

Shape Models of the Human Body for Distributed Inference

by

Silvia Zuffi

Sc. M., University of Bologna, 1995

Sc. M., Brown University, 2010

A Dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island

May 2015

© Copyright 2015 by Silvia Zuffi

This dissertation by Silvia Zuffi is accepted in its present form
by the Department of Computer Science as satisfying the
dissertation requirement for the degree of Doctor of Philosophy.

Date _____

Michael J. Black, Advisor

Recommended to the Graduate Council

Date _____

Erik B. Sudderth, Reader

Date _____

Deva Ramanan, Reader

Approved by the Graduate Council

Date _____

Peter M. Weber, Dean of the Graduate School

Vitæ

Education

Master degree in Electronic Engineering at the Faculty of Electronic Engineering of the University of Bologna, Bologna, Italy, 1995. Title (in Italian): “Applicazione di un benchmark di elaborazione delle immagini ad una macchina parallela floating point e analisi delle prestazioni”, translated in: “Application of the DARPA image understanding benchmark to a parallel computer with SIMD floating point architecture and performance analysis”. Supervisor: Prof. Tullio Salmon Cinotti, final mark 100/100 *magna cum laude*.

Master degree in Computer Science, Brown University, Providence, RI, 2010.

Fellowships

EU COMETT fellowship, 1995.

EU-7FP Marie Curie IOF, grant agreement 235633 BoCap Bone motion Capture from video sequences, 2008 (*declined*).

Work Experience

Feb 1999 — August 2009. Institute for Technologies of Construction, ITC-CNR, (previously Institute for Multimedia Technologies ITIM-CNR), National Research Council, Milano, Italy. Research Scientist on color imaging and reproduction.

January 1998 — February 1999. Movement Analysis Lab., Research Institute Codivilla-Putti, Istituti Ortopedici Rizzoli, Bologna, Italy. Freelancing Researcher for the development of a system for the 3D pose estimation of total knee prosthesis from single frame fluoroscopy.

November 1995 — December 1997. Boconsult I.d.S. S.p.A., Bologna, Italy (now Ducati Sistemi S.p.A.). Software Engineer.

May 1995 — October 1995. Bazis, Leiden, The Netherlands. EU COMETT

fellowship for the design and implementation of the Web portal for the EU project CAMIREMA (Case Mix And Resource Management).

Publications relevant to this Thesis

- (1) **S. Zuffi**, M.J. Black, “The Stitched Puppet: a Graphical Model of 3D Human Shape and Pose”, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE, Boston, MA, June 2015 (*accepted for oral presentation*).
- (2) J. Pacheco, **S. Zuffi**, M. J. Black, and E. Sudderth, “Preserving Modes and Messages via Diverse Particle Selection”, In Proceedings of the 31st International Conference on Machine Learning (ICML), J. Machine Learning Research Workshop and Conf. and Proc., volume 32, pages 1152-1160, Beijing, China, June 2014.
- (3) **S. Zuffi**, J. Romero, C. Schmid, M. J. Black, “Estimating Human Pose with Flowing Puppets”, IEEE International Conference on Computer Vision (ICCV), pages 3312-3319, Sydney, Australia, December 2013.
- (4) H. Jhuang, J. Gall, **S. Zuffi**, C. Schmid, and M. J. Black, “Towards understanding action recognition”, IEEE International Conference on Computer Vision (ICCV), IEEE, pages 3192-3199, Sydney, Australia, December 2013.
- (5) **S. Zuffi**, O. Freifeld, M.J. Black, “From Pictorial Structures to Deformable Structures”, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE, pages 3546-3553, Providence, RI, June 2012.

- (6) O. Freifeld, A. Weiss, **S. Zuffi**, M.J. Black, “Contour People: A Parameterized Model of 2D Articulated Human Shape”, IEEE Conf. on Computer Vision and Pattern Recognition, (CVPR), IEEE, pages 639-646, San Francisco, CA, June 2010.

Dedicated to my family

Acknowledgements

The first person I would like to thank is Michael Black, my advisor. All the work in this thesis would have not been possible without him. And not just because of his brilliant ideas, but for his constant, unconditioned support and infinite patience. Michael is a great guide. Like a mountain guide that has a lot of experience, he helps you enjoying the walk, gives you freedom to explore safely other paths along the journey, but he knows you have to make it to the top in a reasonable amount of time. It has been a real privilege to be advised by him.

I want to thank my family, my mom Martina who has always been so excited for my life in the US, and my sisters Elisa and Claudia.

I also want to thank Shree Nayar, as I believe all started when he hosted me in his lab many years ago.

In my years at Brown I met so many great people. I want to thank Erik Sudderth, who taught me machine learning. I really enjoyed meeting with him and Jason and discuss in front of a whiteboard. I have the suspect he already knew all the answers to the problems we were trying to solve.

Thanks to Lauren Clarke and Genie deGouveia for always being so kind and helpful.

I feel I have been lucky to be at the CS department when Olya, Alexandra, Irina, Foteini, Gen, Anna, Aggeliki, Layla, Sasha were there. I would like to thank them all for being such nice friends. I also want to thank my "extended" office mates: Soumya, Mike, Jason, and Dae Il.

I have learned a lot from the other students and people working with Michael: Deqing, Oren, Alex, Eric, Matt, Javier and Gerard. I also want to thank my latest office mates for sharing wisdom, food and jokes during deadlines: Laura, Martin,

Jonas, Naejin, Cristina; Melanie for all the support and for bringing Rocko in the office every day; my Tuebingen friends Jessica, Sandra and Christian.

And finally a thank you to Andreas, for the support during the writing of the thesis.

Contents

Vitæ	iv
Dedication	vii
Acknowledgements	viii
List of Tables	xiii
List of Figures	xiv
Chapter 1. Introduction	1
1. Contribution	9
2. List of Papers	12
Chapter 2. Related Work	13
1. Models of the Human Body	13
1.1. 2D Body Models	13
1.2. 3D Body Models	17
2. 2D Human Pose Estimation from Single Images	22
2.1. Methods based on Pictorial Structures	23
2.2. Other Methods	29
3. 2D Human Pose Estimation from Video Sequences	31
4. 3D Human Pose and Shape Estimation from 3D Data	32
Chapter 3. Deformable Structure Model	37
1. Definition	37
2. Learning	38
3. Shape Representation	41
4. Pairwise Relationship of Connected Parts	43

5. Generating Model Instances	44
Chapter 4. Human Pose Estimation from Single Images with the Deformable Structures Model	47
1. Inference	48
2. Likelihood	53
3. Image Annotation Tool	56
4. Experiments	59
Chapter 5. Pose Estimation on Video Sequences with the Deformable Structures Model	65
1. Exploiting Optical Flow	65
2. Flowing Puppets	70
3. Image Likelihood	72
4. Inference	74
5. Experiments	77
Chapter 6. Limitations of the Deformable Structures Model	83
1. Multi-view DS Model	83
Chapter 7. Stitched Puppet Model	88
1. Definition	90
2. Learning	95
3. Stitching Potentials	99
4. Generating Model Instances	101
Chapter 8. 3D Mesh Alignment with the Stitched Puppet Model	105
1. Method	106
2. Experiments	115
3. Pose and Shape from Visual Hull	123
Chapter 9. Conclusion	125

List of Tables

- 1 Results (PCP) from [111] for DS model without likelihood (NL), DS model with a fixed shape (NS), and full DS model (DS), with shape variation. For comparison with other methods we report results from [85] (CPS) and Yang and Ramanan (Y&R) [107]. The results reported for Eichner et. al. are taken from [85]. Baseline (PS) is [5]. 61
- 2 Results from [67]. PCP scores for lower arms and average PCP with threshold 0.5 on single and multi-person images. Detection rate for the multi-person partition is the proportion of people for which the head or torso is correctly detected. 63

List of Figures

1	(left) Variability in human poses (©Coco Rocha/Steven Sebring/Study of Pose), (right) variability in body shape and appearance (©2011 - Universal Pictures)	1
2	Organization of shape information in a 3D model [61].	5
3	Pictorial Structures (PS) model (left) and Contour People (CP) model (right). PS is part-based and each part is represented with an independent rectangle. CP is a <i>global</i> model, and represents the body with a closed contour. It admits segmentation into parts (here color coded), but it is not part-based, as each part cannot be independently generated.	6
4	Example of likelihood score map for the PS model. On the right is a score map for the location of the part <i>head</i> in vertical pose.	8
1	The Human Puppet model [44].	13
2	The Cardboard People model [52].	14
3	Pictorial Structures [32].	14
4	Pictorial Structures model of the human body [29].	15
5	Flexible Mixtures of Parts model [107] (right) compared to Pictorial Structures (left).	15
6	Contour Person (CP) model [34]. The first three principal components of a gender-neutral model. For each component, from left to right: -3, 0, 3 standard deviations from the mean in the direction of the respective component.	16

7	Contour Person (CP) model [34]. Examples of the model fit to silhouettes of real people.	16
8	Samples from the model of [27].	17
9	Eigen clothing [37]. The blue contour is always the same naked shape. The red contour shows the mean clothing contour (a) and 3 std from the mean for several principal components (b)-(d).	18
10	The Generalized Cylinder model [1] applied to a Barbie doll.	18
11	Figure reproduced from [81], showing input data (left) and model in estimated pose (right). Note that the 3D model is part-based but parts are rigid and do not connect at joint interfaces.	19
12	(left) The Loosed-limbed model [90]. (center) Super quadrics model [92]. (right) The Sum of Gaussians model [93].	20
13	The SCAPE model [8].	21
14	Articulated models of the human body. The models presented in this thesis are Deformable Structures and Stitched Puppet.	22
15	Human pose estimation. Output can be a set of sticks, body joints, or values for model variables of location and rotation of boxes or 3D shapes.	23
16	Visualization of the discriminative model of [107]. The four models correspond to the best scoring locations for the parts for a model with mixtures of four components. The rectangular patches represent the weights of the HOG template that is used to compute image likelihoods.	28
17	Scans from the CEASAR dataset after hole-filling [3].	35
1	Examples of female DS models in different poses.	37
2	Contours generated from the SCAPE model. The rendered 3D meshes are SCAPE models in two similar poses; the contour-based representations are obtained by per-part projections of the meshes on the image plane, and then extracting the contour from silhouettes.	40

3	Examples of training poses. Note the variability in pose as well as in camera location.	41
4	DS part deformations. (left) Deformations for three example parts. Black is the mean contour. Red and blue are ± 2 standard deviations from the mean along the first 3 principal component directions. Stars mark the joint locations that deform with the contour. (right) Mean part shapes for the female and male body (14-part model). The dots represent joint points (see text).	43
5	Samples from a DS model with 10 shape variables for each part. The colored parts correspond to the mean of the conditional models we obtain given the torso shape. Dotted lines represent samples obtained by sampling the conditional distributions for the upper arms and legs, and then generating the corresponding lower arms and legs, respectively.	46
1	PS to DS. Examples of estimates from a PS model (yellow) [107] mapped to DS (magenta).	51
2	Contour likelihood. The image shows the location of the HOG cells along a limb contour. Cells are located on the boundary, just inside, and just outside the part.	53
3	HOG descriptors are steered to the contour orientation. (a) DS annotation. (b) HOG visualization.	55
4	Likelihood examples. (a) Random examples for the torso. (b) Plot of the likelihood value for the random samples vs. distance computed as overlap score from the ground-truth annotation; note that here the ground-truth point (for which distance is zero) has the highest likelihood. (c) Sample with highest likelihood.	55
5	Contour-based likelihood features. HOG descriptors are computed at contour points (red), inside (green) and outside the contour (blue) in a 3-level pyramid.	56

6	Examples from the DS annotation tool.	59
7	Estimated body pose from [111]– examples where the DS model improves on the PS baseline. DS is solid red, PS is dashed green.	62
8	Estimated body pose from [111]– representative failure cases. DS is solid red, PS is dashed green.	63
9	Results from [67] on Buffy Dataset. (top) Single person images showing a MAP estimate (red) with different solutions for the arm. (bottom) The full set of particles at the final iteration of D-PMP (top). Best pose in the set of retained hypotheses (bottom, red).	64
1	Optical Flow. The second row shows the forward optical flow of the sequence of frames in the first row. In the optical flow images the hue corresponds to the direction as indicated in the image on the right. The saturation corresponds to flow magnitude.	66
2	Flowing puppets. (a) Frame with a hypothesized human “puppet” model; (b) The prediction of the puppet from (a) into the adjacent frame using the puppet flow; (c) Dense flow between frame (a) and its neighboring frames; (d) The flow is approximated by an affine background layer and the motion of a foreground puppet (the puppet flow).	67
3	Puppet flow. (left) Frame with overlapped a DS model; (center) dense optical flow; (right) puppet flow.	68
4	DS puppet layer. (1) Frame; (2) Corresponding puppet layer with parts ordered by fixed order. The warmer the color, the closer to the camera.	71
5	Contour-based likelihood features. HOG descriptors are steered to the contour orientation and computed at contour points (blue), inside (red) and outside the contour (green) in a 3-level pyramid.	73
6	Hand detection. Example of output from the hand detector trained on optical flow. Image (left), optical flow (center), and hand probability map defined from running a flow-based hand detector on the flow (right).	74

