face is obtained by filling the regions that are formed by connecting the landmarks around background, face skin, left eyebrow, right eyebrow, left eye, right eye, nose, upper lip, inner mouth and lower lip. A similarity transformation from the landmarks of the input face to the landmarks of the template face is estimated. The input face and its initial segmentation are warped to the template face by using the similarity transformation, and resized to 500 pixels × 500 pixels. The final segmentation of the input face is obtained by using our model. Optionally, the final segmentation of the input face can be resized back from 500 pixels × 500 pixels and warped back from the template face by using the inverse of the similarity transformation. Fig. 2.9 illustrates our preprocessing pipeline.

Conditional Random Field

We begin the exposition of our model by considering a conditional random field over a four-connected graph. Let $\mathbf{I} = \{I_1, \ldots, I_N\}$ and $\mathbf{X} = \{X_1, \ldots, X_N\}$ be random fields, where $I_i \in \mathbb{R}^3$ and $X_i \in \mathscr{L} = \{l_1, \ldots, l_P\}$ are the color vector and the label of the pixel $i \in \{1, \ldots, N = h \times w\}$, respectively. Let $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ be a four-connected graph, where $\mathscr{V}$ contains all pixels, and $\mathscr{E}$ contains all pixel pairs that have a taxicab metric of one.

The conditional random field $(\mathbf{I}, \mathbf{X})$ over $\mathscr{G}$ is defined by the following Gibbs distribution:

$$P(\mathbf{X} = \mathbf{x}|\mathbf{I}) = \frac{1}{Z(\mathbf{I})} \exp\left(-E(\mathbf{x}|\mathbf{I})\right) \tag{2.12}$$

where Z is the partition function, and E is the following Gibbs energy:

$$E(\mathbf{x}) = \sum_{i \in \mathscr{V}} \psi_u(x_i) + \sum_{i,j \in \mathscr{E}} \psi_p(x_i, x_j) \tag{2.13}$$

where ψ_u is the unary potential, which is the cost of assigning the label x_i to the pixel i, and ψ_p is the pairwise potential, which is the cost of assigning the labels x_i

---

[1]Note that the dataset that was used for training the landmark detection model provided by dlib contains some of the images that we use to test our final segmentation model. To avoid circular analysis, we retrained the landmark detection model on the same dataset that it was originally trained on after removing these images.

and x_j to the pixels i and j, respectively. Note that we omit conditioning on **I** for notational convenience. The pairwise potential is of the following form:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) k_{i,j} \tag{2.14}$$

where μ is a label compatibility function, which is not assumed to be symmetric since it was shown that this assumption improves semantic segmentation results [209], and k is arbitrary pairwise kernels.

Following [89], we approximate the Gibbs distribution with the mean field distribution that minimizes the Kullback–Leibler divergence between the Gibbs distribution and the distributions that are of the following form:[2]

$$Q(\mathbf{X}) = \prod_{i \in \mathcal{V}} Q_i(X_i) \tag{2.15}$$

This approximation results in the following iterative update equation:

$$Q_i(x_i = l) = \frac{1}{Z_i} \exp\left(-\psi_u(x_i) - \sum_{l' \in \mathcal{L}} \mu(l, l') \sum_{i,j \in \mathcal{E}} k_{i,j} Q_j(l')\right) \tag{2.16}$$

Convolutional Neural Network

Following [105], we formulate ψ_u and k as a convolutional neural network, whose architecture is inspired by recent ones proposed in [77, 181, 202].

The network comprises the following layers:

1. One convolution layer that has 32 kernels of size 3×3 with no nonlinearities.

2. Five blocks, where each block comprises the following layers:

    a Two parallel convolution layers that have 64 kernels of size 1×1 with no nonlinearities (i.e. bias layer) and 64 dilated kernels of size 3×3 with gated activation units [181] (i.e. weight layer). The input of the bias layer is the initial segmentation. The output of the bias layer is summed with the activation of the weight layer. The output of the weight layer becomes the input of the next layer.

    b Two parallel convolution layers that have 64 kernels of size 1×1 with no nonlinearities (i.e. residual layer) and 64 kernels of size 1×1 with

---

[2]While the Gibbs energy can be converted to a submodular energy, which makes exact inference (e.g. with combinatorial min cut/max flow algorithms) possible, we resort to approximate inference (i.e. with mean field theory) to be able to formulate it as a recurrent neural network, which makes end-to-end training possible.

rectified linear units (i.e. skip layer). The output of the residual layer is summed with the input of the block, which becomes the input of the next layer. The output of the skip layer is concatenated with the outputs of the skip layers of the remaining blocks along the channel axis, which becomes the input of the next layer after the last block.

3. One convolution layer that has 160 kernels of size 1×1 with rectified linear units.

4. Two parallel convolution layers that have P kernels of size 1×1 with no nonlinearities (i.e., ψ_u) and four kernels of size 1×1 with exponential units (i.e., k).

Dilated kernels are the same as the regular kernels with the exception that successive kernel elements have holes between each other, whose size is determined by a dilation factor. As a result, they increase receptive field size without decreasing receptive field resolution. Note that regular convolution layers can be considered dilated convolution layers with a dilation factor of one.

The dilation factor of the first block is one, which is doubled after every block. The number of blocks (i.e., five) is chosen to be the largest possible value such that the receptive field dimensions of the last block is less than or equal to the pixel dimensions. That is:

$$q = \underset{x}{\operatorname{argmax}} f(x) : f(x) = 3 + 2 \sum_{i=0}^{x-1} 2^i \leq \min(h, w) \tag{2.17}$$

where q is the number of blocks.

Recurrent Neural Network

Following [3], we formulate the iterative update equation (eq. 2.16) as a recurrent neural network. The network comprises (i) a message passing layer, (ii) a compatibility transform layer, and (iii) a local update and normalization layer. Note that only the compatibility transform layer has free parameters.

The layers are implemented as follows: Let ψ_u be a $P \times h \times w$ tensor and k be a $4 \times h \times w$ tensor, which are the outputs of the convolutional neural network. Prior to the first iteration, Q is initialized with ψ_u, and the channels of k are broadcasted to the shape of Q, which results in a set of four $P \times h \times w$ tensors.

- In the message passing layer, Q is shifted up, right, down and left by one pixel, and multiplied (i.e. Hadamard product) with the corresponding elements of k, which results in a set of four $P \times h \times w$ tensors. The elements of this set are

31

summed. As a result, this layer outputs a $P \times h \times w$ tensor, which becomes the input of the next layer.

- In the compatibility transform layer, the input tensor is convolved with P kernels of size 1×1. As a result, this layer outputs a $P \times h \times w$ tensor, which becomes the input of the next layer.

- In the local update and normalization layer, the input tensor is subtracted from $-\psi_u$, exponentiated and normalized (i.e., softmax function). As a result, this layer outputs a $P \times h \times w$ tensor, which becomes the input of the first layer after the first four iterations and the output of the network after the fifth iteration.

Adversarial Training

While this end-to-end trainable convolutional and recurrent neural network formulation of a conditional random field can learn both the unary potential and the pairwise potential, it does not take into account any higher-order potentials that can enforce higher-order consistencies. To be able to enforce higher-order consistencies without explicitly taking into account any higher-order potentials, we train the model by minimizing an adversarial loss function in addition to a segmentation loss function [109].

To this end, next to our model (which is from now on referred to as the generator) we train a discriminator. We denote the output of the discriminator as $D_{\theta_D}(.)$, which is the probability that the input of the discriminator is a ground-truth segmentation. We denote the output of the generator as $G_{\theta_G}(.)$, which is the probabilities of assigning each of the P labels to each of the N pixels of the input. In this context, the goal of the discriminator is to distinguish ground-truth segmentations from generated segmentations, whereas the goal of the generator is to generate segmentations that are indistinguishable from ground-truth segmentations. That is, they play the following minimax game:

$$\min_{\theta_G} \max_{\theta_D} \quad \mathbb{E}_{\mathcal{T}^{(n)} \sim p_{\mathcal{T}}(\mathcal{T}^{(n)})} \log D_{\theta_D}\left(\mathcal{T}^{(n)}\right) +$$
$$\mathbb{E}_{\mathcal{I}^{(n)} \sim p_{\mathcal{I}}(\mathcal{I}^{(n)})} \log\left(1 - D_{\theta_D}\left(G_{\theta_G}\left(\mathcal{I}^{(n)}\right)\right)\right) \tag{2.18}$$

where $\mathcal{I} = \{\mathcal{I}^{(1)}, \mathcal{I}^{(2)}, \ldots\}$ is a set of images, and $\mathcal{T} = \{\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \ldots\}$ is a set of corresponding ground-truth segmentations.

We formulate the discriminator as a convolutional neural network whose architecture is inspired by the architecture in [138]. The network comprises four convolution layers and a fully-connected layer. The ith convolution layer has 2^{6+i} kernels with a size of 3×3, a stride of 2×2, a pad of 1×1 and leaky rectified

units [114]. The activations of the first four convolution layers are normalized along the mini-batch (i.e., batch normalization [70]). The output of the last convolution layer is averaged along the spatial axes (i.e., global average pooling [101]). The fully-connected layer has one kernel with a sigmoid unit.

The discriminator is trained by iteratively minimizing the following discriminator loss function:

$$L_{dis} = -\log D_{\theta_D}\left(\mathscr{T}^{(n)}\right) - \log\left(1 - D_{\theta_D}\left(G_{\theta_G}\left(\mathscr{I}^{(n)}\right)\right)\right) \tag{2.19}$$

Note that L_{dis} is the sum of two sigmoid cross entropy loss functions.

The generator is trained by iteratively minimizing the following linear combination of an adversarial loss function and a segmentation loss function:

$$L_{gen} = L_{adv} + \lambda L_{seg} \tag{2.20}$$

where λ is the coefficient of the segmentation loss function and the constituent loss functions are of the following forms:

$$L_{adv} = -\log D_{\theta_D}\left(G_{\theta_G}\left(\mathscr{I}^{(n)}\right)\right) \tag{2.21}$$

$$L_{seg} = -\sum_{l\in\mathscr{L}}\sum_{i\in\mathscr{V}}\mathscr{T}_{l,i}^{(n)}\log G_{\theta_G}\left(\mathscr{I}^{(n)}\right)_{l,i} \tag{2.22}$$

Note that L_{adv} is a sigmoid cross entropy loss function, and L_{seg} is a softmax cross entropy loss function.

2.2.3 Results

Implementation details

The models were implemented in Chainer with CUDA and cuDNN [170].

The biases of the models were initialized with zero, the weights of the models were initialized with samples drawn from a scaled Gaussian distribution [61], and the coefficient of the segmentation loss function (i.e., λ) was set to 100.

Adam [85] with initial $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = $ 1e-8 was used to iteratively train the models on the combination of the training set and the validation set[3] for 111 epochs. The learning rate (i.e., α) was reduced by a factor of 10 after 100 and 110 epochs.

At each iteration, the discriminator and the generator were updated sequentially. To prevent them from overpowering each other, the training of the discriminator

---

[3]The hyperparameters (i.e., λ, α and the number of epochs) were optimized prior to combining the training set and the validation set.

Table 2.2: **The results of the main experiment on the Part Labels dataset.** Confidence matrix is reported in terms of percentage. The rest of the results are reported in terms of Jaccard index (i.e. intersection over union) of the classes and their arithmetic mean, respectively.

		predicted class			
		[ ]	◯	🖤	🖤
true class	[ ]	97.97	00.73	01.30	——
	◯	01.83	96.37	01.79	——
	🖤	06.35	05.44	88.21	——
Jaccard index		.9656	.9182	.7808	.8882

was suspended or resumed if the following conditions were satisfied, respectively:

$$\frac{L_{dis}}{L_{adv}} < 0.1, \frac{L_{dis}}{L_{adv}} > 0.5 \qquad (2.23)$$

Similarly, the training of the generator was suspended or resumed if the following conditions were satisfied, respectively:

$$\frac{L_{dis}}{L_{adv}} > 10, \frac{L_{dis}}{L_{adv}} < 2 \qquad (2.24)$$

These conditions were selected based on [38].

Datasets

We analyzed the Part Labels dataset and the Helen dataset in our experiments. These datasets are the standard benchmark datasets for semantic face segmentation, which comprise pairs of in-the-wild faces and ground-truth segmentations.

Parts Label dataset comprises 2927 pairs of in-the-wild faces and ground-truth segmentations of background, face skin (including ear skin and neck skin) and hair (including facial hair), which is split in a 1500 pair training set, a 500 pair validation set and a 927 pair test set.

Helen dataset comprises 2330 pairs of in-the-wild faces and ground-truth segmentations of face skin (excluding ear skin and neck skin), left eyebrow, right eyebrow, left eye, right eye, nose, upper lip, inner mouth, lower lip and hair (exclud-

Table 2.3: **The results of the main experiment on the Helen dataset.** Confidence matrix is reported in terms of percentage. The rest of the results are reported in terms of Jaccard index (i.e. intersection over union) of the classes and their arithmetic mean, respectively.

true class \ predicted class	[ ]	○	∼	eye	nose	upper lip	teeth	lower lip	hair	hair+face
[ ]	97.28	00.41							02.30	——
○	01.83	95.46	00.43	00.16	00.36	00.13	00.01	00.18	01.44	——
∼	00.05	19.66	80.22						00.06	——
eye		13.25	00.02	86.73						——
nose		07.23			92.77					——
upper lip		14.16			00.01	80.90	03.63	01.30		——
teeth		02.35				09.49	82.20	05.96		——
lower lip	00.06	09.65					04.46	84.83		——
hair	16.38	02.70	00.11						80.81	——
Jaccard index	.9452	.8933	.6987	.7974	.8884	.6619	.7467	.7580	.6962	.7873

ing facial hair), which is split in a 2000 pair training set, a 230 pair validation set and a 100 pair test set.

Evaluation metrics

Results are reported in terms of confusion matrix and Jaccard index (i.e., intersection over union). Confusion matrix is defined as the square matrix $\mathbf{A}$ where $A_{i,j}$ is the number of pixels whose true class is i and predicted class is j. Jaccard index of class i is defined as follows:

$$J_i = \frac{A_{i,i}}{\sum_j A_{i,j} + \sum_j A_{j,i} - A_{i,i}} \tag{2.25}$$

Jaccard index of all classes is defined as follows:

$$J = \frac{\sum_i J_i}{P} \tag{2.26}$$

35

Main Experiments

We conducted two main experiments on the Labeled Parts and the Helen datasets, in which we evaluated the CnnRnnGan model.

Part Labels Dataset

We iteratively trained one global CnnRnnGan model for segmenting background, face skin and hair. Before the first iteration, the images in the dataset were resized to 106 pixels × 106 pixels. At each iteration, a mini-batch of size 16 was randomly selected without replacement, horizontally and vertically translated by ± 5 pixels, and mirrored in the left-right direction. Then, the mini-batch was cropped to the central 96 pixels × 96 pixels.

In the test phase, the inputs were oversampled (i.e., center and corners) and mirrored (i.e., left-right direction). The outputs were placed to their corresponding locations in the original inputs and averaged. Table 2.2 shows the resulting confusion matrix and Jaccard index. The most common cause of errors was mislabeling the classes as background. The least common cause of errors was mislabeling the classes as hair. All of the classes were segmented with a relatively high accuracy ($J = 0.8882$). Background was the most accurately segmented class ($J_b = 0.9656$). Hair was the least accurately segmented class ($J_h = 0.7808$).

Helen Dataset

We iteratively trained the following five CnnRnnGan models for segmenting different classes:

- One global model for segmenting background, face skin and hair.

- Three local models for segmenting eyebrows, eyes and nose, respectively.

- One local model for segmenting upper lip, inner mouth and lower lip.

The outputs of the global model and the local models were aggregated by resizing the output of the global model to 500 pixels × 500 pixels and placing the non-background outputs of the local models to their corresponding locations in the resized output of the global model.

The global model was trained on the Helen dataset in the exact same way as it was trained on the Part Labels dataset. The local models were trained in a slightly different way than that in which the global models were trained. Before the first iteration, the images in the dataset were cropped to 90 pixels × 90 pixels such that their centers coincided with the centers of the corresponding classes of the average face. At each iteration, a mini-batch of size 16 was randomly selected without replacement, rotated by ± 7.5 degrees, scaled by a factor of 1 ± 0.05, horizontally and vertically translated by ± 5 pixels, and randomly flipped in the left-right direction. Additionally, the initial segmentations were further randomly rotated by ± 0.75

Table 2.4: **Comparison of our results versus the previous state-of-the-art on the Part Labels dataset.** The overall results are reported in terms of pixel and superpixel accuracy, respectively. The rest of the results are reported in terms of F_1 score.

Kae et al. (2013) [76]	——	——	——	——	94.95
Tsogkas et al. (2015) [178]	——	——	——	——	96.97
Liu et al. (2015) [105]	97.10	93.93	80.70	95.12	——
Zheng et al. (2015) [208]	——	——	——	——	96.59
Saxena et al. (2016) [146]	——	——	——	94.82	95.63
Ours	**98.25**	**95.74**	**87.69**	**96.67**	**97.16**

degrees, scaled by a factor of 1 ± 0.005, and horizontally and vertically translated by ± 0.5 pixels. The additional data augmentation was used to further avoid overfitting the training set since the training set had a small overlap with the training set of the landmark detection model. Finally, the mini-batch was cropped to the central 80 pixels × 80 pixels.

In the test phase, the inputs were oversampled (i.e., center and corners) and mirrored (i.e., left-right direction). The outputs were placed to their corresponding locations in the original inputs and averaged. Table 2.3 shows the resulting confusion matrix and Jaccard index. The most common cause of errors was mislabeling the classes as face skin and background. The least common cause of errors was mislabeling the classes as eyes and nose. Importantly, when the non-background outputs of the local models were misclassified, they were almost always misclassified as the output of the global model and almost never as one another, which suggests that the simple post-hoc aggregation of the outputs of the global model and the local models was sufficient. All of the classes were segmented with a relatively high accuracy ($J = 0.7873$). Background and face skin were the most accurately segmented classes ($J_b = 0.9452$ and $J_{fs} = 0.8933$). Hair and upper lip were the least accurately segmented classes ($J_h = 0.6962$ and $J_{ul} = 0.6619$).

Compared to the accuracy of hair segmentations on the Part Labels dataset, accuracy of hair segmentations on the Helen dataset was considerably lower ($J_h = 0.7808$ versus $J_h = 0.6962$). This discrepancy can be attributed to the way in which hair was annotated in the datasets. In the Part Labels dataset, hair was annotated by automatically segmenting images to superpixels and manually labeling the superpixels. In the Helen dataset, hair was automatically annotated by alpha matting.

In the Helen dataset, we observed relatively lower accuracy for hair, eyebrows

Table 2.5: **Comparison of our results versus the state-of-the-art on the Helen dataset.** All of the results are reported in terms of F_1 score. The overall results exclude the background results and the face skin results.

	◯	⌒	👁	〰	...
Smith et. al (2013) [158]	88.20	72.20	78.50	92.20	...
Liu et. al (2015) [105]	91.20	73.40	76.80	91.20	...
Zhou et. al (2015) [214]	——	81.30	87.40	**95.00**	...
Ours	**94.36**	**82.26**	**88.73**	94.09	...

	⬤	⬯	⌣	👄	👤
Smith et. al (2013) [158]	65.10	71.30	70.00	85.70	80.40
Liu et. al (2015) [105]	60.10	82.40	68.40	84.90	85.40
Zhou et. al (2015) [214]	75.40	83.60	80.90	92.60	87.30
Ours	**79.66**	**85.50**	**86.23**	**92.82**	**90.99**

and upper lips compared to the rest of the classes. The relative low accuracy of hair and eyebrows can be attributed to the fact that these classes do not have well defined boundaries making it difficult to isolate them from background and/or face skin. Similarly, the relatively low accuracy of upper lip can be attributed to the fact that this class has shared borders with four other classes (i.e., face skin, inner mouth and lower lip) and often misclassified as belonging to one of them. However, the discrepancy between upper lip, and inner mouth or lower lip is surprising since these classes have the similar properties with upper lip, but might be explained by class imbalance.

Comparison of results versus state-of-the-art

After the main experiments, we compared the results of the CnnRnnGan model on the Part Labels dataset and the Helen dataset versus the earlier results reported in the literature.

Part Labels Dataset

First, we compared our results on the Part Labels dataset versus the following:

- Restricted Boltzmann machine (RBM) and CRF based image labeling method of Kae et al. (2013) [76].

- CNN, RBM and CRF based semantic part segmentation method of Tsogkas et al. (2015) [178].

- CNN and CRF based face labeling method of Liu et al. (2015) [105].

- Convolutional variational autoencoder based semantic segmentation method of Zheng et al. (2015) [208].

- Convolutional neural fabric based semantic segmentation method of Saxena et al. (2016) [146].

To the best of our knowledge, the CnnRnnGan model achieved state-of-the-art results on the Part Labels dataset (Table 2.4). The best overall results in the literature [105, 178] were improved by 1.55 and 0.19 percentage points (pp) from 95.12 to 96.67 and from 96.97 to 97.16 for pixels and superpixels, respectively.[4] The improvements in the best existing hair results were more pronounced compared to those in the rest of the best existing results (= 6.99 pp versus ≤ 1.81 pp).[5]

Helen Dataset

Second, we compared our results on the Helen dataset versus the following:

- Exemplar based face parsing method of Smith et. al (2013) [158].

- CNN and CRF based face labeling method of Liu et al. (2015) [105].

- CNN based face parsing method of Liu et al. (2015) [214].

To the best of our knowledge, our model also achieved state-of-the-art results on the Helen dataset (Table 2.5). The best overall result in the literature [105] was improved by 3.69 pp, from 87.30 to 90.99. The improvements in the best existing face skin, upper lip and lower lip results were more pronounced compared to those in the rest of the best existing results (≥ 3.16 pp versus ≤ 1.90 pp).

Ablation Experiments

Finally, we conducted two sets of ablation experiments on the Part Labels dataset and the Helen dataset, in which we evaluated the variants of the CnnRnnGan model.

Part Labels Dataset

First, we evaluated the effect of removing the different components of the CnnRnnGan model on the Part Labels dataset (Table 2.6).

---

[4]Note that the CnnRnnGan model was trained on pixels only. The superpixel results were obtained by averaging the corresponding outputs of the CnnRnnGan model. While these results are supoptimal since the CnnRnnGan model was not trained on superpixels, they are reported for completeness.

[5]Note that background, face skin and hair results were reported in [105] only.
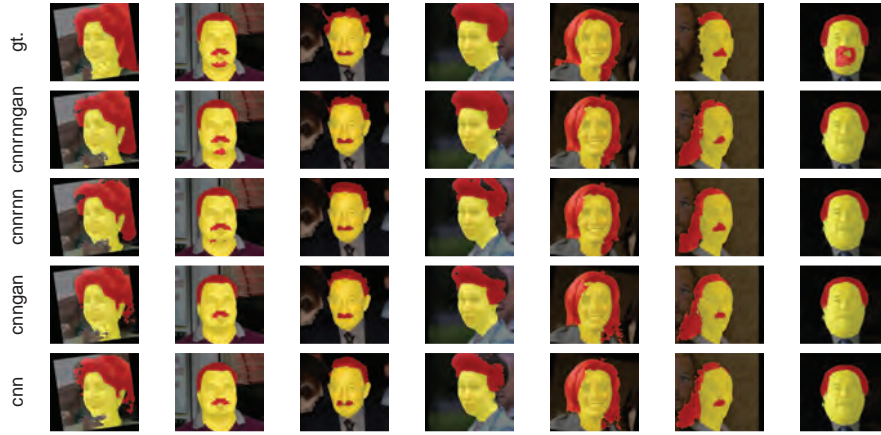
Figure 2.10: **Example segmentations of the variants of the CnnRnnGan model that were evaluated in the ablation experiment on the Part Labels dataset.** The last two columns show failure cases in which none of the model variants achieved satisfactory results. gt. denotes ground-truth.

Table 2.6: **The results of the ablation experiment on the Part Labels dataset.** The results are reported in terms of Jaccard index (i.e. intersection over union) of the classes and their arithmetic mean, respectively.

Cnn	.9617	.9111	.7525	.8751
CnnGan	.9622	.9114	.7574	.8770
CnnRnn	**.9663**	.9177	.7795	.8878
CnnRnnGan	.9656	**.9182**	**.7808**	**.8882**

The CnnRnnGan model achieved the best results except for background. The performance deteriorated by removing the Gan component and keeping the Rnn component (CnnRnn model). Removing the Rnn component and keeping the Gan component (CnnGan model) further deteriorated the results. The results dropped also by removing both the Rnn component and the Gan component (Cnn model). Among all of the classes, the most notable change was observed for hair (= 0.0283).

Fig. 2.10 shows representative examples. In the first column, it can be observed that even though the ground truth had a mistake (the hands were incorrectly labeled

Table 2.7: **The results of the ablation experiment on the Helen dataset.** The results are reported in terms of Jaccard index (i.e. intersection over union) of the classes and their arithmetic mean, respectively.

	[ ]	◯	⌢	👁	⌣⌣	▬	⌒	⌣	◖	◗
init.	.8253	.6358	.4855	.6527	.5325	.5568	.5757	.6001		.5405
1c.	**.9465**	.8770	.6074	.6811	.8562	.5666	.6655	.6667	**.7030**	.7300
init.+1c.	.9408	8805	.6189	.6880	.8618	.5724	.6804	.6738	.6717	.7320
init.+5c.	.9452	**.8933**	**.6987**	**.7974**	**.8884**	**.6619**	**.7467**	**.7580**	.6962	**.7873**

as face skin), particularly the CnnRnn and CnnRnnGan models correctly segmented most pixels. The example in the second column demonstrates the performance of the models in a difficult facial hair case. In this example, all models performed well in segmenting the mustache, but only CnnRnnGan model correctly identified the beard pixels. The third column showcases an example that all models performed well. The examples in the fourth and fifth columns highlight the gradual improvement provided by each additional model component in the correct classification of hair pixels. Especially in the example in the fifth column, it is possible to observe the improvements in the identification of fine details of hair. The first failure case example in column six demonstrates a difficult case for all models. The pixels to the left of the face skin are indeed hair pixels, however they belong to another person in the photograph. All models failed to make this distinction. The last failure case example shows that all models failed to segment the facial hair pixels and incorrectly labeled them as facial skin pixels. This error could be attributed to the low contrast difference between the face skin and facial hair pixels.

Helen Dataset

Subsequently, we evaluated the effect of conditioning the CnnRnnGan model on the initial segmentation and/or training multiple CnnRnnGan models for segmenting different classes on the Helen dataset (Table 2.7). The initial segmentation (init. model) failed to achieve competitive results. These results were considerably improved by training a single CnnRnnGan model for segmenting all of the classes (1c. model). Conditioning the single CnnRnnGan model on the initial segmentation (init.+1c. model) slightly enhanced the performance. The results were once again considerably improved by training multiple CnnRnnGan models for segmenting different classes and conditioning them on the initial segmentation (init.+5c. model), which made them the best for all of the classes except for background and hair. Among all of the classes, the most notable improvements were observed for eyebrows, eyes, upper lip, inner mouth and lower lip ($\in [0.1051, 0.2132]$).

41
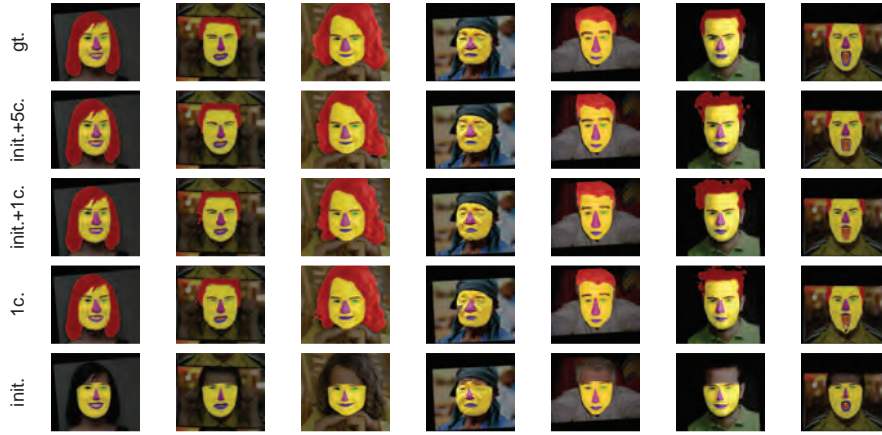
Figure 2.11: **Example segmentations of the variants of the CnnRnnGan model that were evaluated in the ablation experiment on the Helen dataset.** The last two columns show failure cases in which none of the model variants achieved satisfactory results. gt. denotes ground-truth.

We illustrate qualitative examples in Fig. 2.11. In the first five columns of this figure, it is possible to see an increase in performance starting from the simplest initial segmentation model to the complex variants of the CnnRnnGan model. While the initial segmentation does a good job in determining the general locations of each face region, it does not provide a detailed solution. Furthermore, it can be observed that the initial segmentation performs rather poorly in the nose and eyebrow regions, and whenever the expression of the face diverges from a neutral pose in the mouth regions. Among the variants of the CnnRnnGan model, the qualitative differences were minimal. However, the improvement provided by training multiple CnnRnnGan models for segmenting different classes and conditioning them on the initial segmentation (i.e. init.+5c) has resulted in visually distinguishable accuracy differences. This model was able to capture the details better than the remaining two model variants. The last two columns in the figure demonstrate failure cases where all model variants had errors. Models performed poorly in distinguishing hair from background when the background color was similar to the hair color (column 6) and in identifying the mouth regions when the person in the photograph had an extreme facial expression (column 7).

3 3D Hand Pose Recovery

Recently, hand pose recovery attracted special attention thanks to the availability of low cost depth cameras, like Microsoft Kinect [28, 69, 81, 106, 126, 136, 159, 162, 164, 165]. Unsurprisingly, 3D hand pose estimation plays an important role in most HCI application scenarios, like social robotics and virtual immersive environments. Despite impressive pose estimation improvements, 3D hand pose recovery still faces some challenges before becoming fully operational in uncontrolled environments with fast hand/fingers motion, self occlusions, noise, and low resolution [180]. Besides, available datasets mainly provide front-face hand deformations, which are not suitable to compare state-of-the-art approaches against hard cases with large occlusions. Also, to the best of our knowledge, little attention has been paid to incorporate temporal motion information in hand pose recovery problems. As an example, Oikonomidis *et al.*[127] only initialized the model using previous frame.

In this chapter, we consider 3D hand pose recovery in depth images. We mainly propose two solutions for this problem. The first solution combines both spatial and temporal information in a generative approach, while the second solution exploits CNNs as discriminative pose regressor. In both solutions, as in [162], we break the hand pose estimation problem into hierarchical optimization subtasks, each one focused on a specific finger and hand region, while reducing the search space. In the first approach, we present a system for efficient hand pose recovery in non-controlled settings, involving self-occlusions, based on current trends towards minimizing pose parameters in the space of nearest candidates [150, 195]. Consequently, motivated by [213], our estimated joints are applied in a sequence of frames to minimize parameters of a trained bilinear model [4] consisting of shape and trajectory bases. This process further refines the estimation of occluded parts.

In the second solution, a specific tree-shaped CNN architecture is designed allowing local finer specializations for different fingers and hand regions than a global pose. In addition, we model correlated motion among fingers by fusing the features, learned in the hierarchy, through fully connected layers and training the whole network in an end-to-end fashion. The main advantage of this strategy is that the 3D hand pose prediction problem is attained as a global learning task based on

local estimations. In order to improve the final estimation in such high non-linear space of hand configurations, we incorporate appearance and physical penalties in the loss function.

3.1 Related Work

Hand pose estimation has been extensively studied in literature [42], we refer the reader to [69, 150] for a complete classification of earlier works in the field.

Two main strategies have been proposed in the literature for addressing the aforementioned challenges. Model-based generative approaches, whose strategy essentially lies on fitting a predefined 3D hand model to the depth image [32, 115, 127, 136, 150, 164]. However, as a many-to-one problem, designing a global objective function and accurate initialization of model parameters are critical, and due to the fast motion and non-rigid nature of hands, together with finger self-occlusions, it is still a challenge for single-hand trackers to correctly maintain the state of an animated 3D hand model over time.

Temporal information and trajectory analysis, besides the shape itself, provide useful information to analyze shape and recover occluded parts. Works from structure from motion, such as matrix imputation [154], statistical model analysis and non-rigid structure from motion [130, 167], showed the benefits of using temporal information for shape analysis. Zhou *et al.* [213] proposed a spatio-temporal model for the problem of human pose recovery. Although their approach obtains promising results, the complexity of the minimization problem makes it not applicable for all types of pose deformations.

Alternatively to model based techniques, the so-called data-driven approaches consider the available training data to directly learn hand pose from appearance [69, 81, 86, 162, 165, 195]. Contrary to using hand trackers, which lead to model drift over time, single-frame detection methods are initialized at each frame, thus recovering more easily from estimation errors [150]. Multiple procedures based on Random Forests (RF) have emerged consisting of Hough Forests [195], Random Decision Forests [82] and Latent Regression Forests [165], as detailed in [69]. Unfortunately, the number of occluded joints is commonly bigger in hands than in human bodies. As a result, techniques based on RF usually require huge training sets, and some kind of viewpoint estimation is needed in order to improve performance [166]. Some data-driven works analyze the hand in the space of nearest shapes in order to reduce the search space [150] or approximate unknown pose parameters through matrix factorization [28]. Data-driven hand pose approaches have been benefited from recent advances on CNNs as well. Yet, finding features with suitable generalization and discrimination properties in highly nonlinear spaces is a challenging task.

Most CNN-based architectures in data-driven hand pose estimation approaches are specifically designed to be discriminative and generalizable. Although the success of such approaches depends on the availability of enough training data, CNN models cope reasonably well with a highly nonlinear output space. In order to deal with this problem, two main kind of approaches can be distinguished in the literature, namely heat-map and direct regression methods.

Heat-map approaches estimate likelihoods of joints for each pixel as a pre-processing step. In [172], a CNN is fed with multi resolution input images and one heat-map per joint is generated. Subsequently, an inverse kinematic model is applied on such heat-maps to recover the hand pose. Nevertheless, this approach is prone to propagate errors when mapping to the original image, and estimated joints may not correlate with the hand physics constraints. The work of [106] extends this strategy by applying multi-view fusion of extracted heat-maps, where 3D joints are recovered from only three different viewpoints. In this approach, erroneous heat-maps are expected to be improved in the fusion step using complementary viewpoints. The key idea in this work is to reduce the complexity of input data by aligning all data with respect to the hand point cloud eigenvectors. For most heat-map based approaches, however, an end-to-end solution can be only achieved by considerably increasing the complexity of the model, e.g. introducing a cascading approach [190]. Although such approaches used to work well for 2D pose estimation in RGB images, they do not necessarily are able to model occluded joints from complex hand poses in depth data.

As an alternative, a number of works propose direct regression for estimating the joint positions of the 3D hand pose based on image features [125, 126, 162]. As mentioned in [171], contrary to heat-map based methods, hand pose regression can better handle the increase in complexity of modeling highly nonlinear spaces. Although some approaches propose Principle Component Analysis (PCA) to reduce the pose space [106, 125], in general such linear methods typically fail when dealing with large pose and appearance variabilities produced by different viewpoints.

Recently, error feedback [126] and cascading [162] approaches have proven to avoid local minima by iterative error reduction. Authors in [126] propose to train a generative network of depth images by iteratively improving an initial guess. Also, the method proposed in [162] divides the global hand pose problem into local estimations of palm pose and finger poses. Thus, finger locations can be updated at each iteration relative to the hand palm. Contrary to our method, the authors use a cascade of classifiers to combine such local estimations.

Authors in [157] apply a CNN to make use of the resulting feature maps as the descriptors for computing k-nearest shapes. Similarly to our approach, in their method the CNN separates palm and fingers and computes the final descriptor by dimensionality reduction. Differently to our approach, they factorize the feature

45
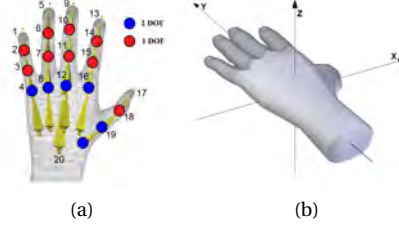
Figure 3.1: Hand models. a) DOF for different joints. Joints are indexed by assigned numbers. This figure also shows how skeleton is fitted inside hand. b) Palm coordinate system. Finger parameters are computed based on this coordinate system.

vectors and nearest neighbors hyper-parameters to estimate the hand pose. In a different way, we propose training the network by fusing local features to avoid non-accurate local solutions, without the need of introducing cascading strategies nor multi-view set-ups.

3.2 Generative 3D hand pose recovery

The basic idea of the proposed method is to recover a hand pose through a combination of part-based model fitting and data-driven approaches in a single frame and, afterward, refine occluded joints in a sequence. As illustrated in Fig. 3.2, we first segment hand and recover palm joints (introduced in Chap. 2). Given the palm joints and segmented fingers, we extract a number of candidates for each finger using a set of predefined examples. We then send these candidates to the optimization process to minimize an objective function which fits a finger model to the segmented finger (Sec. 3.2.1). We minimize the parameters of each finger separately. Finally, occluded joints are refined by solving the coefficients of the trained bilinear model in a sequence of F images (clip). We cluster clips in order to reduce non-linearity (Sec. 3.2.2).

3.2.1 Pose estimation

Given the nearest shapes and their corresponding joint locations, one could minimize coefficients of a weighted sum of basis models (like PCA) to extract hand pose. However we observed that this process does not perform well in practice. Instead, we divide the problem of pose estimation into two sub-problems: palm pose estimation, as global hand pose, and fingers pose estimation. Each problem is

Figure 3.2: Diagram of the proposed method. In the first step, a single-frame hand pose is estimated. First palm joints and finger segments are recovered through nearest shapes. Then finger models are fitted using extracted candidates. In the second step, temporal data is incorporated to refine first step estimation.

solved separately. In the model, palm pose is first detected. We assume palm is rigid and refer to palm pose as a composition of wrist and base joints of all fingers except the thumb in 3D space (i.e. joints 4, 8, 12, 16 and 20 in Fig. 3.1a). Sun *et al.*[162] regress palm pose by iterative refinement of an initial pose. Sharp *et al.*[150] estimate a global view point and iteratively fit a model by generating some hypothesis candidates. It has been shown that NN-based approaches perform well in practice [69]. In this work we rely on extracted nearest shapes to both estimate palm pose and segment the hand. This approach explained in Chap. 2.

To get fingers poses, we fit a simple finger model for each finger separately based on segmented pixels of that finger, and thus an incorrectly segmented finger instantly causes a failure pose. Each finger model S is composed of three cylinders and half-spheres except for the thumb, which is composed of an ellipsoid, two cylinders and three half-spheres. Finger model parameters are computed based on the palm coordinate system (see Fig. 3.1).

Given hypothesis parameters h, camera calibration parameters, palm pose and

finger properties like length and diameter of bones, we can render a 3D model of the finger S and project it onto the image plane. Let I_M, M_M and M_F be the depth image of the projected finger model, the projected finger model mask and the segmented finger extracted from Sec. 2.1.3, respectively. Then, we set the background of I_M to zero and define $M_{in} = M_F \wedge M_M$ and $M_{out} = \neg M_F \wedge M_M$ (see Fig. 3.3). The goal is to find hypothesis parameters h that best fit the model to the finger in query. Therefore we define the objective function $E(h, I)$ to compute the amount of discrepancy between I_M and I with respect to M_F through:

$$E_1 = 1 - \frac{\#M_{in}}{\#M_F + \epsilon},$$
(3.1)

$$E_2 = \begin{cases} 10 & \text{if } M_{in} \subset \varnothing, \\ E_1 \frac{mean(min(|I_M(M_{in}) - I(M_{in})|, \lambda))}{\lambda} & \text{if } M_{in} \not\subset \varnothing, \end{cases}$$
(3.2)

$$E_3 = \frac{\#(I_M(M_{out}) < (I(M_{out}) + \tau))}{\#(M_{out}) + \epsilon},$$
(3.3)

$$E(h, I) = w_1 E_1 + w_2 E_2 + w_3 E_3,$$
(3.4)

where λ and τ are some depth difference thresholds. Term E_1 computes overlapping area between M_M and M_F normalized by $\#M_F$. Term E_2 controls the mismatching of depth in the overlapping area M_{in}. Such a mismatching depth energy is directly related to $\#M_{in}$. We consider this situation in the first case of Fig. 3.3. A small area M_{in} can generate a lower depth mismatching energy which can cause a wrong matching. Therefore we scale E_2 by multiplying it to E_1 as a function of $\#M_{in}$ to reduce the effect of $\#M_{in}$ in the depth mismatching energy. We add term E_3 to avoid finger collision to non overlapping pixels M_{out}. We consider this situation in the second and third cases of Fig. 3.3. We add the term ϵ to avoid division by zero and set it to a low value. The number 10 in E_2 is a maximum energy, and w_1, w_2, and w_3 are some fixed weights.