- 7 Hand detection. Examples of output from hand detector trained on image cues. The red box is the detection with top score; the missing side indicates the position of the wrist. 74
- 8 Particle-based optimization. Particles are initialized on each frame (first row), then the M best are propagated through the flow forward and backward (second row). For a defined number of iterations particles are then locally optimized, then the M best are propagated to the neighbors (third and fourth row). Then the best particle on each frame is returned as the solution (last row). 78
- 9 Results. Accuracy of elbow (left) and wrist detection (right) for different threshold distances from ground truth. We significantly improve over our baseline (Yang and Ramanan [107]) and over the state-of-the-art (Sapp et al. [86]) in wrist detection. FP stands for Flowing Puppet. 79
- 10 Estimated body pose. Successful detection results from 9 test clips are shown (2 frames per clip). Images are shown with the estimated puppet overlaid in white. Below each image is the estimated forward flow field color coded as in [10] with the puppet overlaid in black. 82
- 11 Estimated body pose. Examples of failure cases. In all cases the image evidence supports incorrect poses. 82
- 1 Multi-view DS model. Each figure shows the mean torso for each mixture component, corresponding to 9 discrete camera viewpoints uniformly spaced between the West and East direction (S = South). 86
- 2 Multi-view DS model. DS puppet generated from the mean torso of each mixture component. Note how the orientation of the feet are correctly generated, even if the viewpoint is only accounted for at the torso level, with a view-dependent shape (Figure 1). The shape of the torso determines the appropriate shape for the upper legs (see text). 87

- 1 3D Body Models. (a) A SCAPE body model [8] realistically represents 3D body shape and pose using a single high-dimensional state space. (b) A graphical body model composed of geometric primitives connected by pairwise potentials (image reproduced from [91]). (c) The **stitched puppet model** has the realism of (a) and the graphical structure of (b). Each body part is described by its own low-dimensional state space and the parts are connected via pairwise potentials that “stitch” the parts together. 89
- 2 Region-based face model (image and caption reproduced from [95]). Face posing using interactive region-based (b) and holistic (d) face models. The models drive the human character shown in (a). User-given constraints (black markers) create a wink with a smirk, when issued to the region-based model (b and c). In contrast, the same constraints produce uncontrolled global deformations when the holistic model is used (d and e). 90
- 3 Stitched Puppet Model. To generate an instance of SP we start with the template body (top left), which is segmented into parts. We apply the intrinsic shape deformation to change the body shape (top right). We generate pose deformations for each body part (see text) (bottom left). The pose of the body is defined by the rotation and translation that stitches the parts together (bottom, middle and right). 91
- 4 PCA models. Normalized cumulative variance for the PCA models of the female torso (top) and lower arm (bottom) for shape (left) and pose (right). 93
- 5 Pose-dependent part deformation. An example of the torso PCA model is shown (first three principal components from top to bottom). The center mesh in each row is the mean shape and the left and right meshes correspond to ± 3 standard deviations from the mean. 96
- 6 Training samples. Examples of few training samples from the dataset with variation in pose. Meshes are all defined in a local coordinate system with origin in the middle of the body part. 97

- 7 Intrinsic shape part deformation. An example of the torso PCA model is shown (first three principal components from top to bottom). The center mesh in each row is the mean shape and the left and right meshes correspond to ± 3 standard deviations from the mean. 98
- 8 Pose-dependent part deformation. An example of the head PCA model is shown (first three principal components from top to bottom). The center mesh in each row is the mean shape and the left and right meshes correspond to ± 3 standard deviations from the mean. 99
- 9 Intrinsic shape part deformation. An example of the head PCA model is shown (first three principal components from top to bottom). The center mesh in each row is the mean shape and the left and right meshes correspond to ± 3 standard deviations from the mean. 100
- 10 Pose-dependent part deformation. An example of the upper left leg PCA model is shown (first three principal components from top to bottom). The center mesh in each row is the mean shape and the left and right meshes correspond to ± 3 standard deviations from the mean. 101
- 11 Intrinsic shape part deformation. An example of the upper left leg PCA model is shown (first three principal components from top to bottom). The center mesh in each row is the mean shape and the left and right meshes correspond to ± 3 standard deviations from the mean. 102
- 12 Stitching parts illustration. Parts can be thought of as being connected by springs between the interface points. When the model fits together seamlessly, this stitching cost is zero. During inference, the parts can move apart to fit data and then the inference method tries to infer a consistent model. 103
- 13 Example of SP bodies. Several bodies generated using the SP model. Note the realism of the 3D shapes. 104
- 1 FAUST test scans for one subject. Goal of the *intra-subject* challenge is to bring pairs of these scans into correspondence. 105

2	Example of particles initialization: note that particles are initialized from disconnected SP models randomly sampled. The red arm on the right belongs to the FAUST scan around which the random puppets are generated.	109
3	D-PMP optimization. Inference with 30 particles for 60 iterations. From top to bottom, left to right: initial particles; scan (red) and current solution (light blue) at various steps; the final set of particles. At the end a greedy algorithm resamples all the particles around the current solution.	112
4	Local Levenberg-Marquardt (LM) optimisation. Plot of the average alignment error over a subset of the FAUST training set for different values of the steps in the local LM optimization.	114
5	Intra-subject challenge. Histograms of the errors.	116
6	Inter-subject challenge. Histograms of the errors.	116
7	Intra-subject challenge. One of the worst results, which is due to a mistake in associating scan points to the model in case of contact points. Note in the figure at the bottom left how some points on the right lower arm have been associated with the left hand.	117
8	Intra-subject challenge. Another bad result, which is due to a wrong association between scan points and model vertexes when there are contact points. Note in the figure at the bottom right how the hands have different colors, illustrating the confusion between left and right hands.	118
9	FAUST errors. Examples of mistakes in the alignment of the scan (red) to the model (light blue). (left) The hands are matched to the lower arms; (middle) the lower arm and hand are matched to the upper arm; (right) the hands are not accurately matched.	119
10	Alignment on FAUST. We show the test scan in red and SP in light blue.	120
11	Alignment on FAUST. We show the test scan in red and SP in light blue.	121

- 12 Alignment on FAUST. We show the scan (red) and model (light blue) for the different subjects in the test set to illustrate the quality in the estimation of the body shape. 122
- 13 Alignment to visual hull data. We show the visual hull data in red and SP in light blue. 124
- 14 Alignment to visual hull data. Failure cases; here the subject is almost upside-down and the optimization cannot estimate a proper orientation for the body. 124

Abstract of “Shape Models of the Human Body for Distributed Inference”

by Silvia Zuffi, Ph.D., Brown University, May 2015

In this thesis we address the problem of building shape models of the human body, in 2D and 3D, which are realistic and efficient to use. We focus our efforts on the human body, which is highly articulated and has interesting shape variations, but the approaches we present here can be applied to generic deformable and articulated objects. To address efficiency, we constrain our models to be part-based and have a tree-structured representation with pairwise relationships between connected parts. This allows the application of methods for distributed inference based on message passing. To address realism, we exploit recent advances in computer graphics that represent the human body with statistical shape models learned from 3D scans.

We introduce two articulated body models, a 2D model, named Deformable Structures (DS), which is a contour-based model parameterized for 2D pose and projected shape, and a 3D model, named Stitchable Puppet (SP), which is a mesh-based model parameterized for 3D pose, pose-dependent deformations and intrinsic body shape.

We have successfully applied the models to interesting and challenging problems in computer vision and computer graphics, namely pose estimation from static images, pose estimation from video sequences, pose and shape estimation from 3D scan data.

This advances the state of the art in human pose and shape estimation and suggests that carefully defined realistic models can be important for computer vision. More work at the intersection of vision and graphics is thus encouraged.

CHAPTER 1

Introduction

The ability to automatically extract information about people from images or videos is arguably one of the most interesting and challenging problems in computer vision. Human bodies are very complicated objects: they vary in appearance depending on clothing, body shape, age and gender. They are highly articulated, and can appear in images and videos in a variety of poses, often interacting with other people or objects in various ways (Figure 1). Despite this complexity, a human observer is very effective in using visual cues to recognize people, inferring what they do, how they are dressed, and sometimes their emotions or intentions.

Recognizing people entails estimating their pose and appearance. From this information, action and identity can then be derived. Traditionally the problem of pose estimation has received more attention, the reason being that inferring body pose is a basic component in socially important applications, like video surveillance for detecting inappropriate behavior, for monitoring patients at home or elderly people in



FIGURE 1. (left) Variability in human poses (©Coco Rocha/Steven Sebring/Study of Pose), (right) variability in body shape and appearance (©2011 - Universal Pictures)

their daily activities, or for interpreting sign language. Applications also exist in the entertainment industry, for example for retrieving videos from movie collections based on pose similarity, or for human motion capture using only visual data. Recognizing human appearance, and describing body shape, hair or clothing is relevant to person re-identification tasks, online shopping, and potentially in many social network applications.

In addition to tasks related to images and videos, with the recent availability of technology for capturing 3D data, computer vision has embraced a new set of problems involving the estimation of attributes from 3D or depth images. 3D scanning technologies allow building accurate models of the human body that can be used to define new applications, and to approach existing problems with novel questions and methods.

In this thesis we are mostly interested in human pose estimation from images, videos and 3D data. We will consider tasks that involve the whole human body, thus avoiding methods that concentrate on individual parts, like faces or hands. Also, we will not address tasks at a fine level of body detail, like hand pose estimation. And finally, we will not consider tasks that treat the human body as an atomic element, like the analysis of crowds.

Approaches for human pose estimation can be divided into model-based methods, where the goal is to match a given or learned model of the human body to image or video data to infer the model's variables, or holistic methods that aim at estimating attributes of the body, for example joints locations, with a direct mapping from the input data. These latter techniques have received attention recently due to the popularity of deep neural network architectures. In this thesis we focus on model-based approaches.

A *model of the human body*, or more generically of an object, is the basis of a generative process that, given values for the model variables, produces a domain-specific representation. In computer vision, object models produce visual representations that resemble the appearance of the object in an image. A model has parameters

and variables. Parameters are quantities that typically specify attributes that we are not interested in estimating from test data, or that we cannot estimate at test time for lack of information or for complexity issues. Parameters can be manually set or learned from training data. Variables are the quantities we are interested in automatically estimating from test data. For example, in the case of the human body, we can have parameters specifying the limb size and variables specifying the body pose. We can learn an average size of the limbs from training data and estimate the pose of people on test images.

Building a full model, able to generate a faithful appearance of the object, can be very hard. Arguably the human body is one of the hardest objects to model, due to its own variability. But building a full model for human bodies in images, that can generate the object appearance up to the pixel values, is even harder. Parameters that do not belong to the model play a significant role. For example, the direction of the light, the position of the camera, and the presence of occluding objects.

A *generative model* assumes a probability distribution over the model’s variables. Let \mathbf{l} be a vector of model’s variables, and Θ a vector of model’s parameters. We can generate samples from the model by first sampling from the distribution over the variables, $p(\mathbf{l}|\Theta)$ and then using the obtained value to generate the corresponding model output, or sample. The distribution over a model’s variables is called the *prior*, as it models what we know about the distribution of the variables before seeing any data. Given data D , we can define a *likelihood* distribution that represents the probability of the data given model’s variables and parameters $p(D|\mathbf{l}, \Theta)$. The posterior distribution over the model’s variables given data and parameters is then:

$$(1) \quad p(\mathbf{l}|D, \Theta) = \frac{p(D|\mathbf{l}, \Theta)p(\mathbf{l}|\Theta)}{p(D|\Theta)}.$$

The process of computing the posterior distribution of Eq. 1 is called *probabilistic inference*. We are interested in estimating the posterior $p(\mathbf{l}|D, \Theta)$ as it encodes all the uncertainty about the model and the likelihood, and we can sample the posterior distribution to generate candidate solutions. Often it is hard to compute the full

posterior $p(\mathbf{l}|D, \Theta)$, while it is feasible to estimate its maximum. With *maximum-a-posteriori*, or MAP estimation, we obtain (often approximatively) the most probable solution $\hat{\mathbf{l}} = \arg \max_{\mathbf{l}} p(\mathbf{l}|D, \Theta)$. This is typically not accomplished by computing $p(\mathbf{l}|D, \Theta)$ and then finding its maximizer, but solving an optimization problem:

$$(2) \quad \hat{\mathbf{l}} = \arg \max_{\mathbf{l}} p(D|\mathbf{l}, \Theta)p(\mathbf{l}|\Theta),$$

where we have dropped the denominator as it does not depend on \mathbf{l} . If we take the negative logarithm of Eq. 2, we obtain:

$$(3) \quad \hat{\mathbf{l}} = \arg \min_{\mathbf{l}} (-\log p(D|\mathbf{l}, \Theta) - \log p(\mathbf{l}|\Theta)).$$

We can see the minimisation in Eq. 3 as an energy minimization problem, where we have a *data term* $E_d(\mathbf{l}, \Theta) = -\log p(D|\mathbf{l}, \Theta)$ and a *model term* $E_m(\mathbf{l}, \Theta) = -\log p(\mathbf{l}|\Theta)$. MAP estimation has the disadvantage that all the uncertainty encoded in the prior and the likelihood distributions is lost, and only one solution is obtained. Often the posterior distribution is multi-modal, and it would be useful instead to obtain a set of solutions corresponding to the highest modes of the posterior.