Particle swarm optimization (PSO) is a commonly used approach to minimize such an objective function. However, it is not efficient for minimizing over all possible parameters and it is easily trapped to local minima [80]. In order to cope with such problems, we predefine a low number (300 in our case) of sample fingers which cover most finger poses and evaluate a simple function over all predefined samples to select the best candidates. We use simple facts to design this evaluation
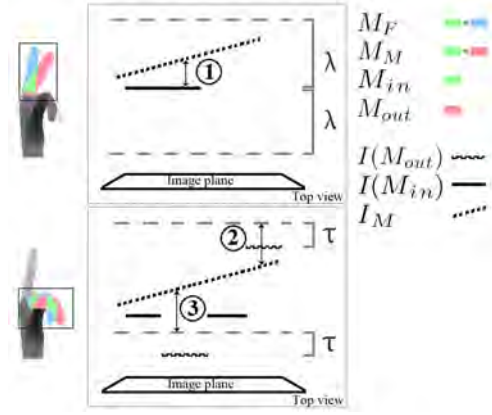
Figure 3.3: Objective function E. We jointly maximize overlapping area M_{in} (E_1) and minimize depth discrepancy between generated model and hand finger (E_2). We show the overlapping area (green) can have a relation to the depth difference. A small depth difference may not guarantee a good matching, and therefore, we penalize it by multiplying it to the normalized non-overlapping area (blue). Hence a small depth difference is only useful if blue area is small as well. In the second and third cases we should avoid collision between the model surface I_M and other finger surfaces available in M_{out} (E_3). τ controls the area between fingers.

function. As the first rule, all finger joints should be located in the hand mask after projecting them onto the image plane. Secondly, the joints should have at least a depth equal to the hand surface depth plus a threshold. Let $J_f^{xyz} \in \mathbb{R}^{3 \times f_N}$ be the matrix of 3D locations of the joints belonging to the finger f and $J_f^{uv} \in \mathbb{R}^{2 \times f_N}$ be the matrix of 2D locations of the joints of finger f after projecting onto the image plane where f_N is the number of joints. Therefore all joints should meet the constraint $I(J_f^{uv}) + \omega \leq J_f^z$, where ω is a constant value. Since we set the background of I to a high value, this constraint satisfies the first rule as well. We consider a third rule for visible fingers such that the joints should not be far from the finger point cloud. We formulate these rules for finger f as:

$$C_{fd} = \{I(J_{fi}^{uv}) - J_{fi}^z + \omega\}, i \in 1, ..., f_N, \tag{3.5}$$

49

$$C_f = \begin{cases} C_{fd} & \text{if } M_F \subset \varnothing, \\ \{C_{fd}, \gamma \| \overline{I^{xyz}(M_F)} - \overline{J_f^{xyz}} \|\} & \text{if } M_F \not\subset \varnothing, \end{cases} \tag{3.6}$$

$$Err(C_f) = \sum_{\{c|c \in C_f \wedge c \geq 0\}} min(c, \varphi), \tag{3.7}$$

where $\overline{I^{xyz}(M_F)}$ is the center of finger point cloud and $\overline{J_f^{xyz}}$ is the center of the candidate joints. ω is a depth threshold that controls the distance of the joints to the hand surface. γ is a weight to balance different terms. Eq. 3.6 is treated as a constrained inequality and therefore negative values are desirable. As a consequence we sum over positive costs limited by constant threshold φ Eq. 3.7 to evaluate each sample finger. Finally a number of samples with the lowest error are selected as candidates and feed into PSO. We set the number of generations and population size to 5 and 30, respectively. For completely occluded fingers (i.e. $M_F \subset \varnothing$) we apply Eq. 3.6 and make an average finger from outcomes. All the thresholds and weight terms are experimentally set to some fixed values as follows: $\tau = 15$, $\lambda = 25$, $w_1 = 0.25$, $w_2 = 0.65$ and $w_3 = 0.1$, $\omega = 8$, $\gamma = 4$ and $\varphi = 50$.

3.2.2   Spatio-temporal pose recovery

Time-varying spatial data is involved in a vast range of computer vision applications [167, 194] and proved to be useful in extracting missing data. Spatial correlation or trajectory analysis of independent points solely fails to model all information in spatio-temporal data. Akhter *et al.* [4] combined two linear shape and trajectory bases learned by discrete cosine transform and SVD to exploit spatio-temporal regularities. We follow this work to generate linear bases of hand data. To train bilinear bases, we have generated a dataset including smooth deformation of fingers in a reference view in a sequence. The advantage of keeping a reference view is that all the frames are previously aligned by their palm joints. Then we extract fixed-length clips by a sliding window over the sequences. A clip is represented by $Q \in \mathbb{R}^{F \times 5D}$ where F is the number of frames and D is the number of parameters for each finger. Clip Q can be factorized by TCB^T (as introduced in [4]) where $T \in \mathbb{R}^{F \times k_t}$ and $B \in \mathbb{R}^{5D \times k_s}$ are learned trajectory and shape structures and $C \in \mathbb{R}^{k_t \times k_s}$ is the coefficient matrix. Given the learned T and B, the goal is to minimize a function over coefficients C in order to extract clip Q at test time.

A common problem with linear basis models like PCA and SVD is that they are sensitive to the correlation coefficient or distribution of the data. A solution is to

divide the space of clips (e.g. clustering) in order to provide more correlation among data. However, this solution is not exact. In, [213] authors search over all clusters to find best models. However, this is not suitable for a huge number of clusters, as in our case. In order to cope with previous issues, we propose a fast and approximate solution to find best models.

In the training step, we apply k-means to cluster data. We regenerate each cluster by extracting νN nearest clips to the cluster centroid where N is the number of clips in the cluster and $\nu > 1$. In fact, we extend each cluster with overlapping to its adjacent clusters. Afterwards, we train bilinear models T and B on each cluster (as described in [4]). This causes the models to be more robust at cluster boundaries.

At test time, given the last clip Q (initialized using Sec. 3.2.1) and parameters visibility $V \in \{0, 1\}^{F \times 5D}$ (extracted from RF), we are able to find nearest clips in a dataset by a trained kd-tree. However, visible and invisible joints have the same weight in the clips and possible errors in the initial estimation can cause a false nearest cluster. More specifically, the task is to find a cluster that best describes both the appearance and occluded parts, and then minimize a function on coefficients C. Therefore we define the objective function $STC(Q, V, T, B, \mu, \sigma)$ as:

$$STC = \sum_{f=1}^{F} \sum_{i=1}^{5D} V_{fi} |Q_{fi} - Q_{fi}^r| + \beta \sum_{f=1}^{F-1} \Psi^{f,f+1}, \tag{3.8}$$

where Q_{fi} extracts the i-th parameter in frame f, $Q^r = TCB^T$ denotes reconstructed parameters through coefficients C, Ψ is a smoothness function among correspondent parameters in frames f and $f + 1$, and β is a regularization weight. We define the smoothness function as:

$$\Psi^{f,f+1} = \sum_{i=1}^{5D} \neg(V_{f,i} \wedge V_{f+1,i}) \left| \frac{Q_{f,i}^r - Q_{f+1,i}^r - \mu_{fi}}{\sigma_{fi}} \right|, \tag{3.9}$$

where μ_{fi} and σ_{fi} are precomputed mean and standard deviation distance for i-th parameter in the frame f for each cluster, respectively. The first term in Eq. 3.8 denotes the appearance cost and the second term penalizes large movements of the occluded joints.

We approximate the best cluster by first extracting a number of nearest clusters, traversing a trained kd-tree using clip Q. This kd-tree is trained based on clusters centroids. Subsequently, we generate a number of random poses around clip Q and evaluate function STC on them for each extracted nearest cluster. Finally, we take that cluster which generates minimum average error.

Efficient minimization of Eq. 3.8 is required. Levenberg-Marquardt algorithm

is a standard minimization technique, although finding a good initial point to minimize Eq. 3.8 makes the problem intractable. In order to overcome this problem, we use PSO with a number of randomly selected particles around Q and apply $T^T R (B^T)^{-1}$ for all random clips to generate initial particles, where R is a random clip and T and B are trained bliniear structures of the best cluster. To have a fair distribution of fingers and removing undesired clips, we apply Eq. 3.7 on all fingers for all random clips and select a subset of best candidates by sorting clips regarding their maximum finger error. As a consequence, the solution is achieved in a few generations. We set the number of generations and population size to 5 and 100, respectively.

We use finger parameters in all frames as a trajectory descriptor which is invariant to finger length and hand shape. Finger parameters have an advantage versus the 3D joints locations since we have more control on them, like adding constraints or generating a more meaningful shape without adding extra regularization. Given that this process mainly improves occlusion recovery, we combine the recovered invisible joints to the visible joints estimated in the initial step as the final pose. In the experiments, we show that initial pose estimation has a low error which is reliable enough to be used in the occlusion refinement process. We apply full rank matrices to train the bilinear model, with $k_s = 7$, $k_t = 7$, clip length $F = 7$ frames and $\beta = 0.1$.

3.2.3 Results

Evaluation metric We used the 3D Euclidean distance from joints to groundtruth for evaluating the different approaches. We also measured the success rate as in [162] to compute percentage of each error threshold.

Evaluation on the synthetic dataset (Chap. 4) For comparison, we used as the baseline a transformed average shape from the nearest neighbors according to our shape descriptor and ICP. We compared PSO vs. greedy for single-frame pose recovery as well. In the greedy approach after applying our population selection proposal (Eq. 3.7), the best candidate was selected by evaluating Eq. 3.4. We include our occlusion refinement approach and we show how it can be combined with greedy to slightly improve occluded joints. Fig. 3.4a shows the per-joint average error (mm) for different approaches. As it can be expected PSO performs slightly better than the greedy approach. However, the difference is not significant and the greedy approach runs faster than PSO. Joints belonging to the palm exhibit accurate palm pose recovery even in quite difficult poses which is quite critical for recovering the pose of individual fingers. Notice that the baseline is the most accurate approach for the thumb joints. A possible explanation is that the thumb has higher movement range than other fingers and it is thus hard to recover with model-based approaches.

(a)

(b)

(c)
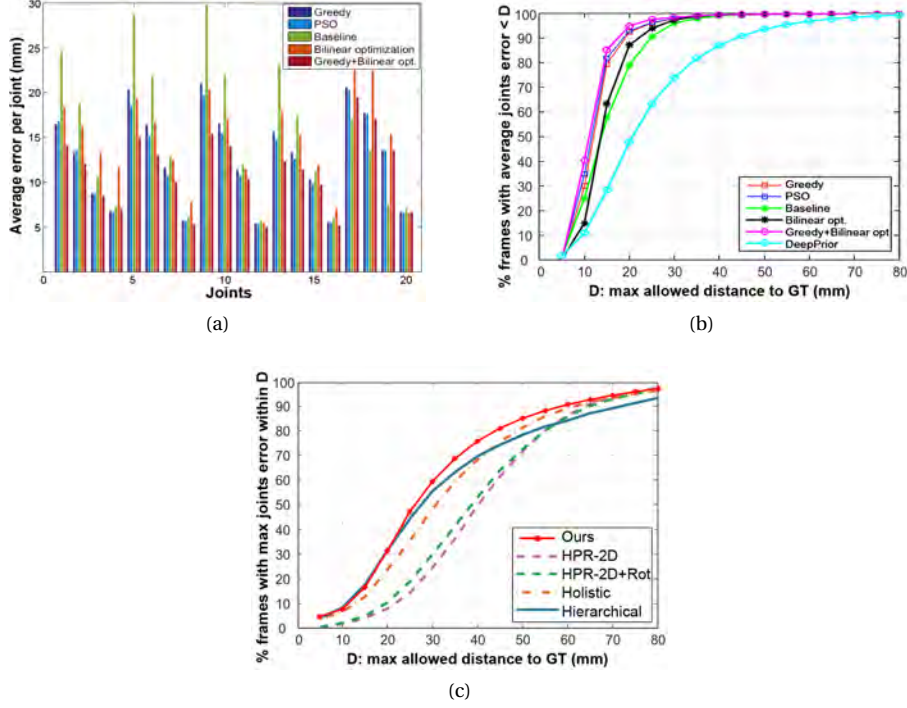
Figure 3.4: a) Error per joint. Joint arrangement is shown in Fig. 3.1a. The mean errors are 12.86, 12.40, 15.16, 14.72 and 11.09, respectively. b) Success rate over different error thresholds on our dataset comparing to DeepPrior ([125]), and c) success rate on the MSRA dataset. Note that we took the state-of-the-art results instantly from [162]. See [162] for details on the methods.

Bilinear optimization solely does not improve the overall error and resulted in lower accuracy than single-frame techniques, but when combined with the greedy solution we could improve occluded parts poses by 3.7mm (i.e. visible joints from greedy and occluded ones from bilinear optimization). Although this is not a big improvement, the results show the benefits of incorporating temporal data. However, increasing the number of frames within each clip adds complexity to the bilinear coefficient optimization and precludes real-time performance.

For the current version of the system, the hand can not be occluded by any other object. Since we use ICP and QDA, model drifts might occur when the number of
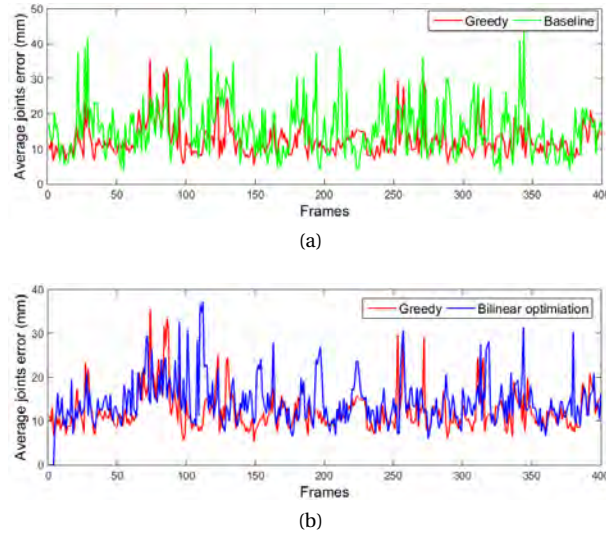
Figure 3.5: a) Greedy vs. baseline. b) Greedy vs. bilinear optimization.

visible pixels from the hand is dramatically reduced (due to pose, viewpoint, camera noise, or missing data). Not availability of nearest shapes does also influence the pose recovery process for both hand segmentation and palm pose recovery tasks.

We also compared our proposal with the DeepPrior ([125]) Convolutional Neural Network approach. Fig. 3.4b illustrates the success rate error among proposed methods and DeepPrior. DeepPrior shows the lowest accuracy. This could be because of the high pose variability and presence of occlusions [125]. We trained DeepPrior with 300K samples, 200 epochs and learning rate 0.001. We also show some qualitative results in Fig. 3.6a and 3.6b.

We compared greedy vs. baseline and greedy vs. bilinear optimization for some examples in Fig. 3.5a and 3.5b. The purpose of these graphs is to compare how different methods behave in a sequence of frames.

Evaluations on MSRA dataset Without accurate hand segments, we were not able to properly evaluate our approach on this dataset. However we used inaccurate hand segments to setup our baseline method on this dataset. To report results and compare to the state-of-the-art on this dataset we applied a 9-fold cross validation, where each fold corresponds to one subject. Fig. 3.4c illustrates success rate of our baseline approach comparing to [162]. Table 3.1 shows per-joint average error in comparison to state-of-the-art approaches. Notice that our baseline method clearly

(a)

(b)
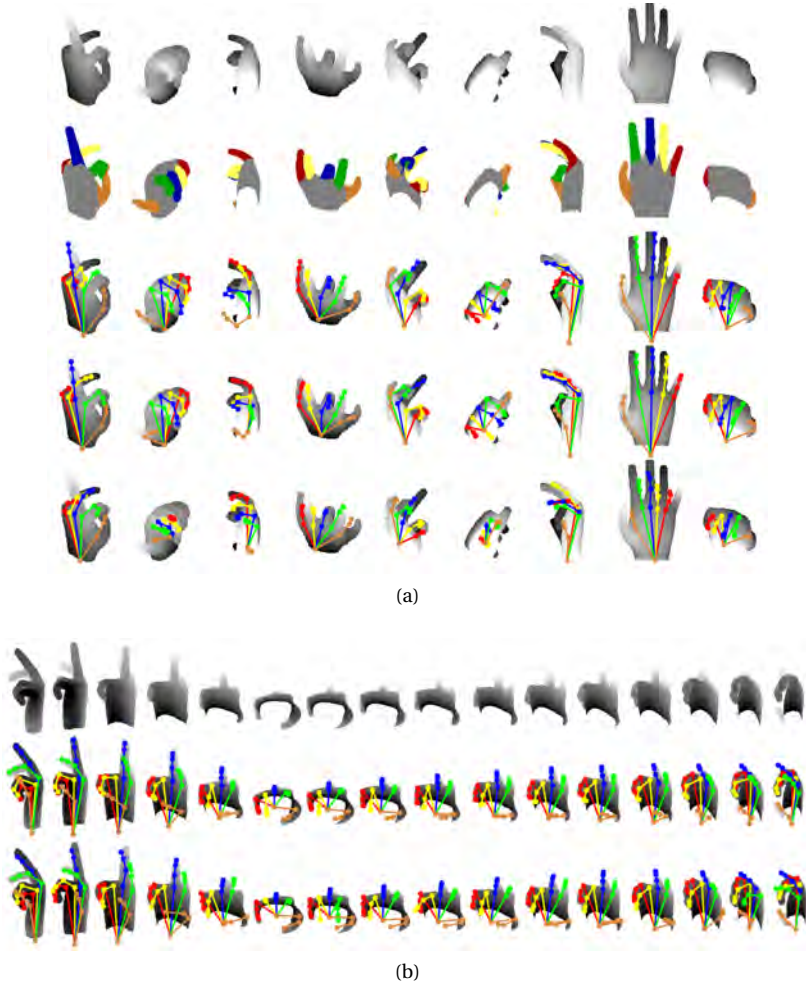
Figure 3.6: Qualitative results on our dataset. a) Comparing different approaches. Columns from top to bottom: depth images, segmentation, baseline, greedy, and DeepPrior [125]. The rows show the frames 1, 12, 27, 42, 75, 83, 244, 301 and 352 from left to right. b) Greedy+bilinear optimization in depth video. Columns from top to bottom: depth images, ground truth, and final results. Results are generated with error lower than 30mm per visible joint for initial step.

Table 3.1: Quantitative results on MSRA dataset. Values are per-joint error in millimeters. Letters R and T go for finger root and finger tail, respectively. We extracted values from the results reported in the papers. Results for [127] obtained from [28].

	IndexR	IndexT	MiddleR	MiddleT	RingR	RingT	LittleR	LittleT	ThumbT	Mean
Oikonomidis et al. [127]	31.0	56.0	32.9	56.0	32.9	49.3	35.1	53.7	22.2	38.2
Choi et al. [28]	22.6	43.5	24.0	44.9	23.1	43.1	21.8	39.5	31.1	29.8
Ge et al. [106]	11.5	16.0	9.0	15.6	9.9	15.1	13.2	16.0	16.7	13.0
Ours (KNN+ICP)	9.5	17.3	7.7	17.1	8.3	15.5	10.6	17.7	14.8	12.8
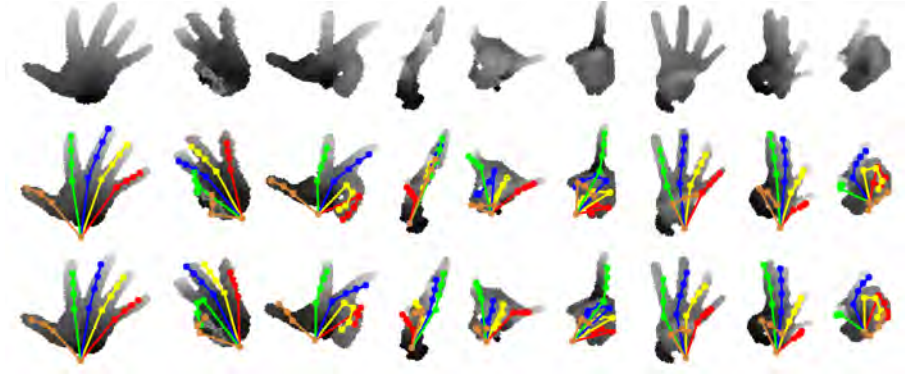
Figure 3.7: Qualitative results on MSRA dataset [162]. Columns from top to bottom: depth images, groundtruth, and our estimation.

outperforms most of the state-of-the-art approaches on this dataset. These results show the robustness and accuracy of our methodology against highly variable poses. Fig. 3.7 shows some qualitative results on this dataset.

Time complexity Our methodology has a high parallelization capability at any stage. It is GPU-friendly since fingers estimations are minimized separately.

Greedy finger minimization needs just evaluating function E over the selected candidates. Initial pose estimation is achieved in real time. Most of the processing time is consumed by PSO optimization over bilinear model coefficients C. We use 5 generations over 100 particles which is comparable to 30 and 100 in [150] respectively. We implemented the whole pipeline in Matlab and C++, which although not optimized, runs at 10 fps.

3.3 CNN based hand pose regression

Given an input depth image $\mathcal{I}$, we refer the 3D locations of n hand joints as the set $J = \{j \in \mathbb{R}^3\}_1^n$. We denote j^{xyz} and j^{uvz} as a given joint in the world coordinate system and after projecting it to the image plane, respectively. We define $n = 20$ for the wrist, finger joints and finger tips, following the hand model defined in [162]. We assume a hand is initially visible in the depth image, i.e. not occluded by other objects in the scene, although may present self-occlusions, and properly been detected beforehand (i.e. pixels belonging to the hand are already segmented [172]). We also assume intrinsic camera parameters are available. We refer to global pose as the whole set J, while, a local pose is a subset of J (e.g. index finger joints).

Considering hand pose recovery as a regression problem with the estimated pose as output, we propose a CNN-based tree-shaped architecture, starting from the whole hand depth image and subsequently branching the CNN blocks until each local pose. We show the main components of the proposed approach in Fig. 3.8. In such a design, each network branch is specialized in each local pose, and related local poses share features in the earlier network layers. Indeed, we break global pose into a number of overlapping local poses and solve such simpler problems by reducing the nonlinearity of global pose. However, since local solutions can be easily trapped into local minima, we incorporate higher order dependencies among all joints by fusing the last convolutional layer features of each branch and train the network for global and local poses jointly. We cover this idea in Sec. 3.3.1. We also apply constraints based on appearance and dynamics of hand as a new effective loss function which is more robust against overfitting than simple $L2$ loss while providing a better generalization. This is explained in Sec. 3.3.2.

3.3.1 Hand pose estimation architecture

In CNNs, generally, each filter extracts a feature from a previous layer, and by increasing the number of layers, a network is able to encode different inputs by growing the Field of View (FoV). During training, features are learned to be activated through a nonlinear function, for instance using Rectified Linear units (ReLU). The complexity and number of training data has a direct relation to the number of filters, layers or complexity of the architecture: an enormous number of filters or layers might cause overfitting, while a low number might lead to slow convergence and poor recognition rates. Interestingly, different architectures have been proposed to cope with these issues [126, 171, 190]. For example, in multi-task learning, different branching strategies are typically applied to solve subproblems [39, 45], and the different subproblems are solved jointly by sharing features. Similarly, we divide global hand pose into simpler local poses (i.e. palm and fingers) and solve each

57
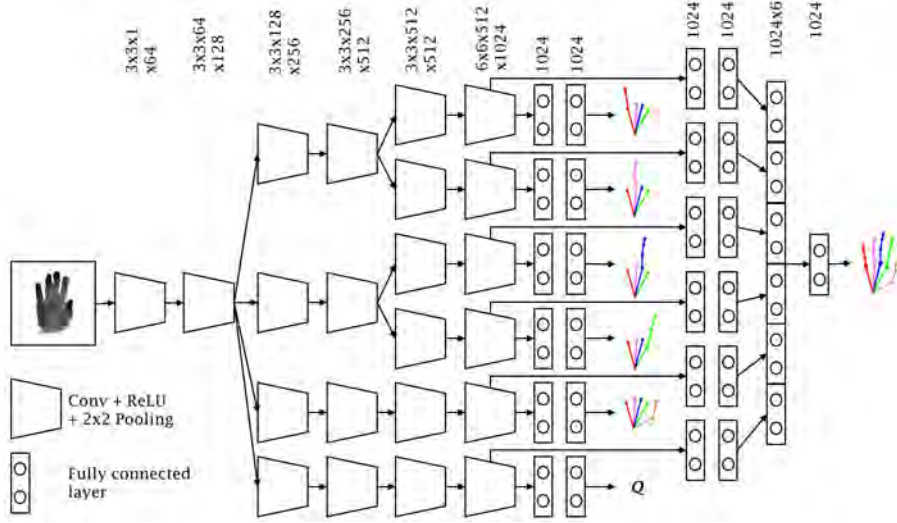
Figure 3.8: Proposed network architecture. Branching strategy connects CNN blocks into a tree-shape structure while regressing local pose at each branch. Each local pose is a 24 dimensional vector. We also include a viewpoint regressor in th network as a rotation matrix in terms of quaternions Q at the output. We then fuse all the features of the last convolutional layers to estimate output global pose. We use Q features in the fusion to extract palm joints more accurate.

local pose separately in a branch by means of a tree-shaped network. We show this architecture in Fig. 3.8.

The proposed architecture has several advantages. Firstly, most correlated fingers share features in earlier layers. By doing this, we allow the network to hierarchically learn more specific features for each finger with respect to its most correlated fingers. Secondly, the number of filters per finger can be adaptively determined. Thirdly, the estimation of the global pose is reduced to the estimations of simpler local poses, resulting the network to train at fast convergence rates.

We define the amount of locality by the number of joints contributing to a local pose. Keeping such locality high (i.e. lower number of joints), in one hand, causes fingers to be easily confused among each other, or detected in a physically impossible location. A low locality value (i.e. higher number of joints), on the other hand, increases the complexity. Besides, local joints should share a similar motion pattern to keep lower complexity. So in the particular implementation in this paper, we assign to each local pose one finger plus palm joints, thus leading to

a 24 dimensional vector.

Training the network only based on local poses omits information about inter-fingers relations. Tompson *et al.* [173] included a graphical model within the training process to formulate joints relationships. Li *et al.* [99] used a dot product to compute similarities for embedded spaces of a given pose and an estimated one in a structural learning strategy. Instead, we apply late fusion based on local features, thus, let the network learn the joint dependencies through fully connected layers for estimating the final global pose. The whole network is trained end-to-end jointly for all global and local poses given a constrained loss function.

Network details Input images are pre-processed with a fix-sized cube centered on the hand point cloud and projected into the image plane. Subsequently, the resulting window is cropped and resized to a 192×192 fixed size image using nearest neighbor interpolation, with zero-mean depth.

As intermediate layers, the network is composed of six *branches*, where each branch is associated to specific fingers as follows: two branches for index and middle fingers, two branches for ring and pinky fingers, one branch for thumb, and one branch for palm. For the palm branch, instead of performing direct regression on palm joints, we make regression on the palm viewpoint, defined as the rotation (in terms of quaternions) between the global reference view and the palm view. As shown in the experimental results, more accurate and reliable optimization is then achieved, since the network is able to model interpolations among different views.

As shown in Fig. 3.8, each convolutional block consists of a convolution layer with 3×3 filter kernels and a ReLU followed by a max-pooling, except for the last block. All pooling layers contain a 2×2 window. The last block contains a convolutional layer with 6×6 filter kernels, providing a feature vector. Fully connected layers are added to the end of each branch for both local and global pose learning. For local pose at each branch there are two hidden layers with 1024 neurons with a dropout layer in between. Similarly, for global pose at each branch, the feature vector is followed by two hidden layers with 1024 neurons with a dropout layer in between. Then, the last hidden layers are concatenated and followed by a dropout and a hidden layer with 1024 neurons. Finally, the global and local output layers provide the estimation of joints with one neuron per joint and dimension.

3.3.2 Constraints as loss function

In regression problems, the goal is to optimize parameters such that a loss function between the estimated values of the network and the ground-truth value is getting minimized. Usually, in the training procedure, an $L2$ loss function plus a regularization term is optimized. However, it is generally known that, in an un-balanced dataset with availability of outliers, $L2$ norm minimization can result in

poor generalization and sensitivity to outliers where equal weights are given to the training data [14]. Weight regularization is commonly used in deep learning as a way to avoid overfitting. However, it does not guarantee the weight updating to bypass the local minima. Besides, a high weight decay causes low convergence rates. Belagiannis *et al.* [14] proposed Tukey's biweight loss function in the regression problems as an alternative to $L2$ loss robust against outliers. We formulate the loss function as $L2$ loss along with constraints applied to hand joints regarding the hand dynamics and appearance, leading to more accurate results and less sensitivity to ground-truth noise.We define the loss function for one frame in the form of:

$$L = \lambda_1 L_{loc} + \lambda_2 L_{glo} + \lambda_3 L_{app} + \lambda_4 L_{dyn}, \tag{3.10}$$

where $\lambda_i \ i \in \{1..4\}$ are factors to balance loss functions. L_{loc}, L_{glo}, L_{app} and L_{dyn} denote the loss for the estimated local and global pose, appearance, and hand dynamics, respectively. Next, each component is explained in detail.

Let $F^l \in \mathbb{R}^{3 \times m}$ be the concatenation of the m estimated joints in each branch of the proposed network and $G^l \in \mathbb{R}^{3 \times m}$ be the ground-truth matrix. Note that m is not necessarily equal to $n = 20$. $F^g \in \mathbb{R}^{3 \times n}$ and $G^g \in \mathbb{R}^{3 \times n}$ are the outputs of the embedded network for estimated joints and ground-truth, respectively. Then, we define local and global losses as:

$$L_{loc} = \sum_{i=1}^{3m} (F_i^l - G_i^l)^2, \tag{3.11}$$

$$L_{glo} = \sum_{i=1}^{3n} (F_i^g - G_i^g)^2. \tag{3.12}$$

A common problem in CNN-based methods for pose estimation is that in some situations estimated pose does not properly fit with appearance. For instance, joints are placed in locations where there is no evidence of presence of hand points, or being physically incorrect [106, 125, 126]. In this paper, during training we penalize those joint estimations that do not fit with the appearance or are physically not possible, and include such penalties in the loss function.

We first assume that, rationally, joints must locate inside the hand area and have a depth value higher than the hand surface, besides, joints must present physically possible angles in the kinematic tree. Therefore, for a given joint j^{xyz} the inequality $\mathscr{I}(j^u, j^v) - j^z < 0$ must hold, where $\mathscr{I}(j^u, j^v)$ is the pixel value at location (j^u, j^v). To avoid violating the first condition (i.e. when a joint is located outside hand area

after projection to the image plane), we set the background with a cone function as:

$$5\sqrt{(u-0.5w)^2 + (v-0.5h)^2} + \phi,$$

where w and h are width and height of the image, and ϕ is a fixed value set to 100. The reason to use a cone function instead of a fixed large value is to avoid zero derivatives on the background. We use hinge formulation to convert inequality to a loss through:

$$L_{app} = \sum_{i=1}^{m} \max(0, \mathscr{I}(j_i^u, j_i^v) - j_i^z). \tag{3.13}$$

We subsequently incorporate hand dynamics by means of the top-down strategy described in Algorithm 1. We assume all joints belonging to each finger (except thumb) should be collinear or coplanar. Thumb has an extra non-coplanar form and we do not consider it in the hand dynamics loss. A groundtruth finger state $s_G \in \{1..4\}$ is assigned to each finger computed by the conditions defined in Algorithm 1. Each finger has a groundtruth normal vector $\mathbf{e}_G$ which is finger direction for the case 1 and finger plane normal vector for the other cases. Therefore, we define four different losses, one of them triggered for each finger (as shown in Algorithm 1). Let A, B, C and D be four joints belonging to a finger starting in A as the root joint and ending in D as fingertip. Then the dynamics loss is defined as:

$$L_{dyn} = \sum_{i=1}^{4} \Delta_i(A, B, C, D, s_G, \mathbf{e}_G), \tag{3.14}$$

where i denotes a finger index. Now we consider each case in Algorithm 1 in the following.

We consider a collinear finger in case 1. A finger is collinear if:

$$\|B - A\| + \|C - B\| + \|D - C\| < \|D - A\| + \kappa,$$

where κ is a threshold defining the amount of collinearity and set to $0.01\|D - A\|$. To compute the loss for a collinear groundtruth finger, the following condition has to be hold: $\rho < \cos(\angle(\overrightarrow{AD}, \mathbf{e}_G)) \leq 1$, where ρ is a threshold. This condition has to be met for $\overrightarrow{AB}$ and $\overrightarrow{AC}$ as well. The cosine function can be extracted through dot

---

Algorithm 1 Top-down strategy for finger dynamics.

input: groundtruth normal vector $\mathbf{e}_G$ defining either finger direction or finger plane normal
input: groundtruth finger state s_G
input: finger joints A,B,C and D (A as finger root)
output: $\Delta(A, B, C, D, s_G, \mathbf{e}_G)$

1: **switch** s_G **do**
2:     **case** 1
3:         $\|\overrightarrow{AB}\| + \|\overrightarrow{BC}\| + \|\overrightarrow{CD}\| < 1.01\|\overrightarrow{AD}\|$
4:         $\overrightarrow{AB} \parallel \overrightarrow{AC} \parallel \overrightarrow{AD} \parallel \mathbf{e}_G$ ▷
5:     **case** 2
6:         $\overrightarrow{AB} \times \overrightarrow{BC} \parallel \overrightarrow{AC} \times \overrightarrow{CD} \parallel \mathbf{e}_G$ ▷
7:     **case** 3
8:         $\overrightarrow{AB} \times \overrightarrow{BC} \parallel \overrightarrow{BC} \times \overrightarrow{CD} \parallel \mathbf{e}_G$ ▷
9:     **case** 4
10:        $\overrightarrow{AB} \times \overrightarrow{BC} \parallel \overrightarrow{AB} \times \overrightarrow{BD} \parallel \mathbf{e}_G$ ▷

product. Therefore, using hinge formulation, the loss is defined as:

$$
\begin{aligned}
\Delta_i(A, B, C, D, 1, \mathbf{e}_G) = \max&\left(0, \rho - \frac{\overrightarrow{AB} \cdot \mathbf{e}_G}{\|\overrightarrow{AB}\|}\right) + \\
\max&\left(0, \rho - \frac{\overrightarrow{AC} \cdot \mathbf{e}_G}{\|\overrightarrow{AC}\|}\right) + \\
\max&\left(0, \rho - \frac{\overrightarrow{AD} \cdot \mathbf{e}_G}{\|\overrightarrow{AD}\|}\right) + \\
\mu\max&(0, \|\overrightarrow{AB}\| + \|\overrightarrow{BC}\| + \|\overrightarrow{CD}\| - 1.01\|\overrightarrow{AD}\|),
\end{aligned}
\tag{3.15}
$$

where μ is a factor to balance different components of the loss function.

We consider a coplanar finger for cases 1, 2 and 3. We define a finger to be coplanar if cross products of all subsets of the finger joints with three members to be parallel. Note that a collinear finger is necessarily coplanar. However, we exclude collinear fingers from this definition due to cross product, as shown in Algorithm 1. For a groundtruth coplanar finger, such cross products must be parallel to the plane normal vector. Therefore, for given joints A, B and C, the following condition must hold:

$$\rho < \cos\left(\angle(\overrightarrow{AB} \times \overrightarrow{BC}, \mathbf{e}_G)\right) \le 1.$$

Given the groundtruth finger is coplanar of case 2, we compute the loss function as:

$$\Delta_i(A, B, C, D, 2, \mathbf{e}_G) = \max\left(0, \rho - \frac{(\overrightarrow{AB} \times \overrightarrow{BC}) \cdot \mathbf{e}_G}{\|\overrightarrow{AB} \times \overrightarrow{BC}\|}\right)$$
$$+ \max\left(0, \rho - \frac{(\overrightarrow{AC} \times \overrightarrow{CD}) \cdot \mathbf{e}_G}{\|\overrightarrow{AC} \times \overrightarrow{CD}\|}\right). \tag{3.16}$$

The loss functions for the other coplanar finger cases are computed in the same way.

3.3.3 Loss function derivatives

All components in Eq. 3.10 are differentiable, thus we are able to use gradient-based optimization methods. In this section we explain derivatives of the constraint loss function in Eq. 3.13. Derivatives of the rest of loss functions are computed through matrix calculations. We first define derivative of L_{app} with respect to $t \in \{j_i^x, j_i^y, j_i^z\}$ through:

$$\frac{\partial L_{app}}{\partial t} = \begin{cases} 0 & \text{if } \mathscr{I}(j_i^u, j_i^v) - j_i^z \le 0 \\ \partial \mathscr{I}/\partial t - \partial j_i^z/\partial t & \text{otherwise.} \end{cases} \tag{3.17}$$

In the following we just consider positive condition of Eq. 3.17. Besides, we omit index i (which denotes i-th joint) from the notations for the easiness of reading. Depth image $\mathscr{I}$ is a discrete multi-variable function of j^u and j^v, where j^u is a multi-variable function of j^x and j^z, and j^v is a multi-variable function of j^y and j^z. Consequently, the total derivative of a depth image can be computed by the chain rule through:

$$\frac{\mathrm{d}\mathscr{I}}{\mathrm{d}t} = \frac{\partial \mathscr{I}}{\partial j^u} \frac{\mathrm{d}j^u}{\mathrm{d}t} + \frac{\partial \mathscr{I}}{\partial j^v} \frac{\mathrm{d}j^v}{\mathrm{d}t} \tag{3.18}$$

$$\frac{\mathrm{d}j^u}{\mathrm{d}t} = \frac{\partial j^u}{\partial j^x} \frac{\mathrm{d}j^x}{\mathrm{d}t} + \frac{\partial j^u}{\partial j^z} \frac{\mathrm{d}j^z}{\mathrm{d}t} \tag{3.19}$$

$$\frac{\mathrm{d}j^v}{\mathrm{d}t} = \frac{\partial j^v}{\partial j^y} \frac{\mathrm{d}j^y}{\mathrm{d}t} + \frac{\partial j^v}{\partial j^z} \frac{\mathrm{d}j^z}{\mathrm{d}t} \tag{3.20}$$

Next, we present components of j^u derivative in detail[1]. Depth image $\mathscr{I}$ is a function of hand surface. However, hand surface given by the depth camera may have noise and not be differentiable at some points. To cope with this problem, we estimate depth image derivatives by applying hand surface normal vectors. Let **s** to be the surface normal vector for a given joint. Then, derivative of $\mathscr{I}$ with respect to u axis is given by the tangent vectors through:

$$\frac{\partial \mathscr{I}}{\partial j^u} = \frac{\mathbf{s}^x}{\mathbf{s}^z}. \tag{3.21}$$

As mentioned, j^{uvz} is the projection of the estimated joint j^{xyz} from world coordinate to the image plane. Note that joints have zero mean and j^{uvz} is extracted after the image has been cropped and resized. Let f_x, p_x, M^{xyz} and M^{uvz} to be the camera focal length and image center for x axis, world coordinate hand point cloud center, and its projection to image plane, respectively. Then j^u is computed as:

$$j^u(j^x, j^z) = \left(\frac{f_x(j^x + M^x)}{j^z + M^z} + p_x - M^u \right) \text{scale}_x + \frac{w}{2},$$
$$\text{scale}_x = \frac{wM^z}{cf_x}, \tag{3.22}$$

where c is the cube size used around hand point cloud to crop the hand image. Using this formulation, derivative of j^u can be easily computed and replaced in Eq. 3.19.

3.3.4   Results

In this section we evaluate our approach on two public real-world datasets: NYU [172] and MSRA [162]. MSRA dataset has less accurate groundtruth comparing to NYU dataset and provides a multi-subject benchmark. We evaluate our approach using two metrics: average distance error in mm and success rate error [169]. Next, we detail the method parameters and evaluate our approach both quantitatively and qualitatively in comparison to state-of-the-art alternatives.

Training

We utilize MatConvNet library [182] on a server with GPU device *GeForce GTX Titan X* with 12 GB memory. We optimize the network using stochastic gradient descent (SGD) algorithm. We set the batch size, learning rate, weight decay and momentum to 50, 0.5e-6, 0.0005 and 0.9, respectively. Although our approach converges in

---

[1]Derivatives belonging to j^v are computed in the same way as j^u
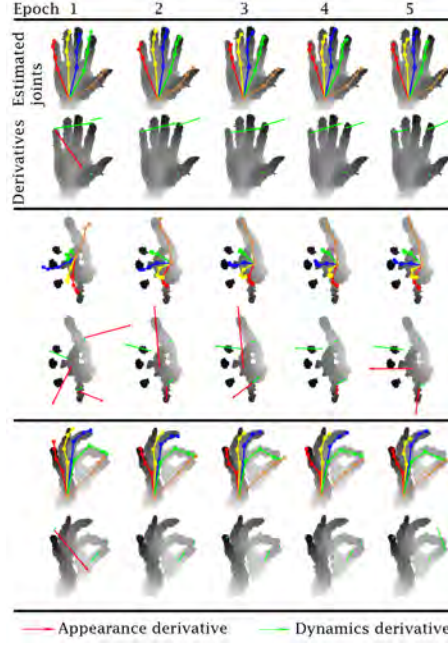
Figure 3.9: Constraints derivatives during training process. Estimated joints along with derivatives of appearance and hand dynamics are illustrated for the first five epochs in the training process. We qualitatively show how proposed network converges very fast in few epochs.

almost 6 epochs, we terminate the process after 20 epochs, while reducing the learning rate by a factor of 10 after epoch 6. Overall, training takes two days while testing takes 50 fps.