From a historical perspective, since the first attempts to build models of objects with simple geometric attributes, researches have paid special attention to modeling the human body. Maybe because in art we are very used to seeing representations of the human body at various levels of abstraction, it feels natural to be able to represent visual cues of the body that are responsible for our perception of human pose and shape in images.

A notable example of a simple geometrical model comes from Marr and Nishihara [61], who depict the human body as a collection of cylinders (Figure 2). Interestingly, they propose a hierarchical representation that has been rarely exploited so far: the most common models are not hierarchical, and are built for a task at a specific scale.

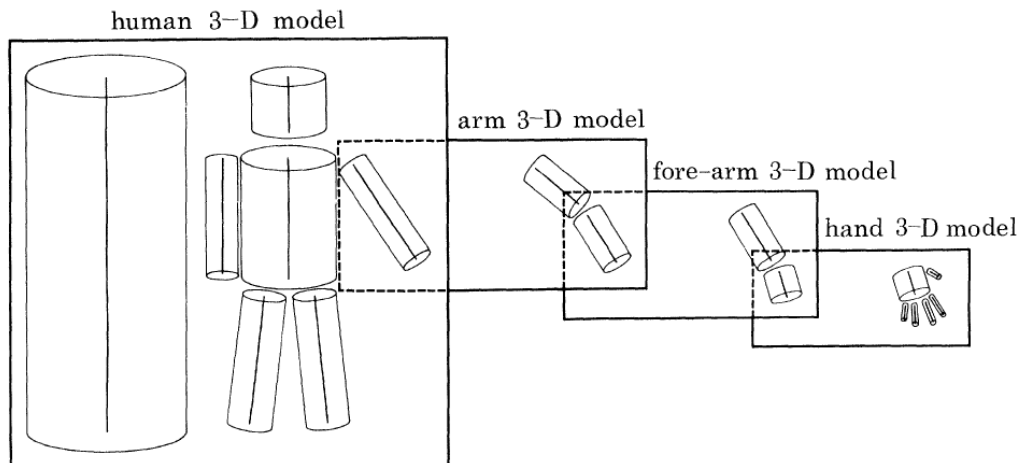


FIGURE 2. Organization of shape information in a 3D model [61].

While old models were simple arguably for technological reasons, the appeal to geometric primitives is dominant in all body models proposed so far for pose estimation. In particular, the most popular model today for 2D pose estimation from single images, the Pictorial Structures (PS) model [30], represents the body as a collection of rectangles. In other words, a human body, that we would draw as a set of curved lines enhancing the curvature of the joints, is depicted as a *puppet*, similar to what Hinton imagined in 1976 [44].

In a puppet model, an articulated body is represented as a set of connected parts that can rotate and sometimes slightly translate at the connecting joints. The model's variables and parameters are the set of part variables and parameters. It is what it is called a *part-based* representation, because each part can be independently generated given its variables and parameters. This is in contrast to *global* models, where to render a part we need to know all model's variables, for example the whole body pose.

Part-based models of interest for this thesis map to a graphical model representation where each body part is represented with a node. Typically pairwise connections between nodes are assumed, and the graph is a tree.

Let \mathbf{l}_i be the set of node variables for the node i that describe its pose with respect to an arbitrary global reference system. In this section we focus our description on

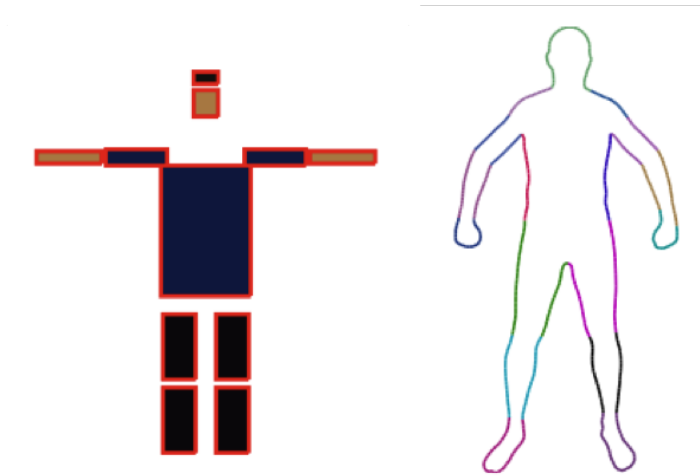


FIGURE 3. Pictorial Structures (PS) model (left) and Contour People (CP) model (right). PS is part-based and each part is represented with an independent rectangle. CP is a *global* model, and represents the body with a closed contour. It admits segmentation into parts (here color coded), but it is not part-based, as each part cannot be independently generated.

2D models, but as we will illustrate later in this thesis, the same framework holds in 3D. In 2D typically \mathbf{l}_i includes the x and y image coordinates of the part center, its rotation angle on the image plane, and sometimes a scale parameter. The scale parameter can model foreshortening if it is applied only to the height of the part, measured as length along the part “bone”, or can model distance from the camera if applied to both width and height. Let the data be an image I . Then, $p(I|\mathbf{l}_i)$ denotes the image likelihood, and $p(\mathbf{l}_i, \mathbf{l}_j)$ a pairwise joint probability over variables of node i and node j . The posterior distribution over the model variables given image data and the model’s parameters can be written as:

$$(4) \quad p(\mathbf{l}|I, \Theta) \propto \prod_{i=1..n} p(I|\mathbf{l}_i) \prod_{(i,j) \in E} p(\mathbf{l}_i, \mathbf{l}_j | \Theta_{ij})$$

where Θ are models’s parameters and E is the set of edges in the graph. Note that the model is defined as a factored prior distribution with $p(\mathbf{l}|\Theta) \propto \prod_{(i,j) \in E} p(\mathbf{l}_i, \mathbf{l}_j | \Theta_{ij})$.

The advantage of such a factored representation is that the large state space of articulated objects can be explored locally at each part. In a part-based model each body part can be rendered on the image independently from the other parts, and thus the factored form of Equation 4 is possible, as the part likelihoods are independent. Inference or optimization methods based on message passing can then be applied to efficiently evaluate any combination of part hypotheses. Here is the advantage with respect to global models for which this factorization is not possible: a wider space of solutions can be explored.

Assuming it is possible to enumerate all the possible states of each body part, the likelihood of each part, for each state, can be pre-computed and stored. Model hypotheses can then be efficiently evaluated over all the state space for global inference. Figure 4 shows an example of likelihood score map for the location of the head in vertical pose: for each body part a set of score maps like the one displayed can be generated for a set of discretized rotations of the part (assuming fixed scale). Clearly there are two requisites for this to be feasible: a small discretized state space and fast-to-compute likelihoods. This is one of the reasons why the most popular part-based model today, Pictorial Structures (Figure 3, left), limits the number of model variables to pose variables. It avoids representing the variability of body shape, either for anatomic shape differences, or for shape deformations with pose. It can efficiently explore the space of all possible poses via dynamic programming, and today it is the prevailing approach for human pose estimation in uncontrolled images, with cluttered background, unknown subject and no initialization.

Another reason for the fact that today’s models are not very different from the simple ones introduced 40 years ago is that building a realistic generative model of the human body, which is able to generate any person in any pose in uncontrolled conditions, is hard.

In the seminal work of Felzenswalb and Huttenlocker [30], where the probabilistic view of Pictorial Structures illustrated in Eq. 4 was introduced, the likelihood was based on contour matching of the model with the image, and the application of



FIGURE 4. Example of likelihood score map for the PS model. On the right is a score map for the location of the part *head* in vertical pose.

the model was limited to images with negligible background clutter to allow reliable silhouette extraction. Current methods based on part-based models for human pose estimation aim at working on uncontrolled images, with cluttered backgrounds. In a full generative approach one should therefore be able to represent the variability of human appearance in uncontrolled conditions. Clearly this is a very hard problem. An alternative, introduced by Andriluka et al. [5], is to learn likelihoods as *discriminative* scoring functions. Image evidence is modeled by computing image features within the bounding boxes associated to the body parts, and then computing scoring functions over features. This is today the prevailing approach. The popularity and representation power of patch descriptors like HOG [23] has in some way diverted the attention from the shape and appearance modeling problem, as the rectangular body parts in PS models are not meant to match human body shape, but to constitute a suitable support region for computing features. More recently, convolutional neural networks are used to learn the features that describe body joints appearance [99]. The design of better generative models for representing human body shape is a research direction that has not encountered interest of the scientific community so far. This is the main topic of this thesis.

As previously stated, if we only model pose, like in PS, we have the advantage of a small state space, which we can discretize and then apply algorithms for discrete inference, namely discrete belief propagation, to output posterior marginal distributions for each part, or a global MAP solution. If we build sophisticated models that

more accurately represent the human body, we need additional variables. Modeling intrinsic body shape, which is the shape due to the bone and muscles structure and fat distribution, would require additional shape variables. The same for modeling pose-dependent shape deformations, which are the shape changes that occur when we move our body and are due to muscle and soft tissue contraction and extension. Modeling clothes or hairstyle would probably require a set of discrete variables representing types of clothes and hair cuts. Some of these attributes would be local to a body part, or global to the person. A more sophisticated model would probably involve image likelihoods too expensive to evaluate at each state space location. Discretization would create a too large state space, and global inference or optimization would not be possible anymore. The question now is: can part-based models also be effective if only approximate or local inference is possible? Would it be feasible to keep the advantage of the efficient solution space exploration of the part-based framework and build more accurate generative models?

The hypothesis of this thesis is that more accurate shape models are powerful representations for 2D and 3D pose estimation, despite the challenges for inference for such complex models. The advantages of a more accurate shape representation is in the possibility of defining new types of likelihoods, based on contour alignment, or color statistics within the foreground body region, that can be better expressed by a contour model than with a simple box-based representation. Also, modeling shape opens up to a wider range of applications for a human body model, where in addition to pose also shape can be estimated.

1. Contribution

In this thesis we pursue the goal of building models of deformable articulated objects that can represent pose-dependent shape deformations. We focus our attention on models of the human body, for which we can exploit existing accurate statistical shape models to generate training data. Differently from existing accurate models of the human body, our 2D and 3D *puppets*, Deformable Structures (DS) and Stitched

Puppet (SP), are part-based representations that can exploit existing algorithms for inference in factored models. The motivation for our work is our interest in filling the void between simple part-based models used in computer vision, like Pictorial Structures (PS) [30], and complex global models used in computer graphics, like SCAPE [8]. With complex shape models we cannot perform global inference with discrete message passing algorithms, as the state space of the model cannot be discretized anymore, being too high dimensional. We demonstrate with our experiments that our models can be successfully applied to hard problems, without good or manual initialization.

The models we propose, in particular the 2D model, appear as a very abstract representation that, as human observers, we can easily associate with a human body. But building effective automatic methods that can match an abstract representation to real images content, with people dressed arbitrarily, with hair, with occlusions, uncontrolled light conditions and unknown camera is probably a task even harder than building the model itself. To this aim we propose a contour-based likelihood based on HOG descriptors.

In a different approach, we could keep the matching strategy simple and change the level of abstraction of the image data. For example, if we were able to convert the pixels into foreground and background masks, or to reliably extract semantic contours from the image, we would have an easy job in fitting the model to data. Here the shape model would be superior to a simpler model with limbs of fixed, rectangular shape. We apply this approach to exploit DS to interpret abstract image information in the task of human pose estimation from video sequences. There, we use DS as a foreground mask for dense optical flow, using optical flow for *flowing* the DS puppet across frames.

In DS, the shape variables do not represent only intrinsic attributes of a human body. In 2D, the projected shape is the effect of the body intrinsic shape, articulated pose, the relative position between the body and the camera, and the camera

parameters. We learn the DS model for the shape variables to represent the deformations for pose and camera for a person of fixed intrinsic body shape. We design a 3D model, named Stitched Puppet (SP), where we instead include intrinsic shape variables, meaning that with SP we can generate samples from the model for people of different height, weight and body type. Clearly intrinsic shape is a global attribute of the body, but we show it can be considered as a local attribute of a part, and reliably estimated. We apply the 3D model to a task where SP is a full model for the input data, the FAUST challenge, where the goal is mesh alignment between 3D scans of people in arbitrary poses. A characteristic of FAUST is that the training set is relatively small to learn powerful discriminative approaches, while we can exploit the efficient exploration of the pose space of our 3D part-based model to successfully fit SP to almost all the test scans without exploiting any prior information about the most likely poses.

In summary our contributions are:

- (1) We introduce Deformable Structures (DS), a 2D contour-based model of the human body with realistic pose-dependent deformations. We define a novel contour-based image likelihood based on HOG descriptors for human pose estimation from images.
- (2) We apply DS to the problem of pose estimation in video sequences. We introduce a novel approach to exploit computed optical flow, that we consider an observation. Leveraging the region-based representation of the DS model, where each part is a closed contour, we define *flowing puppets* as DS models that move in time, with articulated motion, driven by optical flow. With flowing puppets we can propagate solutions from frame to frame across the video sequence without relying on motion prior models.
- (3) We introduce Stitched Puppet (SP), a 3D mesh-based model of the human body that can represent pose-dependent deformations and people of different

intrinsic shape. We apply SP to the problem of pose and shape estimation from 3D scans of people, where we can fit the model to data without any prior on the most likely poses.

2. List of Papers

Papers in the thesis:

- (1) **S. Zuffi**, J. Romero, C. Schmid, M. J. Black, “Estimating Human Pose with Flowing Puppets”, IEEE International Conference on Computer Vision (ICCV), pages 3312-3319, Sydney, Australia, December 2013.
- (2) **S. Zuffi**, O. Freifeld, M.J. Black, “From Pictorial Structures to Deformable Structures”, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE, pages 3546-3553, Providence, RI, June 2012.
- (3) **S. Zuffi**, M. J. Black, “The Stitched Puppet: a Graphical Model of 3D Human Shape and Pose”, *accepted for oral presentation at* IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, MA, June 2015.

Papers relevant to the thesis:

- (1) O. Freifeld, A. Weiss, **S. Zuffi**, M.J. Black, “Contour People: A Parameterized Model of 2D Articulated Human Shape”, IEEE Conf. on Computer Vision and Pattern Recognition, (CVPR), IEEE, pages 639-646, San Francisco, CA, June 2010.
- (2) H. Jhuang, J. Gall, **S. Zuffi**, C. Schmid, and M. J. Black, “Towards understanding action recognition”, IEEE International Conference on Computer Vision (ICCV), IEEE, pages 3192-3199, Sydney, Australia, December 2013.
- (3) J. Pacheco, **S. Zuffi**, M. J. Black, and E. Sudderth, “Preserving Modes and Messages via Diverse Particle Selection”, In Proceedings of the 31st International Conference on Machine Learning (ICML), J. Machine Learning Research Workshop and Conf. and Proc., volume 32, pages 1152-1160, Beijing, China, June 2014.

CHAPTER 2

Related Work

1. Models of the Human Body

1.1. 2D Body Models. In early work, the knowledge of the articulated structure of the human body produced manually-defined 2D representations based on geometric primitives. An example is the puppet model from Hinton [44] (Figure 1), where a person is assumed to be representable with a set of 15 rectangles. The puppet geometry is defined by simple rules such as “*the length of a part, measured along the proximal-distal axis must be greater than its width. The trunk must be wider than any of the upper limb-parts, and each of these, in turn, must be wider than its connected lower limb-part* [44]”.

In the Cardboard People model of [52] each body part is defined by a polygon with 4 vertexes, and corresponding vertexes of adjacent body parts are connected by springs (Figure 2); the model is applied in a tracking scenario and parts are interactively defined on the first frame of the video sequence.

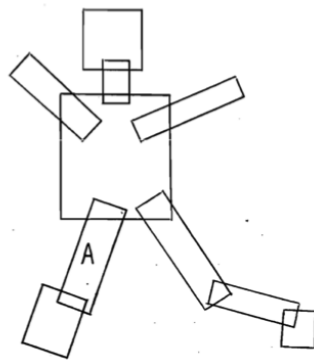


FIGURE 1. The Human Puppet model [44].

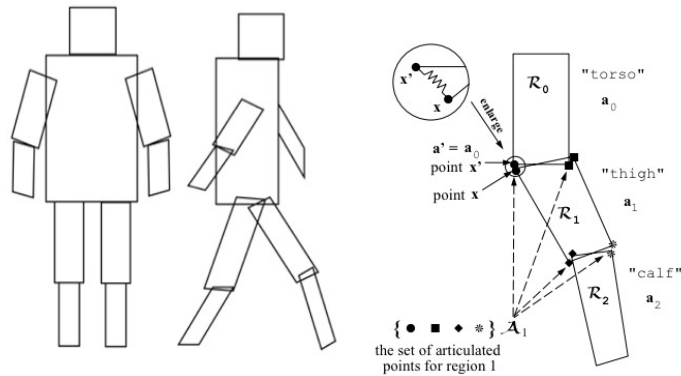


FIGURE 2. The Cardboard People model [52].

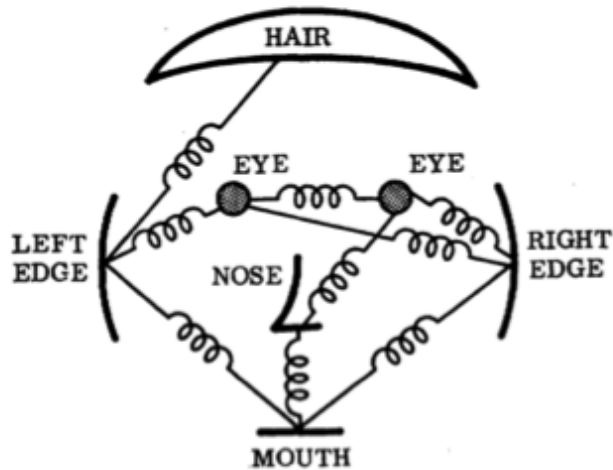


FIGURE 3. Pictorial Structures [32].

Felzenswalb and Huttenlocher revisit the Pictorial Structures (PS) model of Fischler and Elschlager [32] (Figure 3) in the light of efficient inference with dynamic programming. PS defines articulated objects as a collection of rigid body parts connected by springs. The idea of rigid templates connected by single springs traces back to [32]. In their seminal work Felzenswalb and Huttenlocher [30] apply the concept to the human body, with the novelty of learning the size of the parts and the stiffness of the springs from images, and defining the spring model in a probabilistic way; the size of the body parts, represented as rectangles, is fixed. PS is a probabilistic model for human pose with a fixed shape (Figure 4).

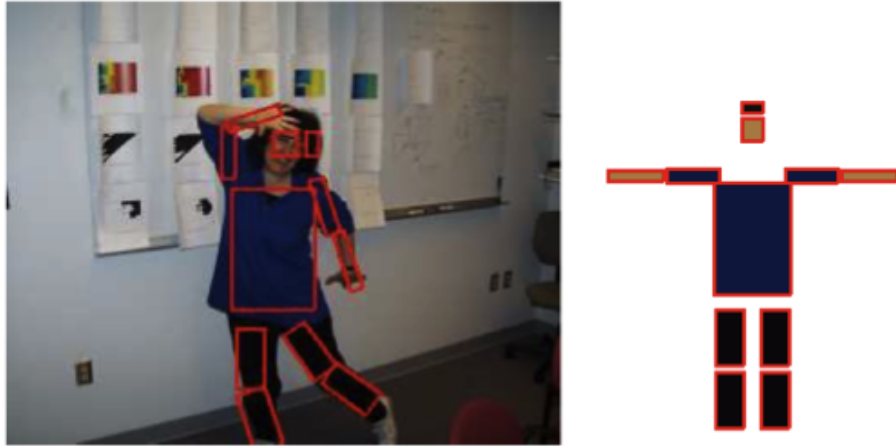


FIGURE 4. Pictorial Structures model of the human body [29].

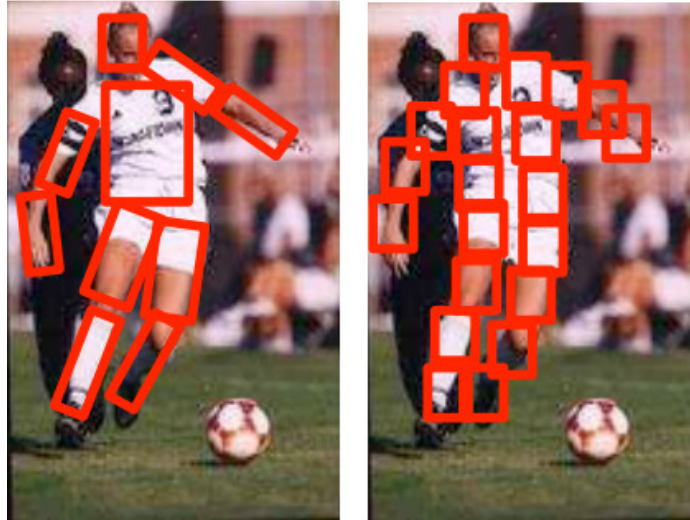


FIGURE 5. Flexible Mixtures of Parts model [107] (right) compared to Pictorial Structures (left).

Yang and Ramanan [107] define a model that resembles PS but has far more parts, that can only translate and do not rotate. Oblique limbs are represented with a set of rectangles suitably translated (Figure 5). This representation is not meant to resemble a human body, but to allow fast pose estimation of people at multiple scales.

Freifeld et al. [34] introduce Contour People (CP), a body model that is parameterized for shape, pose and even camera. Differently from previous 2D models, CP is not part-based, and is learned from projections of the SCAPE model [8]. While not

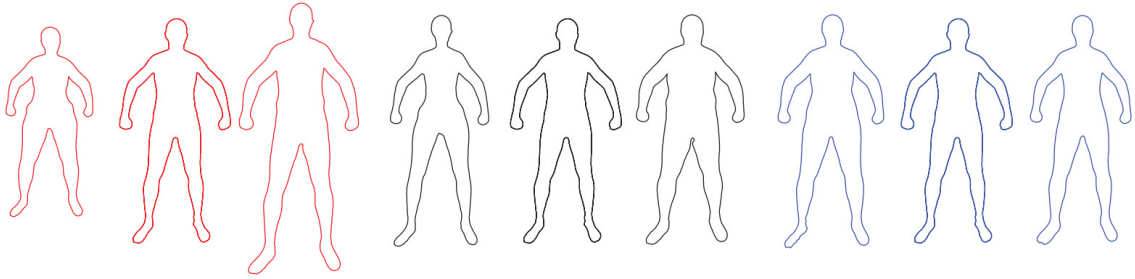


FIGURE 6. Contour Person (CP) model [34]. The first three principal components of a gender-neutral model. For each component, from left to right: -3 , 0 , 3 standard deviations from the mean in the direction of the respective component.

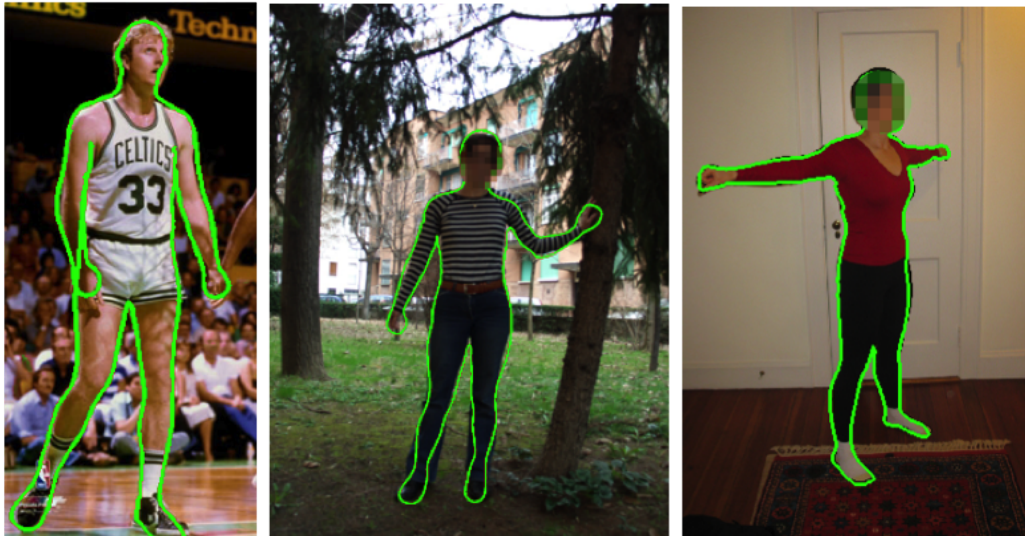


FIGURE 7. Contour Person (CP) model [34]. Examples of the model fit to silhouettes of real people.

being probabilistic, a novelty of the CP model is that it is parameterized for intrinsic body shape (allowing it to generate fat, slim, tall, short people), and for camera. The parameterization is through factored models of contour deformations due to intrinsic shape, camera parameters and pose, where the intrinsic shape model and the camera model are defined by the coefficients of linear models over contour segments, learned with Principal Component Analysis (PCA). Figure 6 shows the first three principal components of a gender-neutral model. Figure 7 shows example of CP fit to real people.