Loss function parameters tuning We set a low value for parameter μ in Eq. 3.15 since it behaves like a regularization and it is not connected to ground-truth. L_{dyn} is mainly a summation of cosine functions while L_{app} is in millimeters. Therefore we set λ_4 higher than λ_3 to balance cosine space with millimeter. Finally, we set parameters λ_1, λ_2, λ_3, λ_4 and μ experimentally to 4, 4, 3, 20 and 0.0005, respectively. We show derivatives of appearance and dynamics loss functions for a number of joints in the first five epochs in Fig. 3, as well as qualitative images of estimated joints.
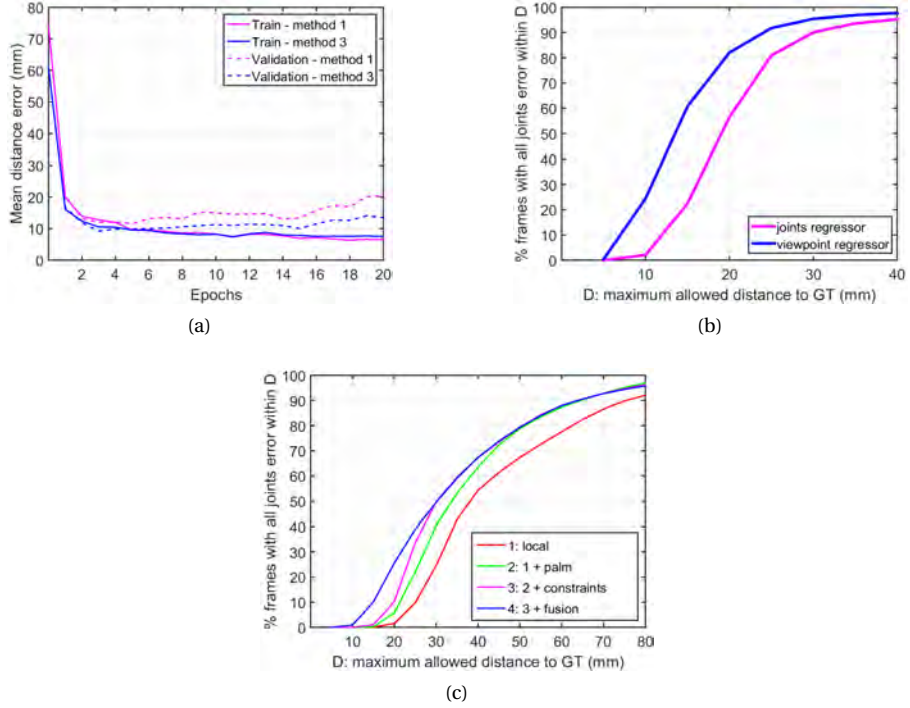
Figure 3.10: Quantitative results comparing baselines on NYU dataset. a) Training process in terms of average error per epoch. b) Comparing palm: joints regression vs. viewpoint regression. c) Maximum success rate, comparing baselines.

Comparison with baselines

We compare with baselines considering four types of our approach on NYU dataset (we denote each by a number): (1) hierarchical network trained just with local poses including one finger in each branch, and without constraints and fusion network (this baseline shows a high locality value), (2) previous baseline along with palm joints included in the local pose in all branches, (3) previous baseline along with constraints, and finally, (4) previous baseline along with fusion network.

First of all, we compare training and validation set trend for baseline methods 1 and 3 in Fig. 3.10a for NYU dataset. It can be seen that, by applying the proposed constraints, method 3 is more robust against overfitting than method 1. Validation error in method 3 does not significantly change from epoch 7 to 15 and starts
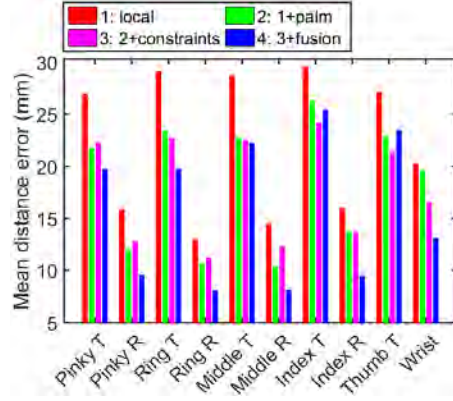
Figure 3.11: Comparison of the proposed network baselines with respect to the mean error per joint on NYU dataset [172].

overfitting slightly after epoch 15. Comparing both methods in epoch 20, method 1 has a lower error in training while its validation error is almost 1.5 times the validation error of method 3.

We evaluate our palm joints vs. palm viewpoint regression in terms of success rate error in Fig. 3.10b. Palm viewpoint regressor gives a rotation matrix in terms of quaternions. We convert quaternions to rotation matrix and use it to transform a predefined reference palm example. As it can be seen in the figure, palm viewpoint regression significantly reduces palm joints error.

We show the success rate with maximum joints error in Fig. 3.10c. As it can be seen, method 1 has the highest error while when palm joints are included in the local pose, method 2 is able to better localize finger joints. By applying constraints in the loss function (method 3), the results are slightly improved for the lower error tolerance while fusion network (method 4) improves method 3. We also illustrate per joint mean error in Fig. 3.11. From the figure, as expected, a very local solution (method 1) performed the worst among the baselines. Comparing method 2 and 3 in average error shows the benefits of applying constraints as loss as well. By including viewpoint features in the fusion network, palm joints mean error was considerably improved by method 4. Although method 4 performed better for the pinky and ring fingertips, it did not achieve the best results for index and thumb fingertips. This opens the future work idea that a selective solution among local and global estimations can be applied as a post-processing solution.

67

(a)

(b)

(c)
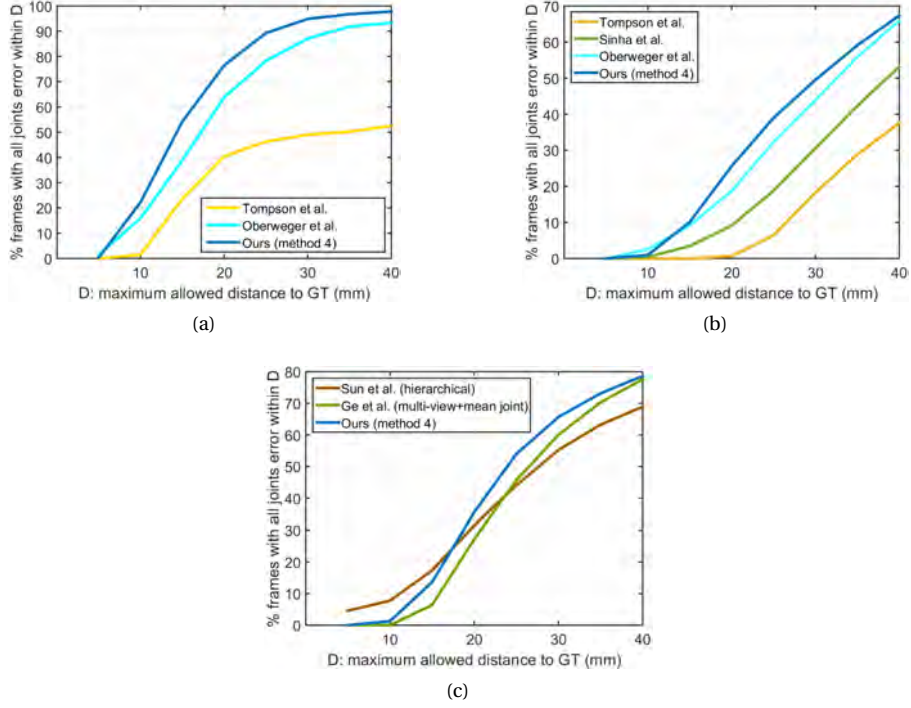
Figure 3.12: State-of-the-art comparison. a) and b) Mean and maximum success rate on NYU dataset. c) Maximum success rate on MSRA dataset.

Comparison with state of the art

We compare our approach to [172], [126], and [157] on NYU dataset and [162] and [106] on MSRA dataset. Note that we compare with state-of-the-art data-driven approaches. Mentioned works use 14 joints (as proposed in [172]) to compare on NYU dataset. For a fair comparison on this dataset we take 11 joints most similar to [172] out of our 20 used joints. We show the results on NYU dataset in Fig. 3.12b based on our method 4. Regarding the maximum success rate we slightly outperform state-of-the-art results. Regarding the average success rate (Fig. 3.12a), we improve state-of-the-art results. We achieve overall mean joint error 15.6 mm vs. 16.5 mm reported in [126]. Some qualitative results on this dataset is shown in Fig. 3.14 comparing to state of the art.

We show the maximum success rate results on MSRA dataset in Fig. 3.12c. For a

(a)

(b)
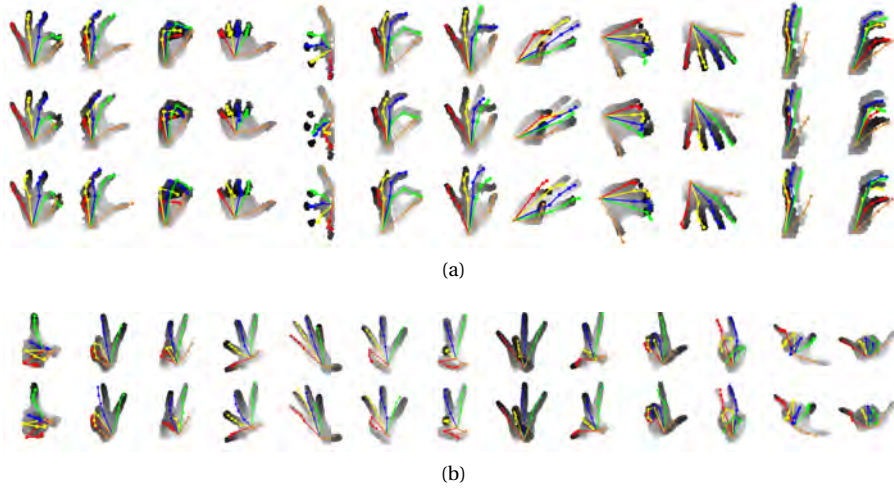
Figure 3.13: Qualitative results. a) NYU dataset. Rows from top to bottom: groundtruth, method (1) and method (4). b) MSRA dataset. Rows from top to bottom: groundtruth and method (4). See text for details of the methods.

fair comparison with CNN-based approaches, we compare to Ge *et al.*[106] without their post-processing results. As it can be seen, we get slightly better results for lower error tolerances comparing to [106]. Although Sun *et al.*[162] has a higher number of good frames for errors lower than 18mm, it performs the worst for higher error rates.
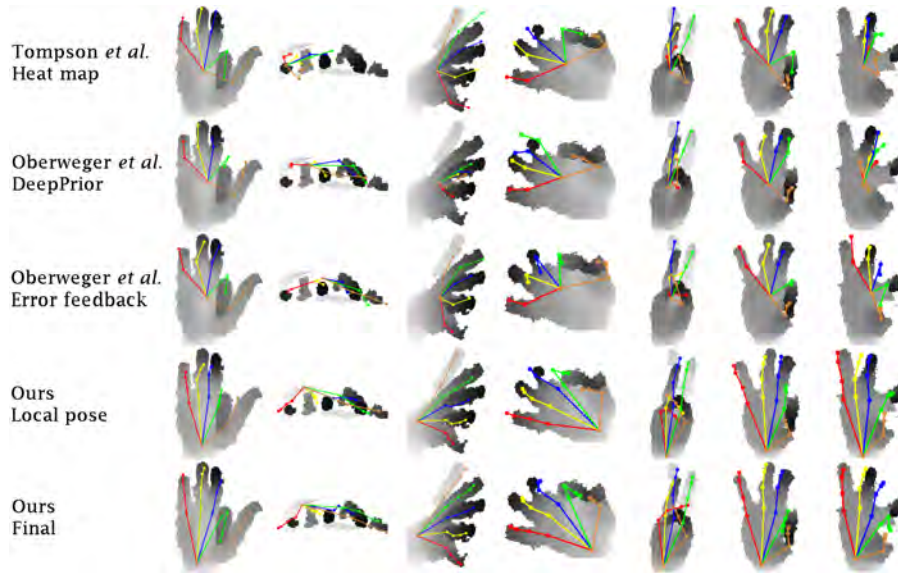
Figure 3.14: Tompson *et al.*[172] just estimates 2D pose using joints heat-map, providing poor pose estimation results in the case of noisy input images (second column). Oberweger *et al.*[125] results (DeepPrior) show that PCA is not able to properly model hand pose configurations. Oberweger *et al.*[126] improved previous results by applying an error feedback loop approach. However, error feedbacks do not provide accurate pose recovery for all the variability of hand poses. In our local pose estimation framework, a separate network is trained for each finger. Then, we fuse learned local features to include a higher order dependency among joints, obtaining better pose estimation results than previous approaches.

4 Applications and datasets

4.1 Soft biometrics measurement

Soft biometrics in contrast to hard biometrics are traits of the human body, like color of the hair, skin, height and weight, that can be used to describe a person. These attributes have a lower power to discriminate and authenticate an individual, but they are easier to compute in comparison to hard biometrics.

Soft biometric traits have been used in video surveillance to track people with single camera systems or even with a discrete joint camera network [33, 34, 140]; as a pre-processing approach to help hard biometric systems to search databases faster or to increase reliability and accuracy [55, 116]; and for other applications like person re-identification [118], supported diagnosis in clinical setups [141], or commercial tools like clothing sizing [27], just to mention a few. Most surveillance systems using soft biometrics have integrated human height as one of their most important cues [34, 71, 140].

Authors in [183] proposed a weight estimation technique that computes weight by summation of coefficients of some soft biometrics like height and calf circumference. Since soft biometrics have semantic correlation in human metrology, these can be computed according to part relations. Recently, [1] studied the problem of predictability and correlation in human metrology applying some statistical measurements between different soft biometrics features in order to make correlation clusters among them to predict unknown body measurements. Authors in [145] used joints estimated by KinectSDK to estimate initial dimensions, afterward multiple Regression of the 2 principal components of estimated body dimensions were applied to estimate other dimensions. Authors in [191] computed body measurements using a regression based approach from body parameters after an accurate scanning of the body.

While most of the biometrics measurements are based on regression on some known body parameters, in this section, first we accurately segment human limbs from a single depth image captured by a Kinect camera (Chap. 2), and as a result we compute traits such as arm and leg lengths, and neck, chest, stomach, waist and

71

Figure 4.1: Red points are non visible points of the model inside occluded mask. Left image is the corresponding depth map.

hip sizes from segmented limbs.

4.1.1 Size measurements

We rely on geometrical body surface measurements and joint locations to compute soft biometrics. Therefore, we need to segment body traits beforehand (as described in Chap. 2). We first align an exemplar model to body point cloud and assume this exemplar model is complete, *i.e.*the coordinates of all body surface is available regardless of self-occlusions. Therefore, after point clouds alignment, one can easily find which areas are self-occluded in the depth image by comparing warped model and depth image using a depth difference threshold. Then, to measure size, we complete occluded body pixels from warped model. Fig. 4.1 shows how occludees are completed in a sample depth image.

After segmenting body, the lengths of arms and legs are easily obtained by computing the joint locations like shoulder, elbow and wrist for arms; or hip, knee and ankle for legs. But the most challenging part in size measurements lies in the estimation of the circumference of body traits like neck, chest or waist. Depending on the training data, one could add more body parameters like principle components as degree of obesity and using such a more robust model, full size measurements are possible from body completion. This can obtain even more accurate results in multi-view systems which require a point cloud nearest neighbor based approach.

Next we describe a geometrical approach to compute camera view circumference of such traits: we estimate the orthogonal plane to the body principal axis so that the intersection of this plane and the body hull surface is used for estimating those measurements. Since the principal axis of the body is the symmetry axis of it, we assume that this axis starts at the mean point of the hip segment boundary and ends at the mean point of the neck segment boundary. These boundaries lie on the segments after estimating labels. The accuracy of principal axis strongly depends on the segmentation accuracy. Similarly, the principal axis of the neck starts at the
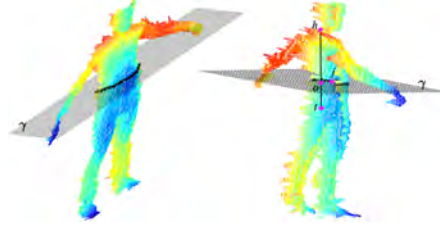
Figure 4.2: Constructing trait curve for size measurement using orthogonal plane γ to the body principal axis.

mean point of the neck segment boundary and ends at the head joint.

As shown in Fig. 4.2, let h be the head point and t be the tail point of the principal axis, j the joint point of a segment, γ be the orthogonal plane to the principal axis crossing at j, and o be the intersection of γ and the principal axis. In this assumption, o is unknown and we compute it using other known points as:

$$o = \alpha(h - t) + t, \tag{4.1}$$

where α is the plane γ factor computed as:

$$\alpha = \frac{\sum (h - t) \circ (m - t)}{\sum (h - t)^2}. \tag{4.2}$$

Let p be a point on body hull that belongs to the selected segment. Point p lies on γ plane if and only if $\overrightarrow{op} \cdot \overrightarrow{oh} = 0$ is satisfied. Since the body hull point cloud is a discrete surface, we threshold the dot product for all points in the segment to estimate the intersection curve. However, the resulting narrow strip of points is still not appropriate for measurements. We divide the resulting strip into non-overlapping segments to extract the mean point of each segment, and then consider the Euclidean distance between neighbor segments. Applying such an interpolation reduces affects of boundary points noises. These measurements can be used in regression based approaches as initial parameters.

For small segments like neck, using completed point cloud and tuning the weights of each segment in SVM segmentation improves segment line analysis. An important parameter is the threshold of dot product which can be tuned for different point cloud densities and segments.
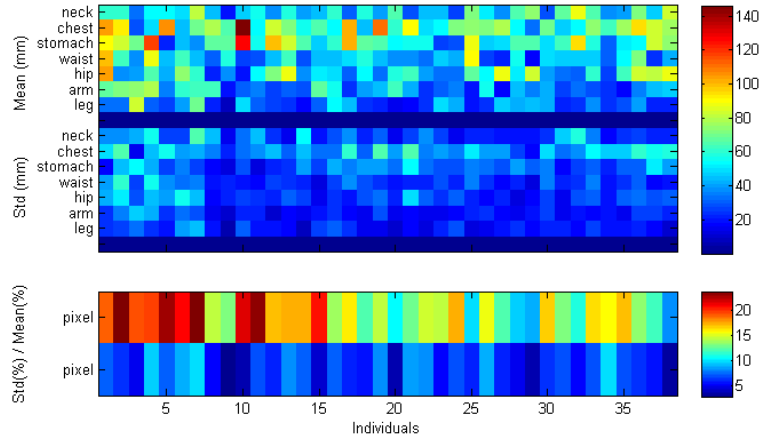
Figure 4.3: Overall size error per person in mm.

Table 4.1: The average mean and standard deviation error in mm for all the data.

	Neck	Chest	Stomach	Waist	Hip	Arm	Leg
	55.76	**69.47**	**64.63**	**46.60**	**55.61**	**41.44**	**30.65**
Our method	±	±	±	±	±	±	±
	33.57	**49.58**	**39.84**	**31.45**	**34.35**	**25.65**	**24.07**

4.1.2   Results

We show in Fig. 4.3 the average limbs size errors among all subjects. This shows that the data distribution among all individuals is not normal and some data is more challenging for measurements. Notice that segment lines in different parts lie into the segments according to the Fig. 2.6 even for small segments like neck.

The accuracy of measurements is directly related to the accuracy of the segmentation and database labels: chest has the highest size errors (because clothes affect mostly on this part) whereas arm and leg have the lowest error values. Other source of errors are the affect of clothes in some poses as well as human faults in taking groundtruth. Besides self occlusions problem for example in the chest part has been solved by completing the point cloud. Table 4.1 summarizes the average mean and standard deviation errors in mm per limb.

4.2 Garment retexturing

As shopping for garments is increasingly moving to a digital domain, the next step after just seeing the desired clothes is to virtually try them on. Due to the fact that an actual try-on of clothes is time-consuming, a virtual alternative has always been desired, and many researchers have been engaged in developing novel strategies and systems to perform such a task [31, 60, 149, 174, 215]. It requires scanning, classification of the body based on gender and size, 3D modeling [48, 59, 206] and visualization. Constrained texture mapping and parametrization of triangular mesh are some popular examples, although they suffer from some deficiencies such as finding the parameter values and manual adjustments [103, 113]. Many researchers have also suggested methodologies for visually fitting garments onto the human body based on dense point clouds [12, 63]. However, garment retexturing in a virtual fitting room is still an open problem [30, 143, 177].

The focus of this section is an application for a specialized fitting room where the images of the person are captured with a Kinect-2 RGB-D camera. There are several steps between taking an RGB-D picture and displaying the final result with a retextured garment. These steps involve segmentation of the garment, garment matching and surface retexturing. Retexturing part involves several challenges. First, a coordinate map must be created between the image of the new texture and the image that is being retextured. This problem is especially difficult in the case of non-rigid and easily transformable surfaces like clothes. One of the biggest problems of easily deformable surfaces is self-occlusion. In many cases, one part of a surface blocks the visibility of another part of the surface. To achieve a realistic result, it is necessary to consider this occlusion aspect in the computation of texture coordinates. Another challenge is to shade the new texture correctly. It is possible to use the color information of the original image, but the lighting, intensity and the original color of the surface are usually not previously known and must be estimated.

The proposed automatic retexturing method, after the segmentation stage uses the point set registration method [72] to find correspondence between the outer 2D contours of the person and the target garment. After the contour matching, the surface topology of the flat 2D garment is approximated using geodesic distance in a global closed form solution using thin plate spline (TPS) [19] and the final result is superimposed onto the segmented area.

4.2.1 Literature review

The matching problem stage can be defined as a correspondence problem, which incorporates pair-wise constraints. Hence, it is often solved with a graph matching

approach [44, 62, 212], which is especially suitable for deformable object matching. Furthermore, additional constraints can be added to the framework in order to reduce the computation time (e.g. clearly, each cloth type is constrained to the body part where it is dressed), or in order to take problem-specific aspects into account.

There exist various techniques for conducting a mapping from 2D image texture space to a 3D surface. Some examples are intermediate 3D shape [17], direct drawing onto the object [57], or using an exponential fast marching method by applying geodesic distance [176, 196]. Many researchers have devoted special attention trying to attempts to enhance the realism of virtual garment representation during the last decade [22]. One of the most frequently used texture fitting methods was proposed by Turquin *et al.* [179], which allows the users to sketch garment contours directly onto a 2D view of a mannequin. The initial algorithm has been further enhanced by many other researchers [200, 205].