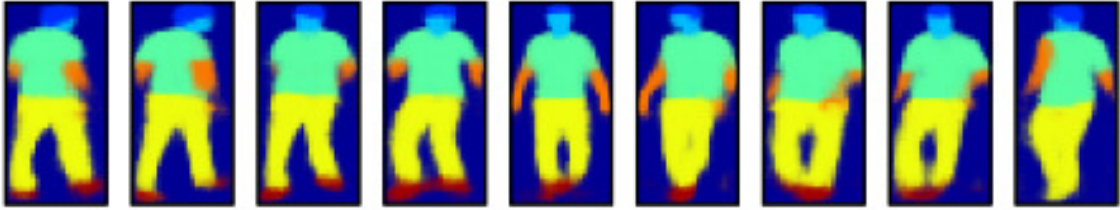


FIGURE 8. Samples from the model of [27].

Eslami et al. [27] introduce a generative shape model of people based on Restricted Boltzmann Machines (RBM). The model has been applied to image segmentation. It can generate pixel-based examples of body parts, but does not provide a parameterization for pose. Figure 8 shows samples from the model.

The models described so far do not address explicit modeling of appearance due to clothing. To this goal, one approach uses compositional models that aim at explicitly describing complex appearances (like a person with or without a hat). They are implemented as AND-OR grammars [110] [19].

Guan et al. [37] provide a contour-based model for dressing the CP model with outer contours that represent different types of clothes (Figure 9); the model does not address the appearance of clothes in terms of color. Instead, a recent work [35] defines a generative model for representing actors in a movie focused on color appearance: it models parts as sets of colored ellipses; it does not include a parametrization for pose.

1.2. 3D Body Models. Even more than 2D models, 3D human body models are tightly associated with the human body skeleton. Given body shape changes are mostly due to articulation, body models have the skeleton-based partitioning into anatomical parts, typically head, torso, upper and lower limbs. Models for hands are usually discarded when working at a full body scale. In the model proposed by Marr and Nishihara [61], instead, the body is represented in a hierarchical way, up to fingers details.

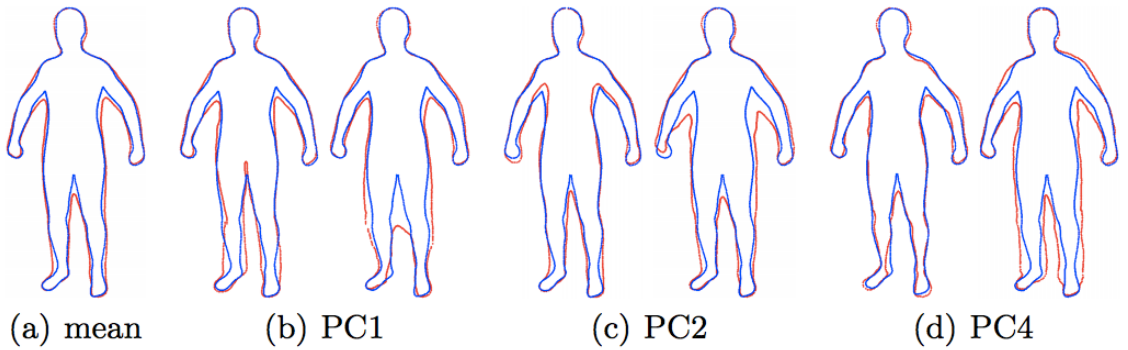


FIGURE 9. Eigen clothing [37]. The blue contour is always the same naked shape. The red contour shows the mean clothing contour (a) and 3 std from the mean for several principal components (b)-(d).

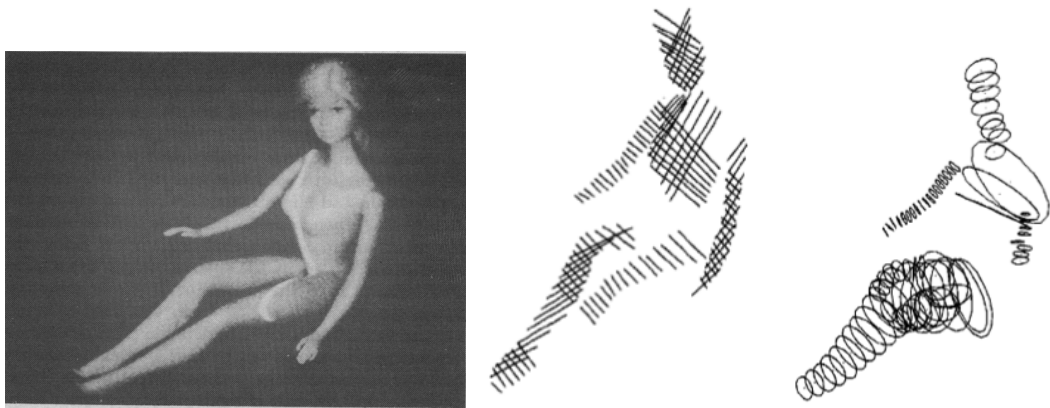


FIGURE 10. The Generalized Cylinder model [1] applied to a Barbie doll.

Agin and Binford define 3D object models using generalized cylinders [1], and apply the method to fit 3D data captured with a laser scan device representing a doll: this can be considered one of the early attempts to align a 3D model to 3D input data of a human body (Figure 10).

Sigal et al. [90] define a loosed-limbed model where parts are loosely connected, and these connections are modeled with Gaussian potentials, then exploited for a non-parametric form of belief propagation. The model is the 3D equivalent of Pictorial Structures, and parts are represented with truncated cones (Figure 12 (left)). The loosed-limbed model was used for tracking, where the size of the body parts was manually defined.

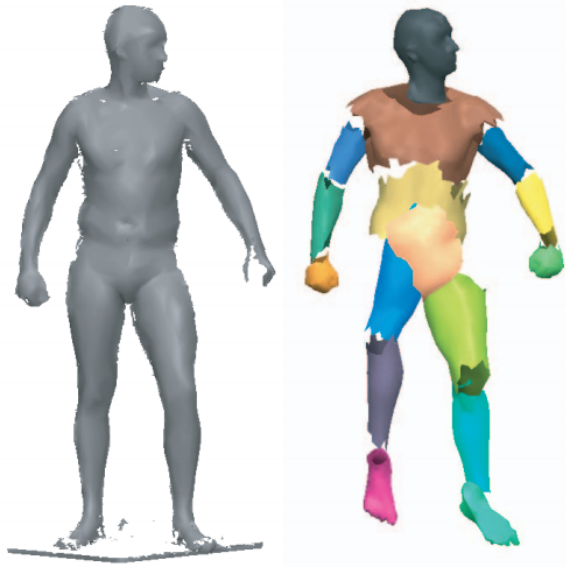


FIGURE 11. Figure reproduced from [81], showing input data (left) and model in estimated pose (right). Note that the 3D model is part-based but parts are rigid and do not connect at joint interfaces.

Rodgers et al. [81] define a part-based 3D model of the human body where body parts have realistic shape, but the model is rigid, and parts do not nicely deform and connect at the joint interfaces (Figure 11). Creating plausible soft deformations for an articulated deformable object with pose is one of the main challenges in designing a body model. In computer graphics, where visual quality is mandatory, building realistic models require a lot of manual effort from animators for rigging and skinning characters, where rigging is the process of creating a skeleton and a set of controls to animate the characters, and skinning is the process that assigns bone influences to the mesh in order to generate visually plausible deformations with pose changes. Such models have an implicit part segmentation induced by the skinning process, but are not part-based. An interesting exception is the work of Miller et al. [62], where they propose defining a database of rigged and skinned parts to be used for automatically assign rigging and skinning to an input mesh by transferring information from the best matching parts.

Sminchisescu et al. [92] use a body model that consists in a skeleton with associated parts that are built with super quadric ellipsoids (Figure 12 (center)).

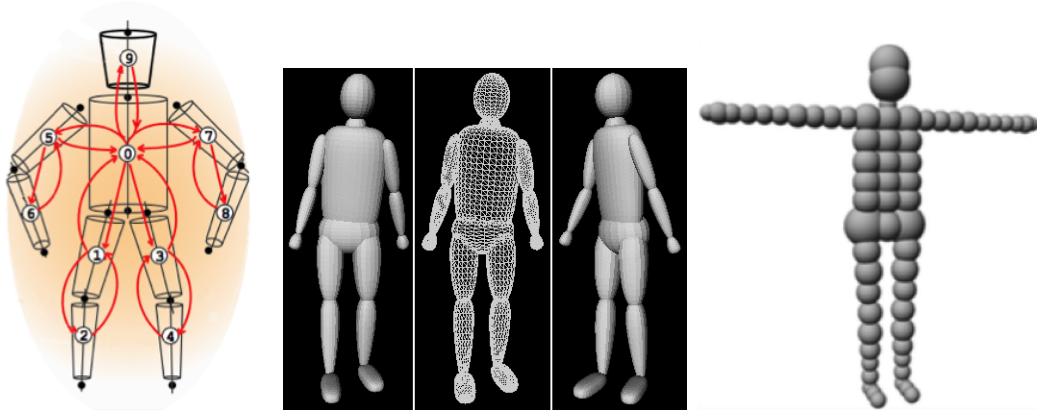


FIGURE 12. (left) The Loosed-limbed model [90]. (center) Superquadrics model [92]. (right) The Sum of Gaussians model [93].

Stoll et al. [93] define a Sum of Gaussians model. Here the body is represented in 3D with a set of spheres of various size (Figure 12 (right)). The model is not part-based, but the simple geometric representation makes it very efficient for inference.

3D Statistical Shape Models. A more recent approach made possible by the availability of 3D scanning devices is the definition of 3D statistical shape models.

Allen et al. [3] analyze the space of the intrinsic shape of the human body. They consider a set of 3D scans of people acquired in approximately the same pose. They obtain a statistical model of the body via Principal Component Analysis (PCA) over shape deformations with respect to a template, for fixed pose.

Anguelov et al. [8] introduce the SCAPE (Shape Completion and Animation of PEople) model, an articulated 3D model where pose-dependent deformations and intrinsic shape are independent factors, and the latter is represented with a low-dimensional model learned with PCA (Figure 13). SCAPE is designed with the goal of faithfully representing the human body, allowing the generation of 3D meshes for different genders, different intrinsic body shape, and with soft deformations of the body to represent the shape variation of muscles and soft tissues due to pose changes. The model has been successfully applied to computer vision tasks [16] [39].

SCAPE is a statistical model learned from two disjoint sets of 3D scans. One set includes a single person in many poses, and the second set includes many people

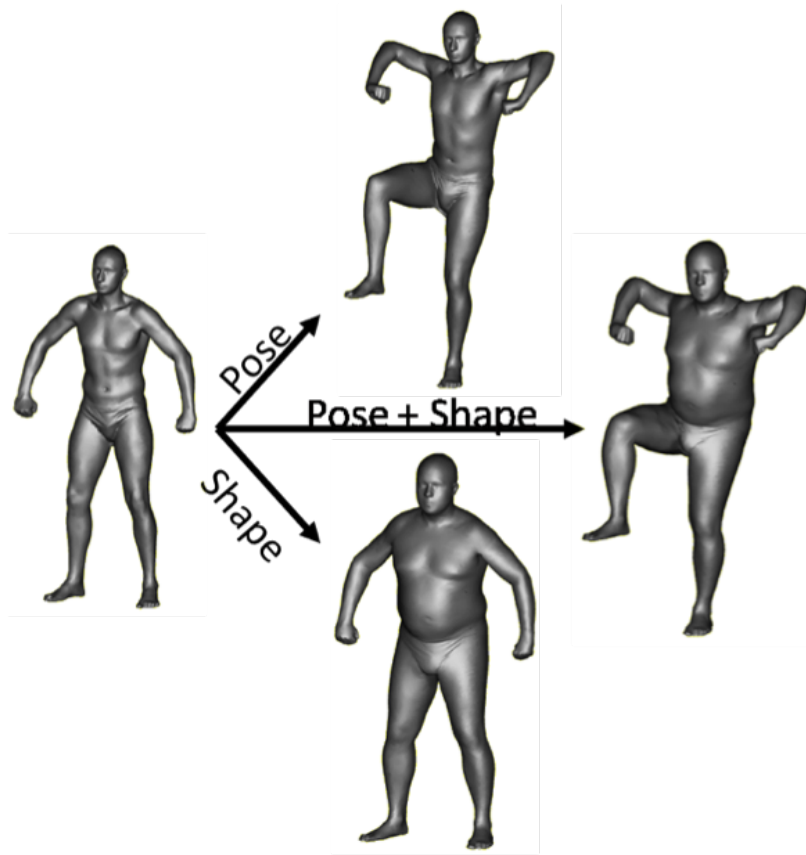


FIGURE 13. The SCAPE model [8].

with different body shape in a template pose. Hasler et al. [42] learn a 3D model similar to SCAPE from a dataset with different people in different poses. Chen et al. [21] introduce the TenBo model, where pose-dependent deformations depend on the intrinsic shape by means of a tensor-based representation.

An illustration of some of the 2D and 3D models presented in this section is presented in Figure 14. Models are divided into those that have a part-based representation vs. global models, and those that have a realistic shape representation vs. simple geometric models. With the models introduced in this thesis, Deformable Structures and Stitched Puppet, we fill two empty spots in the literature, corresponding to part-based models that have a realistic shape.





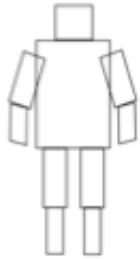



		2D		3D	
part-based		non-realistic	realistic	non-realistic	realistic
		 pictorial structures	 deformable structures	 loose limbed people	 stitched puppet
global		non-realistic	realistic	non-realistic	realistic
		 cardboard people	 contour people	 sum of Gaussians	 SCAPE

FIGURE 14. Articulated models of the human body. The models presented in this thesis are Deformable Structures and Stitched Puppet.

2. 2D Human Pose Estimation from Single Images

Estimating the pose of a person from a single image typically entails providing the location of body joints, in 2D or 3D. Alternatively, outputs are pose parameters of an assumed body model, therefore location and orientation of boxes, sticks or 3D shapes (Figure 15).

In 2D it is very popular to use the prediction of the endpoints of each part to compute the Percentage of Correct Parts (PCP) score [26]. According to PCP, a candidate body part is detected correctly if the endpoints lie within a fraction of the ground-truth part length (usually 0.5). PCP was introduced together with the *Buffy The Vampire Slayer* dataset [31], but the PCP implementation provided with the



FIGURE 15. Human pose estimation. Output can be a set of sticks, body joints, or values for model variables of location and rotation of boxes or 3D shapes.

dataset was less strict than the definition, creating some confusion. In this thesis we compute PCP with the code provided with the Buffy dataset. PCP is not a suitable measure for datasets with large foreshortening and multiple people [108]. In recent works, the joint locations are preferred, and considered a better way to evaluate pose estimation performance.

We illustrate the main approaches to 2D human pose estimation with emphasis on methods based on PS models, as PS-based methods are the most popular and relevant to the work described in this thesis.

2.1. Methods based on Pictorial Structures. 2D or monocular pose estimation has become a very popular topic in computer vision in the last 10 years, since the work of Felzenswalb and Huttenlocher [30], which provides a probabilistic view of a Pictorial Structures (PS) model for human pose estimation. Their generative model is a tree, with each node corresponding to a body part, and variables are location, orientation and scale of the part. The joint probability of the model factors in pairwise distributions learned from images; the image likelihood is defined per part, and efficient MAP inference for the posterior pose is obtained by dynamic programming.

Arguably, the PS model is too crude a representation of people in images, and evaluating image evidence as the degree of matching of the polygon templates with the image only works for simple cases (in their work Felzenswalb and Huttenlocher use Chamfer distance between the model silhouette and a binary mask obtained with background subtraction). For the application of PS models to images in uncontrolled conditions, Andriluka et al. [5] introduce part-based image likelihoods learned from images using discriminative methods. In this case, the rectangular parts do not represent the shape of the body, but are regions within image descriptors are computed.

PS-based methods that use discriminative approaches to evaluate image evidence are today typically formulated as a structured prediction problem [85]. Inspired by the PS model formulation, they model the human body as a collection of loosely connected rigid body parts. Unary and pairwise terms define the part image evidence and the cost of relative distances between parts, respectively. The goal is to estimate the pose of highest score among all the possible configurations of the rigid part templates. How to learn these models is described for example in [55].

After the work of Andriluka et al. [5] that illustrated how generative models can be coupled with discriminative likelihoods, PS-based approaches have encountered a great popularity, and people have intensively worked on trying to improve performance in two ways: adopting more complex image likelihoods and defining multimodal pose models.

Image likelihoods. A characteristic of PS-based models is that the image likelihood is computed for each part independently. It is also in general fast to compute, as it is evaluated for all the model state space. Andriluka et al. [5] use likelihoods modeled with Ada Boost classifiers over shape context descriptors. The most popular approach today is to use HOG (Histogram Of Gradients) [23] features within a log-linear model that is learned with structured prediction [85]. Generic part detectors, learned to detect body parts at any location, orientation, and scale, are typically not very effective. Parikh et al. [68] observe that part detectors are the weak component

in human pose estimation. Yang et al. [107] learn mixture models over clustered-by-pose training data, where it is likely, for example, that they learn different detectors for an upper arm oriented upward, or downward, given a different shading pattern due to the direction of light in natural images. Pischulin et al. [72] also demonstrate that better performance can be obtained with pose-specific appearance models.

Color would be a very powerful feature, as gradient-based features can generate many false positives, especially at small scales. When the problem is to estimate the pose of a person in an image without further information it is hard to define color features. Face and hands have skin color, but one has to assume the person is facing the camera; body models are not accurate enough to identify hand regions. Ramanan et al. [78] first run pose estimation using a generic edge-based likelihood, then use the output to define appearance models for a second run that evaluates edge-based and appearance-based image evidence. Eichner et al. [25] learn location priors for upper body parts in an enlarged upper body person detector window. They support the definition of appearance models for the body parts, given by color histograms with weighted pixel contribution based on the location priors.

Dantone et al. [24] address the problem of defining better part templates using random forest regressors.

Hernandez-Vela et al. [43] exploit Poselets [13] to incorporate mid-level features and obtain more reliable part detectors.

When using more complex likelihoods, it becomes infeasible to compute the likelihood of each part, at each state space location. Sapp et al. [85] propose a cascade of PS models, from coarse-to-fine, where at finer levels more complex likelihoods over a reduced state space are used. Ferrari et al. [31] exploit a person detector to identify foreground regions of interest and estimate the scale of the person in the image.

Pose priors. A limitation in PS models is that the probabilistic model describing relative part locations is too simple: the relative angle between parts is represented in the original formulation of [30] with a von Mises distribution; in [5] is a Gaussian distribution; in reality pose variables have multimodal distributions (for example a

distribution for the lower arm angle can have a mode for the arm straight next to the torso and a mode for the lower arm bent parallel to the waist line), thus modeling training data with a Gaussian model generates unrealistic poses and too wide variance: the model is better used for representing training sets with small pose variations. The Gaussian assumption, that has to hold in a transformed space of relative coordinates, is fundamental for efficient MAP inference, as explained in [30]. Sapp et al. [85] overcome the Gaussian potentials in the transformed space by proposing a more general formulation of the PS model. They suggest a generic log-linear model, with linear pairwise and unary potentials. Potential parameters are learned with structured prediction. With this formulation, efficient inference with a distance transform cannot be performed.

PS-based body models are learned from image annotations that typically do not include viewpoint information (if the person is seen from the front, laterally or from the back). In such generic training sets the distribution of 2D pose is highly multimodal. In order to address the multimodal nature of body pose and part appearance, successful methods consider mixtures models, both for the part templates and for the pairwise interactions [107] [96]. Johnson et al. [51] partition the training data into clusters and learn a different PS model for each of them. Tian et al. [97] use hierarchical latent nodes to implement a tree structured spatial prior, allowing only the non-latent nodes to evaluate image evidence. Sapp et al. [84] introduce MODEC, a multimodal decomposable model, where both unary and pairwise terms are multimodal terms, learned by clustering the body joints in the training set in a normalized image space.

Other approaches define PS models as Conditional Random Fields (CRF) by conditioning the pose model on image data; examples are the work of Sapp et al. [83] and the work of Pischulin et al. [71], where the PS model parameters are retrieved from the output of Poselet [13] detectors run on the image.

Tian et al. [97] build a hierarchical model similar to that proposed by Marr and Nishihara [61]. In their model latent variables define the node *type*, and pairwise terms are distances weighted with type-specific factors.

Chen et al. [20] replace classic parametric pairwise models with conditional probabilities given image patches learned with deep convolutional neural networks. Their model differs from PS as the pairwise relationships between parts are described with a mixture model, and can be one of a set of relationship *types*.

The methods with the best performance today use pose models that are image-dependent.

State space and graph structure. Most of the 2D body models have a tree graph structure with nodes that map to body parts and the root at the torso. Many authors have investigated alternative representations. The part configuration space can include location, rotation and scale or only location and scale. In the former case parts correspond to body limbs, in the latter case more templates are required to model a limb rotation with a set of small templates placed along the rotated limb axis [107] (Figure 5, right). Typically models that consider rotation in the state space work at fixed scale, which is estimated from the image using a person detector. The model of [107] instead, also includes a scale variable, and can estimate simultaneously the pose of many people at different scales.

In a full machine learning approach, models should be blind to prior knowledge of the object structure (i.e. the body skeleton) and one should learn the object partition in rigid parts that best models the training data. Part visibility should be also addressed. Recent work investigates learning the model structure from data [101], but it is more often the case that the model structure is fixed, and so is the number of parts, even for those models that are not anatomically faithful. Figure 16 shows an example of model with 14 body parts and mixtures of four components [107]. Chen et al. [20] use a similar representation but they increase the number of parts, and use 26 parts for a full body and 18 for an upper body model. In their case adding more parts mitigates the risk of overfitting in training a deep convolutional neural network.

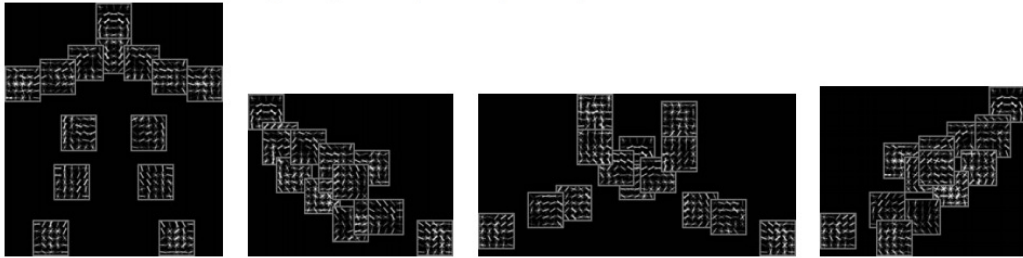


FIGURE 16. Visualization of the discriminative model of [107]. The four models correspond to the best scoring locations for the parts for a model with mixtures of four components. The rectangular patches represent the weights of the HOG template that is used to compute image likelihoods.

Most of the methods assume a tree-structure, but this cannot capture correlations between non connected body parts. Lan et al. [57] augment the tree structure with latent variables to account for coordination between limbs. Interestingly, the work of Wang et al. [101] addresses the following question: “are simple tree models sufficient?”. The answer is yes, but assuming the structure of the tree models is not constrained to be adherent to the human skeleton. Non-tree models can express various dependencies in the body model graph: the similarity relation between the left and the right part of the body in terms of color appearance, occlusion relationships between parts, and relationships between non-connected parts to express likely poses [57].

An alternative representation of the human body in 2D defines a graphical model over joint locations [24] [86]. This definition seems more suitable for datasets with large variation in part length due to foreshortening.

Inference. PS-based methods perform inference with belief propagation (BP). The BP algorithm can solve for the *maximum-a-posteriori* (MAP) solution with Max-Product BP, or for the marginal distributions at each graph node with Sum-Product BP. The Max-Product algorithm is employed to get a consistent (connected) solution; once the messages have been passed through the graph, the solution is given by selecting the state with best max-marginal on the root node, and then traversing the

tree to the leaves to select the node in a backward pass to recover the states that gave the best score. Felzenswalb and Huttenlocher [30] show how to efficiently perform MAP inference exploiting the Distance Transform.

The Sum-Product algorithm computes the marginal distributions at each node. The solution is then given independently for each node. In this case, efficient inference is obtained with a FFT, as messages can be expressed as convolutions.

MAP solutions are in general preferred for the pose estimation problem as they output connected bodies. Andriluka et al. [7] observe that the Sum-Product algorithm is more robust w.r.t. the discretization of the part configuration space.

2.2. Other Methods. While PS-based methods are the most popular approach to human pose estimation, other approaches exist.