Another popular way of mapping a 2D texture onto a 3D surface is by using a single image [210]. As proposed by [212], an estimation of a 3D pose and shape of the mannequin is followed by constructing an oriented facet for each bone of a mannequin according to angles of the pose, and projecting the 2D garment outlines into corresponding facets. Eckstein *et al.* [40] proposed a constrained texture mapping algorithm, which can be used for 2D and 3D modeling, and multi-resolution texture mapping and texture deformation, but it may produce a Steiner vertex effect when a simple solution does not exist. Kraevoy *et al.* [88] introduced a method based on iterative optimization of a constrained texture mapping method. In their method, it is a requirement to specify the corresponding constraint points on the grid model and texture image, the parametrized mesh. Later, Yanwen *et al.* [199] reported a constrained texture mapping method based on harmonic mapping, with interactive constraint selection by the user; the method produces high efficiency, real-time optimization, and adjustment of mapping results. The block based constrained texture mapping methods are also used in order to bring higher speed and lower computational costs [110].

There exist several standard methods for projecting textured surfaces on screen. The simplest shading methods work only by using surface normals independently without considering the overall surface, attempting to estimate the brightness of the surface given some known viewer and light source direction. Examples of this kind of method are the Gouraud shading, Phong shading, and Blinn-Phong shading [151]. However, these methods do not support shadows.

4.2.2 Retexturing approach

We propose an automatic retexturing method covering the stages of garment segmentation, 2D to 3D garment matching and rendering. As pre-processing, we use
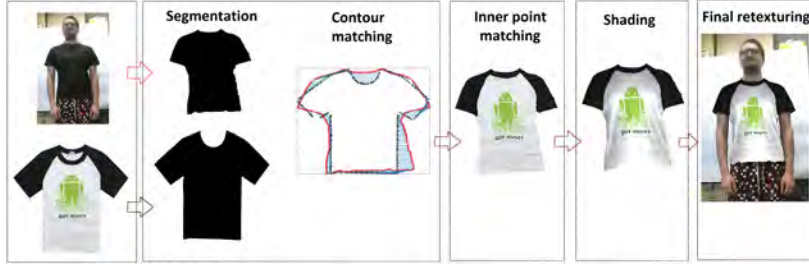
Figure 4.4: Overview of the proposed retexturing method.

RGB, depth and infrared images of the Kinect and segment out the garment from the background. The segmented depth image is used to compute retexturing from a source 2D flat garment image. We reduce the problem of surface point matching to an interpolating problem by using garment contour matching. The interpolation process takes surface topology into account using geodesic distance in a global closed form solution using thin plate spline (TPS) [19]. Thus, 2D garment contours are matched beforehand applying point registration based on Gaussian mixture models [72]. Finally the resulting mapped source image is sampled, and the segmented area can be superimposed using these colors. As a result, realistic rendering is provided showing both qualitative and quantitative advantages in relation to state-of-the-art method alternatives based on thin-plate splines with geodesic interpolation. The proposed retexturing method is visualized in Fig. 4.4.

Segmentation

In order to make accurate measurements in real world units, we standardize the coordinate system of body and garment models according to real world coordinates. Moreover, rich visualization includes aligned image data (RGB and depth images), so as to provide animations as close as possible to the real scenario [62].

The first step of the proposed retexturing method is segmentation of garments from the background. It is necessary to extract a set of points from the image corresponding to the area being retextured. The proposed method works under the following assumptions: the area to be retextured is a shirt (or some other initially known garment) worn by a person, the person is assumed to be standing in front of the camera and is assumed not to occlude the area of interest with his/her hands. The segmentation is done by first extracting pixels and the skeleton of the body using Kinect SDK. Skeleton joint locations along with some artificial joints are used to train the GrabCut algorithm [142] and select areas with desired joints. In the

case of the reference 2D image, the GrabCut algorithm is also applied initializing the background color with the pixels on the borders of the image. This simple automatic segmentation approach worked accurately in our dataset. In case of other non-controlled scenarios, any other automatic or semi-automatic segmentation approach could be considered.

Outer contour matching

Contour matching can be viewed as a point set registration problem, where a correspondence must be found between a scene and a model. A few of the most well known methods for point set registration are iterative closest point [16], robust point matching [56, 161], and Coherent point drift [121] algorithms. For our purposes, a correspondence must be found between highly deformed shapes. Out of available algorithms, we have chosen to use non-rigid point set registration using Gaussian mixture models [72] because of its accurate fitting under different conditions and fast execution time. Additionally, Gaussian mixtures provide robust results even if the shapes have different features, such as different neck lines, hand positions and folds.

Let's define the contour of a garment on a real person as C_R and the contour of the flat garment as C_F. The aim is to create a correspondence between contour models C_R and C_F. In the point matching algorithm, the point sets are represented by Gaussian mixture models. The interpretation is such that a statistical sample is drawn from a continuous probability distribution of random point locations. Afterward the point set registration problem is viewed as an optimization problem, meaning that a certain dissimilarity measure between the Gaussian mixtures constructed from the transformed model set and the fixed scene set is minimized based on L2 distance between the mixtures.

Before finding the corresponding points between the shapes, the contours are down-sampled and normalized. Essentially the used method provides information about how C_R has to be transformed to match C_F. After the transformation is found, nearest neighbor search is used to find the corresponding points between the two contours. Outer contour matching examples are shown in Fig. 4.5.

Inner contour matching

Inner contour matching refers to the process of finding correspondence points between the body surface and the 2D flat garment in order to assign to each body point a color from the garment. This process is mainly a difficult task due to, first, the lack of depth information for the 2D flat garment and the lack of texture for the depth image, and second, dissimilar textures for the source 2D flat garment and target put-on garment. Therefore feature based matching is not applicable.
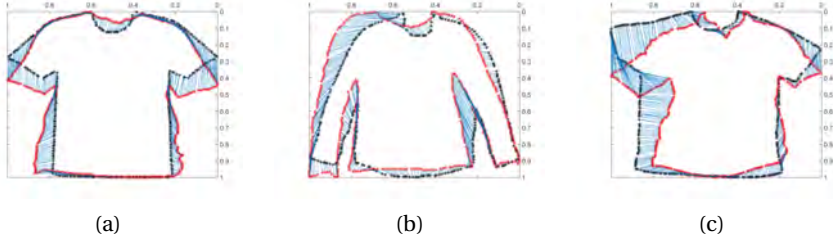
Figure 4.5: Short and long sleeve examples for contour correspondences obtained using point set registration. Red contour corresponds to C_R and blue contour corresponds to C_F.

Conformal based approaches like [193, 204] fail due to the different topologies of the surfaces.

In order to solve this problem efficiently, we first generate a triangulated 3D mesh based on the depth image of the segmented area. To have a smooth shape at boundaries, we apply some morphological filtering at segmented body shape borders. A solution can be obtained by finding an affine deformation matrix for each face triangle to bring both source and target surfaces into alignment according to the matched points of the outer contours. However, we cannot guarantee a perfect matching for near contour points in such a solution due to different surface topologies and depth camera noise in the contours. Instead, we propose to use thin plate splines (TPS) [19] as a solution in closed-form based on a radial basis kernel. Let $X = \{x_1, ..., x_N\} \in \mathbb{R}^3$ be the set of all points belonging to the segmented and discretized body surface Ω. Then, a mapping from x_i to the source image is computed through

$$W(x_i) = \sum_{j=1}^{n} \omega_j \kappa(\|x_i - C_{R_j}\|), \qquad (4.3)$$

where ω is a set of trained coefficients based on C_R and C_F, $\kappa(d) = d^2 \log d$ is a radial basis kernel and n is the number of contour points. This basic formulation is based on Euclidean distance among the points which is not applicable for our problem since contour points do not cover all the surface; besides that, Euclidean distance does not describe the surface topology. Instead we propose a geodesic-based distance to include surface topology. We show this idea in Fig. 4.6.

Since we apply discretized body surface Ω, the Dijkstra algorithm can be used to compute the shortest distance from x_i to each C_{R_j}, $j \in \{1..n\}$. However, we get
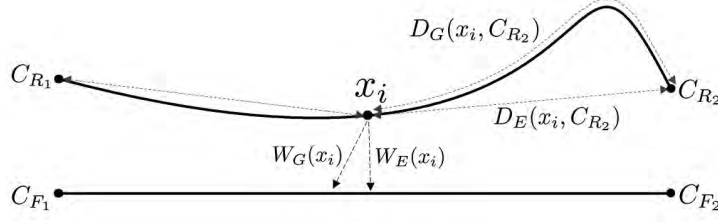
Figure 4.6: Comparing Euclidean to geodesic distance in TPS. TPS finds a mapping between two point sets based on known correspondences. In this image we consider such a mapping between two 2D example lines where end points C_R are matched with C_F. As can be seen, point x_i has equal Euclidean distance D_E to points C_{R_1} and C_{R_2}. In this case, mapping W_E does not take line topology into account, causing a wrong interpolation where points on the right hand side of $W_E(x_i)$ get much denser than the points on its left hand side. This problem can be solved by using geodesic distance D_G in the mapping W_G.

a stairstep-like shortest path which causes an amount of error in the distance, no matter how much we refine mesh. Instead, we follow the fast marching algorithm of [36] to compute a fast and accurate approximation of geodesic distance. The fast marching algorithm is closely related to the Dijkstra algorithm with the difference that it satisfies the Eikonal equation $\|\nabla U(x)\| = 1/s(x), x \in \Omega$ to update the graph where $\nabla U(x)$ is the gradient of the action map U and $s(x)$ is a positive outwards speed function at point x. $U(x)$ is a function of time at point x that describes the evolution of the surface with respect to $s(x)$ and surface gradient. We assume the surface is differentiable at all points. Starting from x_i, at each iteration, the algorithm sweeps outwards one grid point with respect to $s(x)$ to locate the proper grid point to update. Then geodesic distance can be computed for two vertices v_i and v_j from the shortest path $L = \{L_1, ..., L_m\}$ by

$$\Gamma(v_i, v_j) = \sum_{l=1}^{m-1} \|L_l - L_{l+1}\| \tag{4.4}$$

To compute geodesic distance efficiently, we set a flag for cell d_{ij} of the distance table as 1 if vertices v_i and v_j already exist on a larger optimum path, avoiding recomputing the optimum path for them.

Then we rewrite the TPS formulation to compute the coefficient matrix ω as

$$\omega = \begin{bmatrix} \dot{K}_{n\times n} + \lambda I & [1|C_{R_{n\times3}}] \\ [1|C_{R_{n\times3}}]^\top & 0 \end{bmatrix}_{(n+4)^2}^{-1} \begin{bmatrix} C_{F_{n\times3}} \\ 0 \end{bmatrix}_{(n+4)\times3}, \tag{4.5}$$

where $\dot{K}_{ij} = \Gamma(C_{R_i}, C_{R_j})^2 \log \Gamma(C_{R_i}, C_{R_j}) \forall i, j \in \{1, ..., n\}, i \neq j$. λI is a regularization term and is added to the kernel $\dot{K}$ where I is the identity matrix and $\lambda \in \mathbb{R}$. λ values close to zero make the kernel sensitive to wrong correspondences, and values far from zero tend to an affine transformation. We set λ to -1000, and by doing so, the visualization becomes more realistic and less noisy.

Afterward, a solution can be achieved by applying trained coefficients as

$$W = [\ddot{K}_{N\times n}|1|X_{N\times3}]\omega \tag{4.6}$$

where $\ddot{K}_{ij} = \Gamma(X_i, C_{R_j})^2 \log \Gamma(X_i, C_{R_j})$. Matrix W includes warped points to the 2D shirt image. We assign each point the color of its corresponding pixel from the shirt image.

Shading

The shading effect of the garment is achieved using an adaptation of method [8] which is an automatic technique for garment retexturing and shading, where the shading information is acquired from Kinect 2 infrared information and is superimposed on the inner shape results. It is worth noticing that shadow mapping on the garment is not the main contribution of this section, and thus its usage and coverage are limited to the extent demanded for visualizing the results illustrating the effectiveness of the proposed mapping method.

The general procedure for obtaining the final visualization is as follows. The point cloud corresponding to the area of interest provided by the Microsoft Kinect 2 camera is triangulated and rendered as described in the previous section. The image created as a result of mapping in the previous steps is used as a texture image, such that each vertex corresponds to a point on the image. Afterward, the rendered image is modified by the corresponding infrared values for each pixel. Finally, the segmented area in the Kinect frame is replaced by the color information from the previous step.

For the sake of enhancing the representation quality, the point cloud is preprocessed before rendering, since it usually is noisy. More clearly, smoothing the depth image with a Gaussian filter is considered, which, according to our experiments, significantly improves the results.
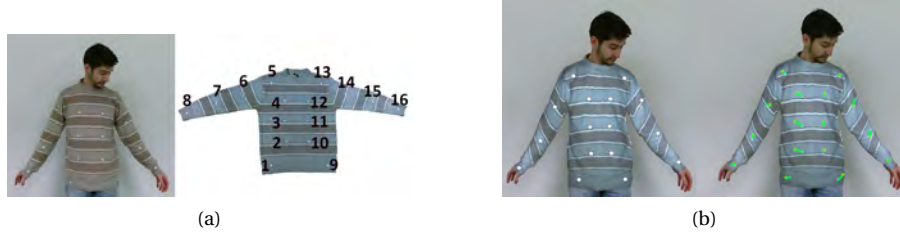
Figure 4.7: a) Landmarks used in the second dataset on the flat garment (right) and landmark locations after putting the garment on as ground truth. Landmarks are shown by indices for comparison purposes. b) Retextured garment and estimated landmarks (left) and displacement arrows to ground truth landmarks (right) for computing error.

4.2.3   Results

In order to evaluate our method, we created two datasets: first dataset for qualitative evaluation and second dataset with attached landmarks for quantitative evaluation (see details in Chap. 4). We used two metrics for evaluation of our method: qualitative comparison using the mean opinion score (MOS), and quantitative comparison using the mean square error (MSE). The MOS score was measured by showing 91 sets of images from the first dataset to 41 people. Each person was asked for an opinion about which one of the images in each set looks visually more realistic. The MSE was measured on the second dataset by retexturing the flat version of the shirt and computing the average distance from retextured landmarks to ground truth landmarks. Fig. 4.7b shows the process of computing MSE. We compare our method with two state-of-the-art methods, nonrigid iterative closest point (NRICP) [5] and coherent point drift (CPD) [121], using introduced evaluation metrics.

All compared results were produced with the same set of parameters that were determined empirically. The setup parameters for matching the contours needed for the point registration algorithm [72] are set as follows (see original paper for the definition of parameters): sigma, which is the scale parameter of Gaussian mixtures, is set to 0.2 and 0.1, and the maximum number of function evaluations at each level is set to 50, 500, 100, 100 and 100. The point registration algorithm uses contours with 400 points. After the transformation and point correspondence are found, the contour is further down-sampled to 120 points and used for the inner point matching. A larger number would have resulted in a long computation time, whereas a smaller number of points resulted in some undersampled parts and

Figure 4.8: Retexturing effects for different necklines

produced inferior mappings. 120 points were chosen as a compromise between the execution time and the resulting mapping quality.

Evaluation

We separated long and short sleeve images in the results to analyze them separately. We show the MOS percentage in Table 4.2. The results illustrate that our method outperforms state-of-the-art methods by a large margin regarding realistic view. This can also be seen qualitatively in Fig. 4.11. We added correspondences between flat garment and body contours in the third column of Fig. 4.11 to see the effect of outer contour matching on the retexturing results. It can be seen that the final retextured image still has a realistic appearance even with small misalignment in outer correspondences. However, a small misalignment can have a local impact. This can mainly be seen in the long sleeves. If the source and target garments have different features, for example if a collar is present in the put-on image and not present in the flat image, some unnatural effects may be seen; the same goes for different neck lines as shown in Fig 4.8. The NRICP algorithm has the worst visual results due to the different topologies of the surfaces between flat garment and body, and the CPD algorithm has difficulties with aligning surfaces in the boundary regions.

MSE values are shown in Table 4.3. As seen from the visual results, in most cases our method is more accurate than state-of-the-art methods regarding marker distances to ground truth. Our method generates a lower error for short sleeves than long sleeves. However, this is not a significant change according to the MSE results. Often our method performs better than other methods for almost each marker in Fig. 4.9 and 4.10 where samples represent different garments, and landmarks are the white circles that are placed on the garment, as is shown in Fig. 4.7. Our method is more stable among different persons and different markers in comparison to the state-of-the-art methods. However, the long sleeves error as seen in Fig. 4.10(a) fluctuates among different persons due to higher variation in hand position. Marker numbers 8 and 16, which were placed at the end of the sleeves, have the highest error in both long and short sleeve garments. This happens due to slight point misalignment in outer contour matching.

Table 4.2: Mean Opinion Score (MOS) comparison

Method	T-shirt Votes	T-shirt Percentage	Long sleeve Votes	Long sleeve Percentage
NRICP [5]	77	2.68%	32	3.69%
CPD [121]	485	16.88%	245	28.23%
Ours	2311	80.44%	591	68.09%

Table 4.3: Marker mapping error

Method	MSE for T-shirts	MSE for Long sleeves
NRICP [5]	115.400 px	215.349 px
CPD [121]	83.850 px	190.618 px
Ours	75.005 px	105.884 px

(a)   (b)
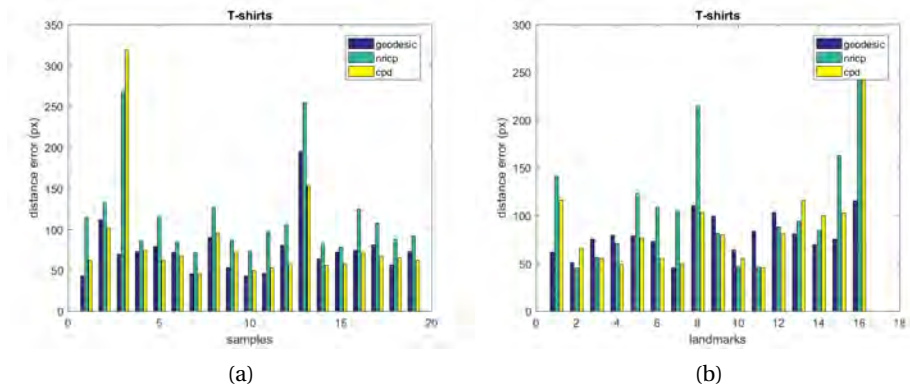
Figure 4.9: Method comparison for t-shirts: graph (a) shows the sample pixel distance error for each sample and graph (b) shows the pixel distance error for each of the landmarks.

At current stage, the method is implemented in Matlab, and the processing time for one Kinect frame on an Intel Core i7-2670QM 2.2GHz CPU takes from 5 to 10 minutes. The calculation time depends on the pixel count in the retextured surface. We believe that execution time can be greatly reduced by converting the method to a low level programming language and using cloud computing solutions.
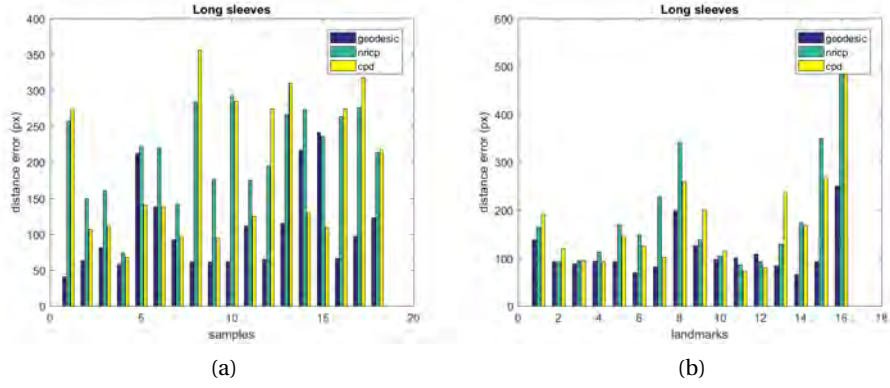
Figure 4.10: Method comparison for garments with long sleeves: graph (a) shows the sample pixel distance error for each sample and graph (b) shows the pixel distance error for each of the landmarks.

4.3 Datasets

During the time of this dissertation, we have created a number of datasets: human body segments and soft biometrics, synthetic hand segments and pose, garment and body, multi-modal Italian gestures. In the following we explain each dataset in details.

4.3.1 Human body

This dataset contains 1155 frames of 38 individuals, 7 females and 31 males, with a resolution of 640 × 480 pixels captured by a Kinect using the OpenNI library [129]. Each frame consists of RGB and depth image, label of each body pixel, a complete model in the self-occlusion cases, as well as, ground truth values of the front views of limbs sizes with a ±20 mm human error in measurements. We manually labeled pixels which caused non-homogenous segments among all subjects (a sample segmentation is available in Fig. 2.1b). Subjects rotate facing the camera in a range of ±60° such that the whole body was observable. Fig. 4.12 shows a number of samples in the dataset. We used this dataset for body segmentation and soft biometrics measurement.