In PS-based methods with belief propagation inference, messages are initially passed from the leaves to the root node, implementing a bottom-up processing. This process accumulates evidence going from the leaves to the root to predict where the torso can be. An alternative approach estimates the pose of parts given the estimated pose of the parent. Hara et al. [41] use a manually defined dependency graph that, for each node, makes a prediction for the child state with a regressor over image features computed in a patch centered at the node current state.

Wang et al. [103] exploit Poselets, a novel notion of body parts introduced in [13], which are obtained from clustering in appearance and configuration space. Wang et al. [103] go beyond the rigid body part representation of PS and learn a hierarchy of Poselets that represents parts but also portions of the body.

If an accurate pose is estimated, it should be easier to segment a person on an image. And, if a segmentation is provided, then pose estimation can be more accurate. To be effective, these two tasks should be integrated in a common framework rather than constitute a processing pipeline. Mori et al. [64] exploit segmentation to identify body parts, torso and limbs, that are then put together in a full human body. Wang et al. [102] describe an approach based on dual decomposition. Rothrock et al. [82] incorporate foreground-background segmentation at the part level.

Ladicky et al. [56] estimate the pose of multiple people and their segmentation. In their work they address the problem of occlusion, which has been neglected so far from the majority of the human pose estimation methods. Most of the datasets used for evaluating human pose estimation methods have been defined for all the body parts to be visible. If a part is occluded, it might happen that the method can successfully predict where the part is, but more often errors occur, as training images typically do not include occluded examples. Ladicky et al. can address occlusion between people, not from other parts of the scene.

Recently there has been also interest in pose estimation and segmentation from stereo images. Sheasby et al. [87] extend the formulation of [102] to include disparity images. Vineet et al. [100] propose an efficient approach based on Mean Field to estimate human pose, segmentation and depth. Alahari et al. [2] estimate pose and segmentation of multiple people from 3D movies. They explicitly represent depth ordering and occlusion between people. Gkioxari et al. [36] introduce Armlet classifiers to approach the pose estimation problem in a holistic way. Puwein et al. [76] use branch-and-bound to find pose solutions that are consistent with a foreground mask of the subject. Ramakrishna et al. [77] introduce Pose Machines, a method to estimate joint locations that throws away body models to predict pose with a sequence of multi-class classifiers. Kiefel et al. [53] introduce Fields Of Parts, a new representation of the pose estimation problem, where binary variables are defined for each state to represent the presence of a part at that position, orientation and scale. The advantage of this representation is that it nicely incorporates segmentation and arbitrary graph connections.

Recently, methods based on deep convolutional neural networks for human pose estimation have been proposed [58] [66] [99]. The work of Tompson et al. [99] integrates a convolutional neural network and a graphical model in a hybrid architecture obtaining state-of-the-art results for joint location prediction in the most challenging datasets.

3. 2D Human Pose Estimation from Video Sequences

With pose estimation from video sequences we refer to the combination of monocular pose estimation with tracking in uncontrolled environments. The problem is sometimes referred to as *articulated motion parsing* [86], and entails estimating pose on each frame of a video sequence without manual initialization. Similarly *tracking by detection* assumes independent pose estimates on video frames, and a successive step to enforce temporal consistency.

In a typical tracking application some initial information is provided, for example manual annotation of the pose on the first frame of a sequence. Tracking problems are often formulated as Bayesian filtering problems, where the current state is estimated based on history of observations and previous states. Articulated motion parsing is challenging as, differently from tracking applications, there is no reliable frame to exploit for building for example an appearance model of the subject. Burgos Artizzu et al. [18] merge single-frame pose estimates in coherent estimates across video frames with a novel form of non-maximum suppression for articulated objects. Andriluka et al. [4] [6] model temporal coherence among detections with a hierarchical Gaussian process latent variable model.

Ramanan et al. [79] [80] assume there is at least one frame in the video sequence where the person is in an easy-to-detect pose. Based on this detection, they build a person-specific appearance model and perform independent pose estimation on each frame using the estimated appearance model.

A similar approach is used by Buehler et al. [17] who introduce temporal information by identifying key frames with reliable poses. These effectively become anchor frames, and they exploit temporal coherence to link poses in the intermediate frames. Ferrari et al. [31] formulate a spatio-temporal parsing problem and perform simultaneous pose estimation over a set of frames. In addition to single-frame potentials, the model includes temporal dependencies that represent continuity of appearance and pose. These methods rely on static image likelihoods.

Sapp et al. [86] exploit optical flow information to locate foreground contours. This integrates well with their pose estimation method, which exploits image contours and region-based likelihoods. The idea of using flow discontinuities as a cue for pose estimation dates at least to [92] on 3D body pose estimation in monocular video.

Tokola et al. [98] go beyond the tracking-by-detection approach to include between-frame consistency at the detection stage. They generate a set of tracking hypotheses rather than detection hypotheses, and then a successive stage selects among these paths. The method is thus named tracking-by-selection.

Fragkiadaki et al. [33] exploit optical flow for segmenting body parts and propagate segmentations over time.

Cherian et al. [22] implement a two-stage approach: they use the model from [107] to perform pose estimation on each frame, with the difference that the body graph is expanded to include the wrists and elbows in a successive frame. They use the method of [69] to select a set of diverse solutions. Then, a global energy over the video sequence that favors temporal consistency is optimized over the discrete set of candidates. Often a candidate pose is correct for one arm and wrong for the other arm, thus candidate poses are split into limbs, the optimization is over limb partitions of the body model. An approximate procedure then performs optimization for each body part across the sequence in a top-down fashion, starting at the head. They improve over all previous methods [33] [86] [98] [112] on the VideoPose2.0 dataset [86], and introduce a new dataset, Poses in the Wild.

4. 3D Human Pose and Shape Estimation from 3D Data

The availability of 3D statistical models of the human body allows estimating detailed pose and shape for people of different gender and body type. Given current 3D statistical models focus on bodies in minimal clothing, fitting these models to images so far requires people dressed in tight or minimal clothing. An exception is the DRAPE model [38], which models clothes and their deformation with pose, but it has not been applied to computer vision problems. Balan et al. [15] estimate the

body shape under clothes, but in controlled pose and viewing conditions. Guan et al. estimate the shape and 3D pose of naked people from images, given 2D pose [39]. The methods of [38] and [15] are based on the SCAPE model [8].

Rodgers et al. [81] fit a part-based 3D model of the human body to 3D range scan data for arbitrary pose. The model is rigid and does not allow shape estimation, but interestingly they define the body model in a part-based fashion, formulate the problem in a probabilistic framework, and optimize the resulting Markov network with loopy belief propagation. This approach allows exploiting low-level detectors to suggest part placement hypotheses.

3D statistical models of the human body that can represent people of any shape in any pose also allow taking a model-based approach for the alignment of 3D meshes.

3D Mesh Alignment. The problem of mesh alignment is fundamental to the learning of statistical 3D shape models. Given a set of samples in the form of 3D meshes acquired for instance with a 3D scanner, the goal is to compute correspondences between all the vertexes of all the samples. If the meshes are of a human face, for example, all the vertexes at the tip of the nose have to be in correspondence. After the alignment, performing statistical analysis on the data is possible.

There is a large literature for methods to match 3D shapes. Bronstein et al. [14] align shapes by preserving geodesic distances between corresponding points on the shapes. This method relies on the assumption that the geodesic distance is preserved with pose changes, which is not true when, for example, aligning two human bodies of different intrinsic shape. Zhang et al. [109] propose a method to align articulated shapes of similar objects, for example two different types of dinosaurs. They rely on a computed set of correspondences whose possible matches are scored by a deformation cost. Here the deformations are not learned from real data. The MÖBIUS voting [59] and Blended Intrinsic Maps [54] methods are instead based on conformal mapping, where only angles are preserved.

From the point of view of building statistical models, alignment methods of interest are those that assume the 3D shape is represented with polygons or point clouds;

methods that focus on aligning full shapes and not only portions of them; methods that output dense correspondences; and finally methods that are robust to missing data. Here we review the methods that have been applied to 3D scans of people.

In case of data with significant holes in the mesh, model-based techniques are preferred. The advantage of assuming a model is that if the object is not convex and smooth where data is missing, it is hard to fill holes properly. For example, in the case of the human body, often a big portion of the extremities of the feet is missing. Without prior knowledge about what feet look like, it is not possible to assign values to the missing data in order to obtain a realistic result. Also, when building a statistical model, the object class is often known, so it is feasible to assume a reference model is available to help with the registration.

Methods for mesh alignment can be divided in marker-based methods and marker-free methods. In case of articulated objects like the human body, marker-based methods are usually applied, as often they are the only reliable way of providing a coarse alignment. In marker-based methods the markers can be physically placed on the body at anatomical points easy to detect with palpation before scanning, or manually placed by an operator on the meshes. Both solutions are time-consuming and error prone.

Allen et al. [3] register 3D scans of people from the CEASAR (Civilian American and European Surface Anthropometry Resource Project) dataset (Figure 17). CEASAR includes thousands of range scans of male and female subjects in the age range 18 – 65 collected in the United States and Europe. The subjects were wearing tight clothes, and 74 markers were placed on the body before scanning. The alignment process is formulated as an optimization of an energy function, where the variables to estimate are affine transformations from a template mesh to the data mesh, for each triangle. The energy is composed by a data term that favors a template mesh to match the data, a marker term, that favors the markers to overlap, and a smoothness term, that favors solutions where neighboring triangles undergo the same affine transformation. The registration technique employs markers in a first stage, where



FIGURE 17. Scans from the CEASAR dataset after hole-filling [3].

the data term is disabled, then in a successive stage markers have weaker relevance, as they are not reliable for a precise alignment. Also, the optimization proceeds in a multi-scale fashion, first on low-resolution meshes, to avoid local minima. The energy minimization is robust to holes: if the data point closest to a vertex in the template is on a boundary edge of the data mesh, its data term is disabled, and its affine transformation is defined through the smoothness terms from neighboring triangles. This has the effect that holes in the data mesh are filled by seamlessly transformed triangles of the template surface. The template mesh is obtained by aligning a mesh generated from an artist to one of the CAESAR scans using 58 manually selected landmarks.

Given a set of k meshes aligned to a template with n vertexes, Allen et al. [3] compute Principal Component Analysis (PCA) on the $k \times n$ vertexes of the template transformed according to the estimated alignment transformations for each data mesh. The resulting PCA space describes the variability in the intrinsic shape of people. Note that here the authors assume body pose is fixed, but it is likely that there is some pose variation among subjects, which could have been partially removed by exploiting the markers location. Exploiting the PCA model, data meshes can then be parameterized by PCA coefficients instead of affine transformations. This generates a new form for the energy to optimize for the alignment, where instead of the smoothness term for neighboring affine transformations one would minimize a likelihood for the PCA coefficients, defined as $E_p = \sum_{i=1..K} (p_i/\sigma_i)^2$, where K is the number of coefficients considered, p_i is a PCA coefficient, σ_i^2 is the corresponding variance.

In learning the SCAPE model, Anguelov et al. [8] apply a method based on ICP (Iterative Closest Point) for non-rigid objects [40]. The location of synthetic markers is computed with the Correlated Correspondence (CC) algorithm [9]. To initialize the algorithm, 4 – 10 markers are manually placed by hand on each pair of scans.

Wuhrer et al. [105] introduce a method for automatically assign markers to the meshes to align.

In these works, a template mesh in a canonical pose is used. The template can be designed by an artist or can be a manually refined version of one of the 3D data meshes. In the aligning process, the template mesh is aligned to the data, but in this process it cannot deform, as the goal of the alignment is to provide meshes in correspondence to learn the deformation model. Deformations can occur, and are usually regularized with some geometrical constraint. Regularization can act as a *smoothness* constraint [3] or as a *as rigid as possible* constraint. The latter case is used for near-isometric deformations, and is useful when aligning meshes of the same person, or when manipulating a mesh of a character.

A better computation of mesh correspondences would be possible with a template that can deform according to a learned model, but this would constitute a chicken-and-egg problem, as to learn the deformation model, the meshes have to be in correspondence. Hirshberg et al. [45] simultaneously perform alignment over a large corpus of scans and learn the deformation model. They also address the problem of aligning scans of people of different shapes in different poses.

Deformable Structure Model

The Deformable Structures Model (DS) is a part-based articulated model of human body in 2D. We call instances of the model *DS puppets*, to underline the part-based representation, as the body appears as composed by flat pieces that are connected at the joints (Fig. 1) and can rotate around them. Differently from a typical rigid puppet, in DS parts change their shape when their relative rotation angle changes.

With DS, we want to capture the correlation between the shape of body parts and their relative pose. For example, if a person has an arm straight, the upper and lower arm have a specific shape, with elongated muscles. When the person bends the arm, the limbs shape change, as muscles shorten and the elbow joint becomes visible. In DS we want to correlate relative pose and shape through a probabilistic model learned from data.