85

Figure 4.11: Images created by the proposed retexturing method, (a) is the original image, (b) is the image of a shirt, (c) shows the shape correspondence, (d) is the retextured image based on the geodesic mapping, (e) is mapping using the Coherent Point Drift (CPD) algorithm and (f) is mapping using the non-rigid Iterative Closest Point (ICP) algorithm.

4.3.2 Synthetic hand

We created this dataset for hand pose recovery and segmentation to evaluate our methods in any view point under strong self-occlusions. This dataset contains

Figure 4.12: Sample images used in the body segmentation and soft biometrics measurement dataset.

single frame hand images, hand mocap data and sequences of hand deformations in test set.

Data generation Datasets were generated with Blender 2.74 using a detailed, realistic 3D model of a human adult male hand (Fig. 2.1c). The model was rigged using a hand skeleton (Fig. 3.1a) with four bones per finger, reproducing the distal, intermediate, and proximal phalanges, as well as the metacarpals. The thumb finger had no intermediate phalanx and was controlled with three bones. Additional bones were used to control palm and wrist rotation. Unfeasible hand poses were avoided by defining per-bone rotation constraints. All finger phalanges had only 1-DoF rotation (for finger flexion/extension) but metacarpals had 2-DoF rotation to allow for finger adduction/abduction. This resulted in 4-DoF per finger (except for the thumb), which proved to be enough to reproduce all reasonable poses in the context of gesture-based interaction (see some sample poses in Fig. 4.13).

The original male model was deformed manually to fit a female hand by displacing the original vertices through Blender's *shape keys*. This allowed us to generate an arbitrary number of intermediate models through the blend shapes defined by the linear interpolation of the vertices from their original (male hand) to the modified (female hand) positions.

Points on the hand's surface were assigned a unique color label identifying the underlying skeleton joint, as shown in Fig. 2.1d. The palm center was assumed to be roughly at the metacarpals' centroid.

The animated hand model was rendered using a virtual camera reproducing the image resolution and the intrinsic parameters of the target depth sensor (Kinect-2).

Figure 4.13: Upper two rows are some sample poses, lower two rows are a small sample set of the depth images generated for the test set. The image shows ten interpolation frames between four predefined hands poses.

The virtual camera was always aiming at the hand, from a view direction which was chosen randomly from a uniform discretization of the Gauss sphere (we used 320 directions associated with the normal vectors of a subdivided icosahedron).

Training datasets We generated two different training sets. For the first dataset, we generated three pieces of data: a color image (pixel labels), a depth image, and a text file containing the location of the skeleton joints. Each training example was generated by randomly choosing a view-direction and a hand pose (Fig. 4.13). We generated over 600K samples for this dataset and used it for RF training and nearest neighbor extraction.

For the second dataset, we just produced the text files containing the joints locations. Camera viewpoint was fixed in this dataset in order to benefit from a reference viewpoint and palm joints were aligned. We provided temporal data in this dataset including a smooth interpolation between pairs of key poses. Key poses were chosen either randomly or from a small set of predefined poses. We included different deformation speeds in this dataset. The unique motion range of the thumb (which includes opposition-reposition, besides flexion-extension and adduction-abduction) forced us to prevent finger self-intersections by inserting additional frames. This guaranteed feasible and natural hand movements. We generated over 1200K frames for this dataset and used it to extract clips and train bilinear model.

Test dataset For generating this dataset we followed the same rule as our second dataset except we produced the color labels, depth images, and text descriptions, and camera rotations were smooth along pose interpolation frames (see Fig. 4.13). We generated over 8K frames for this dataset.

Figure 4.14: Sample images used in our dataset. a) T-shirts used in first data set, b) Long sleeve shirts used in first data set and c) People who participated in creating the first data set.

4.3.3 Garment/body

This dataset was taken using the Kinect 2 RGB-D camera to test our proposed retexturing method. According to [198], Kinect 2 can capture frames starting from 0.5 meters and has depth accuracy error smaller than 2 mm in the cater part of the frame. The error increases towards edges of the frame, and it also increases with greater measurement distances. The best distance for scanning objects is the 0.5 to 2m range. To achieve the best depth resolution, the people were scanned at a distance of 1.5 to 2 meters where the error in the horizontal and vertical plane is

the smallest. Each image contains a person facing the camera in a pose that does not significantly occlude the worn garment. The garments segmented from the original database were retextured using another database consisting of images of flat shirts. The flat shirt database was captured with various cameras providing decent quality images, as depth was not required. The first data set contained 91 retextured images with 14 people (11 males and 3 females). This data set used 13 flat garments (4 long sleeve garments and 9 t-shirts). The second data set contained 39 retextured images with 5 people (4 males and 1 female). This data set used 8 flat garments (4 long sleeve garments and 4 t-shirts). We physically attached 16 landmarks to garments in the second dataset. This was done in order to determine the retexturing precision by retexturing the same garment onto itself. Fig. 4.7a shows a sample of a real put-on image and the landmarked garment itself. Some samples of both datasets are shown in Fig. 4.14 and 4.15.

(a)

(b)

(c)
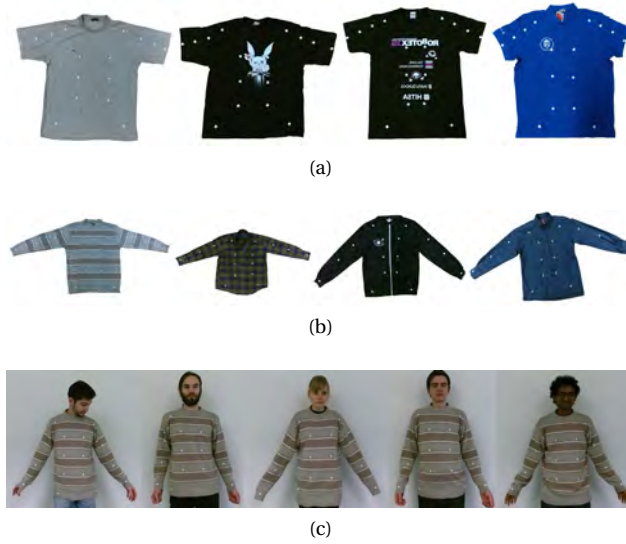
Figure 4.15: Sample images used in our second dataset. a) T-shirts used in first data set, b) Long sleeve shirts used in first data set and c) People who participated in creating the first data set.

Training seq.	Validation seq.	Test seq.	Sequence duration	FPS
393 (7,754 gestures)	287 (3,362 gestures)	276 (2,742 gestures)	1-2 min	20

Modalities	Num. of users	Gesture categories	Labeled sequences	Labeled frames
RGB, Depth, User mask, Skeleton	27	20	13,858	1,720,800

Table 4.4: Main characteristics of the *Montalbano* gesture dataset.

4.3.4 Montalbano: Italian gestures

This multi-modal gesture recognition dataset was created as the third track of Chalearn LAP 2014. The RGBD data contains nearly 14K manually labeled (beginning and ending frame) gesture performances in continuous video sequences, with a vocabulary of 20 Italian gesture categories. This third track focused on multi-modal automatic learning of a set of gestures with the aim of performing user independent continuous gesture recognition.

RGB   Depth   User mask   Skeletal model
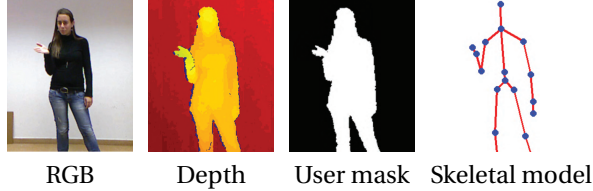
Figure 4.16: Different modalities of the Montalbano dataset.

In all the sequences, a single user is recorded in front of a Kinect, performing natural communicative gestures and speaking in fluent Italian. Examples of the different visual modalities are shown in Fig. 4.16. In ChaLearn LAP 2014 we have focused on the user-independent automatic recognition of a vocabulary of 20 Italian cultural/anthropological signs in image sequences.

A list of data attributes for this dataset is described in Table 4.4. The main characteristics of the database are:

• Largest dataset in the literature, with a large duration of each individual performance showing no resting poses and self-occlusions.

• There is no information about the number of gestures to spot within each sequence, and several distractor gestures (out of the vocabulary) are present.

• High intra-class variability of gesture samples and low inter-class variability for some gesture categories.

5 Conclusions

In this thesis we addressed human segmentation and hand pose recovery in various modalities. We showed a close relationship between segmentation and pose recovery in face and hand where one can help recovering the other. The aim is to provide accurate low and mid-level computer vision solutions applicable in high-level applications. Feature extraction, as a low-level technique, plays an important role in computer vision. We developed feature extraction using both hand crafted and CNN-based techniques. The usage of feature extraction can be application dependent. However, regardless of the application, a difference between hand crafted descriptors and CNN-based features is that the latter is data dependent and the features can be learned. This causes CNN-based features to be more powerful for specific data and application, but may fail to describe images outside the scope of the training data. In this thesis we developed hand crafted features based on both stand-alone shape structure and learned training data. More specifically, we developed a 2.5D shape descriptor which computes spatial dependencies of shape regions with respect to the shape itself and also other shapes in the dataset. For the latter task, we trained RF classifier on training data to discriminate shape regions and we showed that even with low accuracy in the RF output, we still have reasonable discriminative power by computing shape structure into bins.

We applied human body and hand segmentation in depth images using example-based approaches. We showed that segmentation by exploiting discriminant analysis among nearest neighbors is efficient using a rigid alignment of shapes. However, it may not be accurate for smaller segments specially in lack of enough data samples. Therefore, a non-rigid alignment of shapes can generalize better in smaller datasets. We applied iterative correspondence matching using shape context point descriptor. We showed accurate human segmentation even for small segments like wrist in a gathered dataset. We also worked on RGB modality for face segmentation. Other than depth modality where objects can segmented from the background by background subtraction and depth thresholding, in RGB modality background is mainly segmented as a separate label along with other object labels. We proposed a CNN-based approach to segment face along with hair and background. While CRF based

approaches use a fixed pair-wise kernels, we let the network learn pair-wise kernels without explicitly training unary potentials. We showed training this network using generative adversarial network significantly improve hair segmentation which is the most challenging part of face. In this technique we fed an initial segmentation into the network based on face landmarks which helped the network better localize segments.

We also worked on 3D hand pose recovery in depth images and proposed two solutions based on generative models and CNNs. In generative model, we fit a simple finger model separately for each finger based on hand segmentation. While we used landmarks to help face segmentation, here we applied vice-versa and showed hand segmentation could help to reduce search space of model parameters and thus a faster and more reliable pose recovery even with a greedy solution and a set of predefined finger candidates. We also used temporal joint trajectories to recover occluded joints based on appearance and trajectory smoothness function. We used linear models to learn trajectories. To better exploit linear models in highly nonlinear space of hand pose trajectories, we clusterized data. Minimizing the model parameters over the whole clusters is not efficient for real-time applications, so we approximated nearest cluster by efficiently searching for best cluster. Gradient based minimization approaches suffer from initial parameter guess. So we used particle based minimization with a normal distribution of particles. As a result, we slightly improved occluded joints in a highly variable viewpoint dataset that we gathered.

In our hand pose recovery solutions, we heavily rely on breaking a complex problem into simpler problems and solve them individually. For instance segmenting hand using global shape structure and solving palm and fingers separately. We applied CNNs similarly as pose regressors to learn specialized features for finger and palm poses as simpler pose problems than the whole set of hand joint locations. For this task we developed a tree-structure network which solves each finger and palm in each branch. This network is followed by a fusion network to learn global consistency of the joints. We also applied physical and appearance constraints to the network as loss functions to correct infeasible hand configurations. It was also useful to avoid groundtruth noise in the training dataset. We introduced palm viewpoint regression as a rotation matrix in terms of quaternions and showed much more stable results than palm joints regression.

Humans are the subjects in many applications in computer vision and human analysis has a spread domain from medical diagnosis, sports analysis and security to entertainment, retailing and fashion design. Finally, the aim is to develop techniques to solve real life application dependent problems. During this thesis we developed some applications consisting of human soft biometrics measurements and human garment retexturing. For both applications, we relied on segmen-

tation and surface geometry. In the latter application, garment was segmented and retexturing formulated as garment points matching to a source garment. We matched inner contour points efficiently by interpolating target points based on target surface topology and outer contour points matching.

5.1 Future work

In this thesis we used single modality as the input to the problem, either RGB or depth. However, fusing multi-modal data can help reducing ambiguities in one modality by another. Even including motion patterns like optical flow as another modality may produce more stable pose recovery in CNNs for instance [119].

Although our example-based **human body segmentation** worked well in our dataset, it has not been tested on more complex datasets. In our proposed nonrigid warping, we do not apply constraints on connectivity of adjacent pixels in the model and therefore incorrect correspondences may cause an unrealistic model deformation. Applying spring-like constraints in the deformation model avoids this phenomena. Even statistical linear shape models like SMPL [108] can be applied for more stable model deformation.

Our proposed shape descriptor worked well in practice for **hand pose recovery**. We will apply our generative hand pose recovery in real datasets. Also, our temporal occluded joints refinement just relies on initial single frame pose estimation and does not recover pose from failures. As future work we will develop a spatio-temporal model based on appearance and trajectory motion patterns. Our synthetic hand model enables us to provide a huge amount of data for temporal motion analysis not only for generative models but also for discriminative CNN models. The availability of such data is quite important for the success of temporal CNNs, recurrent neural networks and memory cells. CNN-based structured learning in hand pose recovery has not been vastly studied and following the works in human pose recovery, joints can be extracted by modeling hand structure into CRF inference or by using sample candidates and structural SVM. Efficient feature extraction is a key reason in the success of a CNN model and more intelligent hand crafted connection of layers are proposed instead of letting standard networks learn feature activations in specific regions by their own.

In the proposed **applications**, we will apply mentioned generative statistical models like SMPL. These models are symmetric, able to give pose and used for segmentation. Therefore, **soft biometrics** can be extracted by regressing fitted shape PCA components or even by our proposed geometric-based approach (*e.g.* geodesic distance). SMPL can be applied in **garment retexturing** as well. So after fitting the model into body, given any pose or viewpoint, source garment contours can

be matched to predefined labeled points on the model. Then this model can be retextured and projected to body surface.

References

Bibliography

[1] Donald Adjeroh, Deng Cao, Marco Piccirilli, and Arun Ross. Predictability and correlation in human metrology. *IEEE International Workshop on Information Forensics and security*, 2010.

[2] A. Agarwal and B. Triggs. A local basis representation for estimating human pose from cluttered images. *ACCV*, 1:50–59, 2006.

[3] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, dec 2006.

[4] Ijaz Akhter, Tomas Simon, Sohaib Khan, Iain Matthews, and Yaser Sheikh. Bilinear spatiotemporal basis models. *TOG*, 31(17), 2012.

[5] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[6] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 623–630, 2010.

[7] Vassilis Athitsos, Carol Neidle, Stan Sclaroff, Joan Nash, Alexandra Stefan, Quan Yuan, and Ashwin Thangali. The american sign language lexicon video dataset. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[8] Egils Avots, Morteza Daneshmand, Andres Traumann, Sergio Escalera, and Gholamreza Anbarjafari. Automatic garment retexturing based on infrared information. *Computers & Graphics*, 59:28–38, 2016.

[9] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[10] Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[11] ALC Barczak, NH Reyes, M Abastillas, A Piccio, and T Susnjak. A new 2d static hand gesture colour image dataset for asl gestures. 2011.

[12] Sandro Barone, Alessandro Paoli, and Armando Viviano Razionale. Three-dimensional point cloud alignment detecting fiducial markers by structured light stereo imaging. *Machine Vision and Applications*, 23(2):217–229, 2012.

[13] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures for multiple human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1669–1676, 2014.

[14] Vasileios Belagiannis, Christian Rupprecht, Gustavo Carneiro, and Nassir Navab. Robust optimization for deep regression. In *ICCV*, pages 2830–2838, 2015.

[15] Serge Belongie, Jitendra Malik, and Jon Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.

[16] N.D. McKay Besl, Paul J. A method for registration of 3-d shapes. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 239—-256. IEEE, 1992.

[17] E.A. Bier and K.R. Sloan. Two-part texture mappings. *IEEE Computer Graphics and Applications*, 6(9):40–53, 1986.

[18] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016.

[19] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[20] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *European conference on computer vision*, pages 44–57. Springer, 2008.

[21] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4733–4742, 2016.

[22] W.C. Chang and W.C. Chang. Real-time 3d rendering based on multiple cameras and point cloud. In *7th International Conference on Ubi-Media Computing and Workshops*, pages 121–126.

[23] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[24] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

[25] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[26] Xianjie Chen and Alan L Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems*, pages 1736–1744, 2014.

[27] Yu Chen, Duncan P. Robertson, and Roberto Cipolla. A practical system for modelling body shapes from single view measurements. In *BMVC*, pages 1–11, 2011.

[28] Chiho Choi, Ayan Sinha, Joon Hee Choi, Sujin Jang, and Karthik Ramani. A collaborative filtering approach to real-time hand pose estimation. *ICCV*, 2015.

[29] Ira Cohen, Ashutosh Garg, and Thomas S Huang. Emotion recognition from facial expressions using multilevel hmm. In *in In Neural Information Processing Systems*, 2000.

[30] Frédéric Cordier, W Lee, H Seo, and Nadia Magnenat-Thalmann. Virtual-try-on on the web. *Laval Virtual*, 2001.

[31] Morteza Daneshmand, Alvo Aabloo, Cagri Ozcinar, and Gholamreza Anbar-jafari. Real-time, automatic shape-changing robot adjustment and gender classification. *Signal, Image and Video Processing*, pages 1–8, 2015.

[32] Martin de La Gorce, David J. Fleet, and Nikos Paragios. Model-based 3d hand pose estimation from monocular video. *PAMI*, 33(9), 2011.

[33] Meltem Demirkus, Kshitiz Garg, and Sadiye Guler. Automated person catego-rization for video surveillance using soft biometrics. volume 7667, 2010.

[34] Simon Denman, Alina Bialkowski, Clinton Fookes, and Sridha Sridha-ran. Identifying customer behaviour and dwell time using soft biometrics. *Springer*, 409:199–238, 2012.

[35] Emily L. Denton, Soumith Chintala, Arthur Szlam, and Robert Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *CoRR*, abs/1506.05751, 2015.

[36] Thomas Deschamps and Laurent D. Cohen. Fast extraction of minimal paths in 3d images and applications to virtual endoscopy, 2001.

[37] Jian Dong, Qiang Chen, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. Towards unified human parsing and pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 843–850, 2014.

[38] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *CoRR*, abs/1602.02644, 2016.

[39] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. pages 1538–1546, 2015.

[40] I. Eckstein, V. Surazhsky, and C. Gotsman. Texture mapping with hard con-strains. *Computer Graphics Forum*, 20(3):95–104, 2001.

[41] Michael Elad and Peyman Milanfar. Style-transfer via texture-synthesis. *CoRR*, abs/1609.03057, 2016.

[42] Ali Erol, George Bebis, Mircea Nicolescu, Richard D. Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1–2):52 – 73, 2007.

[43] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, jun 2014.

[44] Huijie Fan, Yang Cong, and Yandong Tang. Object detection based on scale-invariant partial shape matching. *Machine Vision and Applications*, 26(6):711–721, 2015.

[45] Xiaochuan Fan, Kang Zheng, Yuewei Lin, and Song Wang. Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. pages 1347–1355, 2015.

[46] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.

[47] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International journal of computer vision*, 61(1):55–79, 2005.

[48] Sid Ahmed Fezza and Mohamed-Chaker Larabi. Color Calibration of Multi-View Video Plus Depth for Advanced 3D Video. *Signal, Image and Video Processing*, pages 1–15, 2015.

[49] Golnaz Ghiasi and Charless C Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *European Conference on Computer Vision*, pages 519–534. Springer, 2016.

[50] Ke Gong, Xiaodan Liang, Xiaohui Shen, and Liang Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. *arXiv preprint arXiv:1703.05446*, 2017.

[51] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[52] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *The IEEE International Conference on Computer Vision (ICCV)*, 2009.

[53] Yagmur Güçlütürk, Umut Güçlü, Marcel A. J. van Gerven, and Rob van Lier. Deep impression: Audiovisual deep residual networks for multimodal apparent personality trait recognition. *CoRR*, abs/1609.05119, 2016.

[54] Yagmur Güçlütürk, Umut Güçlü, Rob van Lier, and Marcel A. J. van Gerven. Convolutional sketch inversion. *CoRR*, abs/1606.03073, 2016.

[55] Guodong Guo, Guowang Mu, and K Ricanek. Cross-age face recognition on a very large database: The performance versus age intervals and improvement using soft biometric traits. *ICPR*, pages 3392–3395, 2010.

[56] Chui Haili and Rangarajan Anand. A new point matching algorithm for non-rigid registration. In *Computer Vision and Image Understanding - Special issue on nonrigid image registration (Volume:89 , Issue: 2-3)*, pages 114—141. Elsevier Science Inc., 2003.

[57] Pat Hanrahan and Paul Haeberli. Direct wysiwyg painting and texturing on 3d shapes. *ACM SIGGRAPH computer graphics*, 24(4):215–223, 1990.

[58] S. L. Happy and Aurobinda Routray. Automatic facial expression recognition using features of salient facial patches. *IEEE Transactions on Affective Computing*, 6(1):1–12, jan 2015.

[59] Jacques Harvent, Benjamin Coudrin, Ludovic Brèthes, Jean-José Orteu, and Michel Devy. Multi-view dense 3d modelling of untextured objects from a moving projector-cameras system. *Machine vision and applications*, 24(8):1645–1659, 2013.

[60] Stefan Hauswiesner, Matthias Straka, and Gerhard Reitmayr. Free Viewpoint Virtual Try-on with Commodity Depth Cameras. In *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, pages 23–30. ACM, 2011.

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[62] P. Henry, M. Krainin, E. Herbst, X. Ren, and D Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *International Journal of Robotics Research*, 31(5):647–663, 2012.

[63] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Rnvironments. In *12th International Symposium on Experimental Robotics (ISER)*. Citeseer, 2010.

[64] Antonio Hernandez-Vela, Nadezhda Zlateva, Alexander Marinov, Miguel Reyes, Petia Radeva, Dimo Dimov, and Sergio Escalera. Graph cuts optimization for multi-limb human segmentation in depth maps. *CVPR*, pages 726–732, 2012.

[65] Antonio Hernandez-Vela, Nadezhda Zlateva, Alexander Marinov, Miguel Reyes, Petia Radeva, Dimo Dimov, and Sergio Escalera. Human limb segmentation in depth maps based on spatio-temporal graph cuts optimization. *JAISE*, 4:535–546, 2012.

[66] Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *Advances in Neural Information Processing Systems*, pages 1495–1503, 2015.

[67] Gary B Huang, Manjunath Narayana, and Erik Learned-Miller. Towards unconstrained face recognition. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[68] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[69] James S. Supancic III, Gregory Rogez, Yi Yang, Jamie Shotton, and Deva Ramanan. Depth-based hand pose estimation: methods, data, and challenges. *arXiv:1504.06378v1*, 2015.

[70] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[71] E Jeges, I Kispal, and Z Hornak. Measuring human height using calibrated cameras. *Proceedings of HSI*, pages 755–760, 2008.

[72] Bing Jian and Baba C. Vemuri. Robust point set registration using gaussian mixture models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1633–1645. IEEE, 2010.

[73] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. doi:10.5244/C.24.12.

[74] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

[75] David Joseph Tan, Thomas Cashman, Jonathan Taylor, Andrew Fitzgibbon, Daniel Tarlow, Sameh Khamis, Shahram Izadi, and Jamie Shotton. Fits like a glove: Rapid and reliable hand shape personalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5610–5619, 2016.

[76] Andrew Kae, Kihyuk Sohn, Honglak Lee, and Erik Learned-Miller. Augmenting crfs with boltzmann machine shape priors for image labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2019–2026, 2013.

[77] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aäron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *CoRR*, abs/1610.10099, 2016.

[78] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. Institute of Electrical and Electronics Engineers (IEEE), jun 2014.

[79] Roland Kehl, Matthieu Bray, and Luc Van Gool. Full body tracking from multiple views using stochastic sampling. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 129–136. IEEE, 2005.

[80] James Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. Springer, 2010.

[81] C. Keskin, F. Kirac, Y.E. Kara, and L. Akarun. Real time hand pose estimation using depth sensors. *ICCV Workshops*, pages 1228–1234, 2011.

[82] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. *ECCV*, 7577:852–863, 2012.

[83] Khalil Khan, Massimo Mauro, and Riccardo Leonardi. Multi-class semantic segmentation of faces. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 827–831. IEEE, 2015.

[84] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, jul 2009.

[85] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[86] Furkan Kirac, Yunus Emre Kara, and Lale Akarun. Hierarchically constrained 3d hand pose estimation using regression forests from single frame depth data. *Pattern Recognition Letters*, 50(0):91 – 100, 2014.

[87] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. *CoRR*, abs/1611.09577, 2016.

[88] V. Kraevoy, A. Sheffer, and C. Gotsman. Matchmaker: Constructing constrained texture maps. 22(3), 2003.

[89] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*, 2012.

[90] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[91] Marcel Körtgen, Marcin Novotni, and Reinhard Klein. 3d shape matching with 3d shape contexts. In *In The 7th Central European Seminar on Computer Graphics*, 2003.

[92] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V. Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[93] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S. Huang. Interactive facial feature localization. In *Computer Vision – ECCV 2012*, pages 679–692. Springer Nature, 2012.

[94] Erik Learned-Miller, Gary B. Huang, Aruni RoyChowdhury, Haoxiang Li, and Gang Hua. Labeled faces in the wild: A survey. In *Advances in Face Detection and Facial Image Analysis*, pages 189–248. Springer Nature, 2016.

[95] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016.

[96] Kuang-chih Lee, Dragomir Anguelov, Baris Sumengen, and Salih Burak Gokturk. Markov random field models for hair and face segmentation. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pages 1–6. IEEE, 2008.

[97] Younjeong Lee, Ki Yong Lee, and Joohun Lee. The estimating optimal number of gaussian mixtures based on incremental k-means for speaker identification, 2006.

[98] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. *CoRR*, abs/1604.04382, 2016.

[99] Sijin Li, Weichen Zhang, and Antoni B Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. *ICCV*, 2015.

[100] Xiaodan Liang, Si Liu, Xiaohui Shen, Jianchao Yang, Luoqi Liu, Jian Dong, Liang Lin, and Shuicheng Yan. Deep human parsing with active template regression. *IEEE transactions on pattern analysis and machine intelligence*, 37(12):2402–2414, 2015.

[101] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[102] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[103] Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J Gortler. A Local/global Approach to Mesh Parameterization. In *Computer Graphics Forum*, volume 27, pages 1495–1504. Wiley Online Library, 2008.

[104] Si Liu, Xinyu Ou, Ruihe Qian, Wei Wang, and Xiaochun Cao. Makeup like a superstar: Deep localized makeup transfer network. *CoRR*, abs/1604.07102, 2016.

[105] Sifei Liu, Jimei Yang, Chang Huang, and Ming-Hsuan Yang. Multi-objective convolutional learning for face labeling. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical and Electronics Engineers (IEEE), jun 2015.

[106] Junsong Yuan Liuhao Ge, Hui Liang and Daniel Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. *CVPR*, 2016.

[107] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[108] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248, 2015.

[109] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *CoRR*, abs/1611.08408, 2016.

[110] Lok Ming Lui, Ka Chun Lam, Tsz Wai Wong, and Xianfeng Gu. Texture map and video compression using beltrami representation. *SIAM Journal on Imaging Sciences*, 6(4):1880–1902, 2013.

[111] Ping Luo, Xiaogang Wang, and Xiaoou Tang. Hierarchical face parsing via deep learning. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2480–2487. IEEE, 2012.

[112] Khoa Luu, Chenchen Zhu, Chandrasekhar Bhagavatula, T. Hoang Ngan Le, and Marios Savvides. A deep learning approach to joint face detection and segmentation. In *Advances in Face Detection and Facial Image Analysis*, pages 1–12. Springer Nature, 2016.

[113] Yuewen Ma, Jianmin Zheng, and Jian Xie. Foldover-Free Mesh Warping for Constrained Texture Mapping. *IEEE Transactions on Visualization and Computer Graphics*, 21(3):375–388, 2015.

[114] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, 2013.

[115] Alexandros Makris, Nikolaos Kyriazis, and Antonis Argyros. Hierarchical particle filtering for 3d hand tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 8–17, 2015.

[116] Amit Mhatre, Srinivas Palla, Sharat Chikkerur, and Venu Govindaraju. Efficient search and retrieval in biometric databases, spie defense and security. In *Symposium, March-2005*, 2001.

[117] Ajmal Mian, Mohammed Bennamoun, and Robyn Owens. An efficient multimodal 2d-3d hybrid approach to automatic face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1927–1943, nov 2007.

[118] Andreas Møgelmose, Albert Clapés, Chris Bahnsen, Thomas B. Moeslund, and Sergio Escalera. *Tri-modal Person Re-identification with RGB, Depth and Thermal Features*. IEEE, 2013.

[119] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4207–4215, 2016.

[120] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[121] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:2262 – 2275, 2010.

[122] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[123] Natalia Neverova, Christian Wolf, Graham W Taylor, and Florian Nebout. Hand segmentation with structured convolutional learning. In *Asian Conference on Computer Vision*, pages 687–702. Springer, 2014.

[124] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[125] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. *Computer Vision Winter Workshop*, 2015.

[126] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. *ICCV*, 2015.

[127] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis Argyros. Efficient model-based 3d tracking of hand articulations using kinect. *BMVC*, pages 101.1–101.11, 2011.

[128] Gabriel L Oliveira, Abhinav Valada, Claas Bollen, Wolfram Burgard, and Thomas Brox. Deep learning for human part discovery in images. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1634–1641. IEEE, 2016.

[129] OpenNI. Available at `http://openni.org/`, 6/2012.

[130] Marco Paladini, Adrien Bartoli, and Lourdes Agapito. Sequential non-rigid structure-from-motion with the 3d-implicit low-rank shape model. *ECCV*, pages 15–28, 2010.

[131] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[132] Pedro HO Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, pages 82–90, 2014.

[133] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2013.

[134] Gerard Pons-Moll, Andreas Baak, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bodo Rosenhahn. Multisensor-fusion for 3d full-body human motion capture. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 663–670. IEEE, 2010.

[135] Francisco Pujol, Mar Pujol, Antonio Jimeno-Morenilla, and María Pujol. Face detection based on skin color segmentation using fuzzy entropy. *Entropy*, 19(1):26, jan 2017.

[136] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. *CVPR*, 2014.

[137] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 413–420. IEEE, 2009.

[138] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[139] Deva Ramanan. Learning to parse images of articulated bodies. *NIPS*, pages 1129–1136, 2006.

[140] Yang Ran, Gavin Rosenbush, and Qinfen Zheng. Computational approaches for real-time extraction of soft biometrics. *ICPR*, pages 1–4, 2008.

[141] Miguel Reyes, Albert Clapés, José Ramírez, Juan R. Revilla, and Sergio Escalera. Automatic digital biometry analysis based on depth maps. *Computers in Industry*, (0):–, 2013.

[142] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004.

[143] Hideo Saito, Shigeyuki Baba, and Takeo Kanade. Appearance-based virtual view generation from multicamera videos captured in the 3-d room. *Multimedia, IEEE Transactions on*, 5(3):303–316, 2003.

[144] Shunsuke Saito, Tianye Li, and Hao Li. Real-time facial segmentation and performance capture from RGB input. *CoRR*, abs/1604.02647, 2016.

[145] I. Samejima, K. Maki, S. Kagami, M. Kouchi, and H. Mizoguchi. A body dimensions estimation method of subject from a few measurement items using kinect. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 3384–3389, Oct 2012.

[146] Shreyas Saxena and Jakob Verbeek. Convolutional neural fabrics. *CoRR*, abs/1606.02492, 2016.

[147] Timo Scharwächter, Markus Enzweiler, Uwe Franke, and Stefan Roth. Stixmantics: A medium-level model for real-time semantic scene understanding. In *European Conference on Computer Vision*, pages 533–548. Springer, 2014.

[148] Carl Scheffler and Jean-Marc Odobez. Joint adaptive colour modelling and skin, hair and clothing segmentation using coherent probabilistic index maps. In *British Machine Vision Association-British Machine Vision Conference*, number EPFL-CONF-192633, 2011.

[149] Sarbartha Sengupta and Parag Chaudhuri. Virtual Garment Simulation. In *Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pages 1–4. IEEE, 2013.

[150] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Leichter, Kim Christoph, Rhemann Ido, Alon Vinnikov Yichen Wei, Daniel Freedman Pushmeet Kohli Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3633–3642. ACM, 2015.

[151] Peter Shirley, Michael Ashikhmin, and Steve Marschner. *Fundamentals of computer graphics*. CRC Press, 2009.

[152] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. 2011.

[153] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[154] Xianbiao Shu, F. Porikli, and N. Ahuja. Robust orthonormal subspace learning: Efficient recovery of corrupted low-rank matrices. *CVPR*, pages 23–28, 2014.

[155] Edgar Simo-Serra, Arnau Ramisa, Guillem Alenyà, Carme Torras, and Francesc Moreno-Noguer. Single image 3d human pose estimation from noisy observations. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2673–2680. IEEE, 2012.

[156] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[157] Ayan Sinha, Chiho Choi, and Karthik Ramani. Deephand: robust hand pose estimation by completing a matrix imputed with deep features. pages 4150–4158, 2016.

[158] Brandon M. Smith, Li Zhang, Jonathan Brandt, Zhe Lin, and Jianchao Yang. Exemplar-based face parsing. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*. Institute of Electrical and Electronics Engineers (IEEE), jun 2013.

[159] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[160] I.-O. Stathopoulou, E. Alepis, G.A. Tsihrintzis, and M. Virvou. On assisting a visual-facial affect recognition system with keyboard-stroke pattern information. *Knowledge-Based Systems*, 23(4):350–356, may 2010.

[161] Gold Steven, Rangarajan Anand, Lu Chien-Ping, Suguna Pappu, and Mjolsness Eric. New algorithms for 2d and 3d point matching:: pose estimation

and correspondence. In *Pattern Recognition (Volume:31)*, pages 1019—-1031, 1998.

[162] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *CVPR*, 2015.

[163] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[164] David Joseph Tan, Tom Cashman, Jonathan Taylor, Andrew Fitzgibbon, Daniel Tarlow, Sameh Khamis, Shahram Izadi, and Jamie Shotton. Fits like a glove: Rapid and reliable hand shape personalization. *CVPR*, 2016.

[165] D. Tang, H.J. Chang, A. Tejani, and T-K. Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. *CVPR*, 2014.

[166] Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. *ICCV*, pages 3224–3231, 2013.

[167] Lili Tao and Bogdan J. Matuszewski. 3d deformable shape reconstruction with diffusion maps. *BMVC*, 2013.

[168] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, et al. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)*, 35(4):143, 2016.

[169] Jonathan Taylor, Jamie Shotton, Toby Sharp, and Andrew Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. pages 103–110, 2012.

[170] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[171] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. pages 648–656, 2015.

[172] Jonathan Tompson, Murphy Stein, Yann LeCun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *TOG*, 33(5), 2014.

[173] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *NIPS*, 2014.

[174] Jing Tong, Jin Zhou, Ligang Liu, Zhigeng Pan, and Hao Yan. Scanning 3D Full Human Bodies Using Kinects. *Transactions on Visualization and Computer Graphics*, 18(4):643–650, 2012.

[175] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[176] A Traumann, M Daneshmand, S Escalera, and G Anbarjafari. Accurate 3d measurement using optical depth information. *Electronics Letters*, 51(18):1420–1422, 2015.

[177] Andres Traumann, Gholamreza Anbarjafari, and Sergio Escalera. A new retexturing method for virtual fitting room using kinect 2 camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 75–79, 2015.

[178] Stavros Tsogkas, Iasonas Kokkinos, George Papandreou, and Andrea Vedaldi. Semantic part segmentation with deep learning. *CoRR*, abs/1505.02438, 2015.

[179] E. Turquin, M. P. Cani, and J. F. Hughes. Sketching garments for virtual characters. In *ACM SIGGRAPH*, 2007.

[180] Jorge Usabiaga, Ali Erol, George Bebis, Richard Boyle, and Xander Twombly. Global hand pose estimation by multiple camera ellipse tracking. *Machine Vision and Applications*, 2009.

[181] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016.

[182] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

[183] Carmelo Velardo, Antitza Dantcheva, Angela D'Angelo, and Jean Luc Dugelay. Bag of soft biometrics for person identification. *Multimedia Tools Appl.*, 51:739–777, 2011.

[184] Nan Wang, Haizhou Ai, and Shihong Lao. A compositional exemplar-based model for hair segmentation. In *Computer Vision – ACCV 2010*, pages 171–184. Springer Nature, 2011.

[185] Nan Wang, Haizhou Ai, and Feng Tang. What are good parts for hair shape modeling? In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 662–669. IEEE, 2012.

[186] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Joint object and part segmentation using deep learned potentials. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1573–1581, 2015.

[187] Yang Wang, Duan Tran, and Zicheng Liao. Learning hierarchical poselets for human parsing. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1705–1712. IEEE, 2011.

[188] Jonathan Warrell and Simon JD Prince. Labelfaces: Parsing facial features by multiclass labeling with an epitome prior. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 2481–2484. IEEE, 2009.

[189] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[190] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. *CVPR*, 2016.

[191] Alexander Weiss, David Hirshberg, and Michael J. Black. Home 3d body scans from noisy image and range data. *ICCV*, pages 1951–1958, 2011.

[192] Aaron Wetzler, Ron Slossberg, and Ron Kimmel. Rule of thumb: Deep derotation for improved fingertip detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 33.1–33.12. BMVA Press, September 2015.

[193] T. Windheuser, U. Schlickewei, F.R. Schmidt, and D. Cremers. Geometrically consistent elastic matching of 3d shapes: A linear programming solution. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2134–2141, 2011.

[194] Lu Xia, Chia-Chih Chen, and J.K. Aggarwal. View invariant human action recognition using histograms of 3d joints. *HAU3D*, 2012.

[195] Chi Xu and Li Cheng. Efficient hand pose estimation from a single depth image. *ICCV*, pages 3456–3462, 2013.

[196] S. Xu and J. Keyser. Texture mapping for 3d painting using geodesic distance. In *18th meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2014.

[197] Yaser Yacoob and Larry S Davis. Detection and analysis of hair. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1164–1169, 2006.

[198] L. Yang, L. Zhang, H. Dong, A. Alelaiwi, and A. El Saddik. Evaluating and improving the depth accuracy of kinect for windows v2. In *IEEE Sensors Journal (Volume: 15, Issue: 8)*, pages 4275–4285. IEEE, 2015.

[199] G. Yanwen, Y. Pan, X. Cui, and Q. Peng. Harmonic maps based constrained texture mapping method. *Journal of Computer Aided Design and Computer Graphics*, 7:1457–1462, 2005.

[200] Z. Yasseen, A. Nasri, W. Boukaram, P. Volino, and Magnenat-Thalmann N. Sketch-based garment design with quad meshes. *Computer-Aided Design*, 45(2):562–567, 2013.

[201] Mao Ye, Xianwang Wang, Ruigang Yang, Liu Ren, and Marc Pollefeys. Accurate 3d pose estimation from a single depth image. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 731–738. IEEE Computer Society, 2011.

[202] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *CoRR*, abs/1511.07122, 2015.

[203] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhand Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. *arXiv preprint arXiv:1704.02612*, 2017.

[204] Wei Zeng, Yun Zeng, Yang Wang, Xiaotian Yin, Xianfeng Gu, and Dimitris Samaras. 3d non-rigid surface matching and registration based on holomorphic differentials. *ECCV*, pages 1–14, 2008.

[205] M. Zhang, L. Lin, Z. Pan, and N. Xiang. Topology-independent 3d garment fitting for virtual clothing. *Multimedia Tools and Applications*, 2013.

[206] Yaoye Zhang, Zhengxing Sun, Kai Liu, and Yan Zhang. A Method of 3D Garment Model Generation Using Sketchy Contours. In *Sixth International Conference on Computer Graphics, Imaging and Visualization*, pages 205–210. IEEE, 2009.

[207] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13(2):119–152, 1994.

[208] Haitian Zheng, Yebin Liu, Mengqi Ji, Feng Wu, and Lu Fang. Learning high-level prior with convolutional neural networks for semantic segmentation. *CoRR*, abs/1511.06988, 2015.

[209] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*. Institute of Electrical and Electronics Engineers (IEEE), dec 2015.

[210] B. Zhou, X. Chen, Q. Fu, K. Guo, and P. Tan. Garment modeling from a single image. *Computer Graphics Forum*, 32(7):85–91, 2013.

[211] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[212] Feng Zhou and Fernando De la Torre. Factorized graph matching. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 127–134. IEEE, 2012.

[213] Feng Zhou and Fernando De la Torre. Spatio-temporal matching for human detection in video. *ECCV*, 8694:62–77, 2014.

[214] Yisu Zhou, Xiaolin Hu, and Bo Zhang. Interlinked convolutional neural networks for face parsing. In *International Symposium on Neural Networks*, pages 222–231. Springer, 2015.

[215] Zhenglong Zhou, Bo Shu, Shaojie Zhuo, Xiaoming Deng, Ping Tan, and Stephen Lin. Image-based Clothes Animation for Virtual Fitting. In *SIG-GRAPH Asia 2012 Technical Briefs*, page 33. ACM, 2012.

[216] Hongyuan Zhu, Fanman Meng, Jianfei Cai, and Shijian Lu. Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation*, 34:12–27, 2016.

[217] Long Zhu, Yuanhao Chen, Yifei Lu, Chenxi Lin, and Alan Yuille. Max margin and/or graph learning for parsing the human body. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[218] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. Technical report, arXiv:1705.01389, 2017. https://arxiv.org/abs/1705.01389.

[219] Silvia Zuffi and Michael J Black. The stitched puppet: A graphical model of 3d human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3537–3546, 2015.

[220] Silvia Zuffi, Oren Freifeld, and Michael J Black. From pictorial structures to deformable structures. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3546–3553. IEEE, 2012.