1. Definition

In the DS model each body part is represented by 3 types of variables: location, orientation, and shape. Let $\mathbf{l}_i = (\mathbf{c}_i, \theta_i, \mathbf{z}_i)$ be the variables of the part i . Then, \mathbf{c}_i is

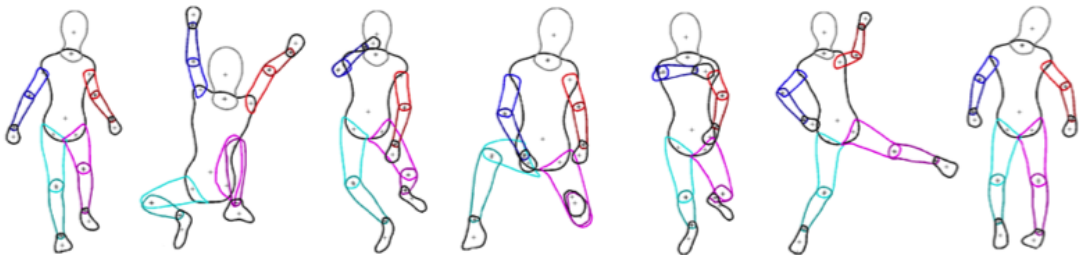


FIGURE 1. Examples of female DS models in different poses.

the 2D location of the center of the part; θ_i is the angle of rotation of the part around its center; and \mathbf{z}_i is a set of *shape coefficients*.

DS is a probabilistic model over pose and shape. In a graphical model representation, it constitutes a tree-structured graph, with the torso part as the root, and a node for each body part. The edges of the graph map to anatomical joint connections, and parts relationships are therefore pairwise.

Let Θ denote the parameters of the model, and \mathbf{l} the configuration of all its parts. The DS model is defined as:

$$(5) \quad p(\mathbf{l}|\Theta) = \frac{1}{Z(\Theta)} \prod_{(i,j) \in E} p_{ij}(\mathbf{l}_i, \mathbf{l}_j | \Theta_{ij}),$$

where E is the set of edges, Z is the partition function, and $p_{ij}(\mathbf{l}_i, \mathbf{l}_j | \Theta_{ij})$ are pairwise parametric distributions with parameters Θ_{ij} .

2. Learning

With the DS model we aim at representing the shape of people in images. Ultimately, one application of the model is for 2D human pose estimation, therefore our goal is in principle to model the appearance of people, in terms of shape, on generic images. Following the same approach of the PS model, which is learned from images, would require manual annotations on many images to learn a model able to capture the variability of human appearance. It would be hard to generate such data: while for pose annotation a user would just click on joints locations, for annotating shape she would have to segment the person from the background, and also define a partitioning of the foreground silhouette into parts, and this would be an ambiguous task. For example, if a person is wearing a skirt, it is not clear how a part-based model would account for it. It is more appropriate to focus on modeling the actual shape of the naked human body, where the segmentation into parts clearly reflects its articulated nature. Once we have defined the body model, we can design a likelihood that is robust to appearance changes. With this goal in mind, we have the advantage of being able to use computer graphics to generate our training data, without the need

of laborious manual annotations on images. In fact, as illustrated in the previous chapters, models exist to easily generate examples of the minimally-clothed human body in 3D.

The DS model is learned from training contours derived from SCAPE [8], a parametric 3D model of articulated human shape. SCAPE can generate 3D meshes of human bodies for different genders and intrinsic shape, where with intrinsic shape we refer to the body shape that is related to height, weight and in general to the body type of a person. Also, the model can generate realistic shape deformations of muscles and fat with pose changes. In the DS model our focus is modeling pose-dependent shape deformations, and therefore we consider a fixed intrinsic shape for generating training data. The DS model is gender-specific, and we learn two different models generating training data for female and male genders.

Given a gender and an intrinsic body shape, we generate training contours of body parts for different poses. Using an approach similar to [34] we generate random SCAPE poses and random cameras, and project the random SCAPE models on the image plane to create 2D training contours. The relative pose between random cameras and SCAPE models is approximatively frontal-view. The training contours are, for each body part, closed contours. This is because we exploit the mesh segmentation of SCAPE to project each part independently and compute the contour from the obtained part silhouette. Figure 2 shows two examples of SCAPE models and corresponding projection with parts represented as closed contours.

Training samples are generated, for the male and female genders, with fixed intrinsic shape. An ideal choice would be to learn models for the mean intrinsic shape in the training data. In the SCAPE model we use, the average male and female have quite big bodies, as they represent average people in the US. Therefore, for the male model we picked the average intrinsic shape, as we perform experiments on images from a TV series where the male characters have strong bodies. For the female model, instead, we picked a thin body, as the average female in SCAPE is too big compared with the female characters in the datasets we use for our experiments.

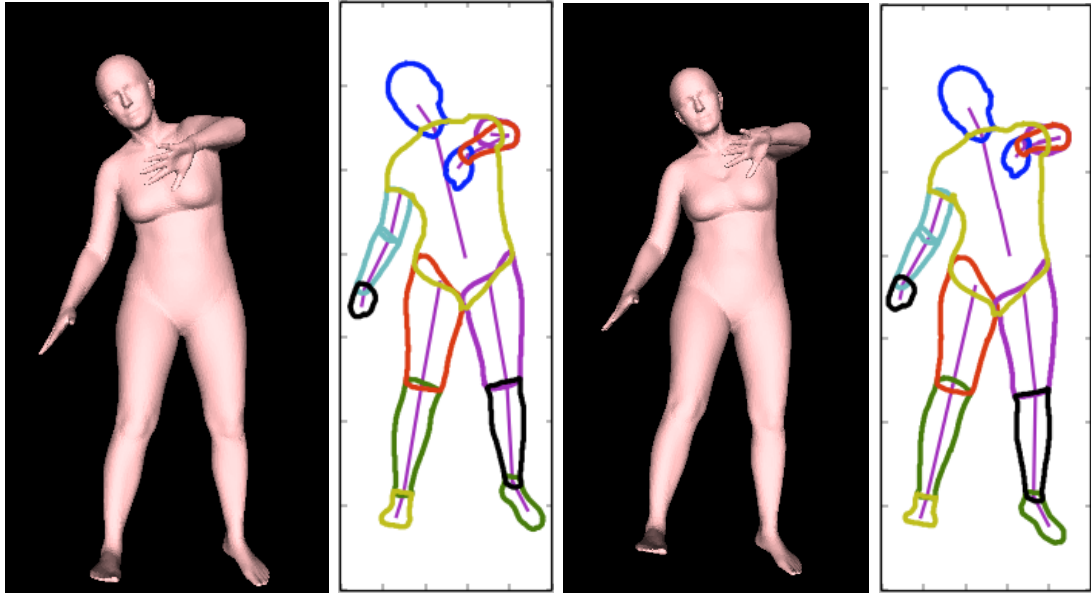


FIGURE 2. Contours generated from the SCAPE model. The rendered 3D meshes are SCAPE models in two similar poses; the contour-based representations are obtained by per-part projections of the meshes on the image plane, and then extracting the contour from silhouettes.

Each DS model is learned from 3000 mirrored samples. Figure 3 shows example poses in the training set for the female body; note the variability of pose and orientation of the body relative to the camera. While the camera is frontal-view, the random noise we add generates also slightly lateral poses (see Figure 3).

Each part is rendered as a separate 2D closed contour and discretized into a fixed number of contour points plus two additional “joint” locations at the proximal and distal ends of the part. The two “joints” define a local coordinate system for the part (Figure 2) and a line through them divides the part into two sides. Each side of the part is sampled to a fixed number of points, evenly spaced according to the arclength. We represent therefore each part with a fixed number of points.

We learn models with various numbers of body parts: 10 parts, consisting of the head, torso, upper and lower limbs, where the hands and feet are included in the lower limbs; and 14 parts with hands and feet treated as independent parts. We use

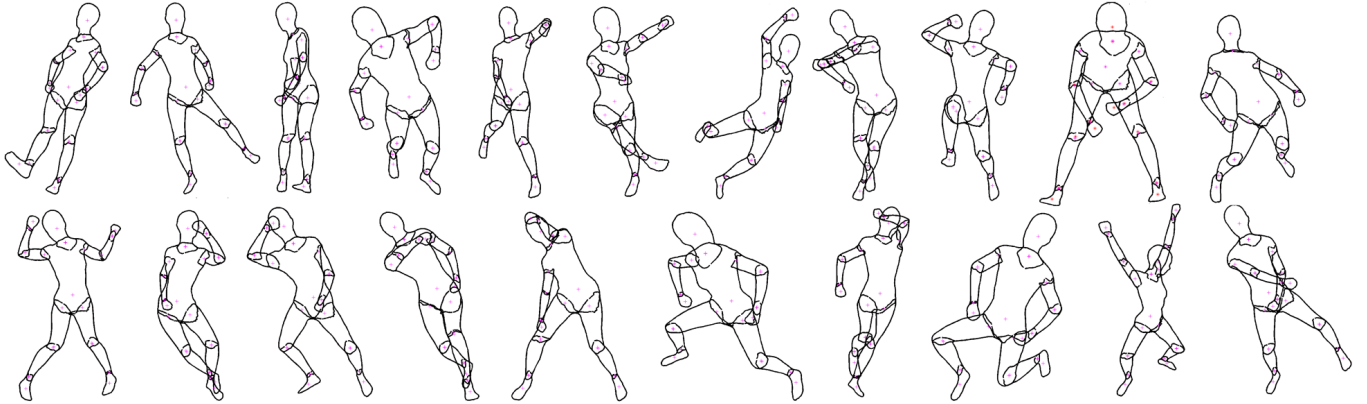


FIGURE 3. Examples of training poses. Note the variability in pose as well as in camera location.

the 10-part model for our pose estimation experiments, for a more direct comparison to traditional PS models.

3. Shape Representation

The shape of each part is learned independently and then these local shape models are coupled in the graphical model by the pairwise potentials. The training examples for each part are aligned to a common coordinate system, which has one axis corresponding to the bone of the part, as defined by the joint points. The mid-point of the bone is the part center; the bone defines the part length and rotation. We vectorize the set of contour points \mathbf{p}_i and joint points \mathbf{y}_i , all in the local frame of reference, and form training vectors composed of the contour and joint points. We put these training vectors into a large matrix of training data, and perform Principal Component Analysis (PCA). We can then represent a training vector with a low-dimensional linear model as:

$$(6) \quad \begin{bmatrix} \mathbf{p}_i \\ \mathbf{y}_i \end{bmatrix} = \mathbf{B}_i \mathbf{z}_i + \mathbf{m}_i,$$

where \mathbf{p}_i is a vector of contour points and \mathbf{y}_i is a vector of joint points. The vector \mathbf{m}_i represents the mean contour (and joints) of part i . \mathbf{B}_i is a matrix containing the eigenvectors of the training data corresponding to the dominant eigenvalues. Finally,

\mathbf{z}_i is a vector of linear shape coefficients that are used to represent different part shapes.

Figure 4 shows the mean contour and joint points for the head, upper leg and torso, together with contours and joint points at 2 standard deviations from the mean, for the first three PCA basis components. The first component of the PCA representation typically corresponds approximately to foreshortening along the major axis.

Most of the joint points correspond to actual anatomical joints. For a limb, we call *proximal* the joint that is closer to the torso, and *distal* the other joint. The segment connecting proximal and distal joints is the part bone. The extremities (head, hands and feet) have one distal point that is not an anatomical center of rotation, but is defined at the average of the part points. The torso has 6 joint points: shoulders, hips, neck and belly button. While not an anatomical joint, the belly button is the distal point for the torso; with the neck join it defines the torso bone, and is thus used to compute the torso orientation and length.

Equation 6 represents points in local frame. In order to draw a part on an image, we first compute contour points in global coordinates as:

$$(7) \quad \tilde{\mathbf{p}}_{i,k} = R(\theta_i) \mathbf{p}_{i,k} + T(\mathbf{c}_i),$$

where $R(\theta_i)$ is a rotation matrix, $T(\mathbf{c}_i)$ is a translation matrix, and k indicates the k -th point in the vector \mathbf{p}_i . In order to draw a part on an image, we need to convert the DS coordinates, which refer to an arbitrary global frame with normalized coordinates, to image coordinates. We compute the following:

$$(8) \quad \hat{\mathbf{p}}_{i,k,x} = N_x (s_{DS} \tilde{\mathbf{p}}_{i,k,x} + 0.5)$$

$$(9) \quad \hat{\mathbf{p}}_{i,k,y} = N_y (s_{DS} \tilde{\mathbf{p}}_{i,k,y} + 0.5),$$

where N_x and N_y are width and height of the image in pixels, and s_{DS} is the DS scale parameter.

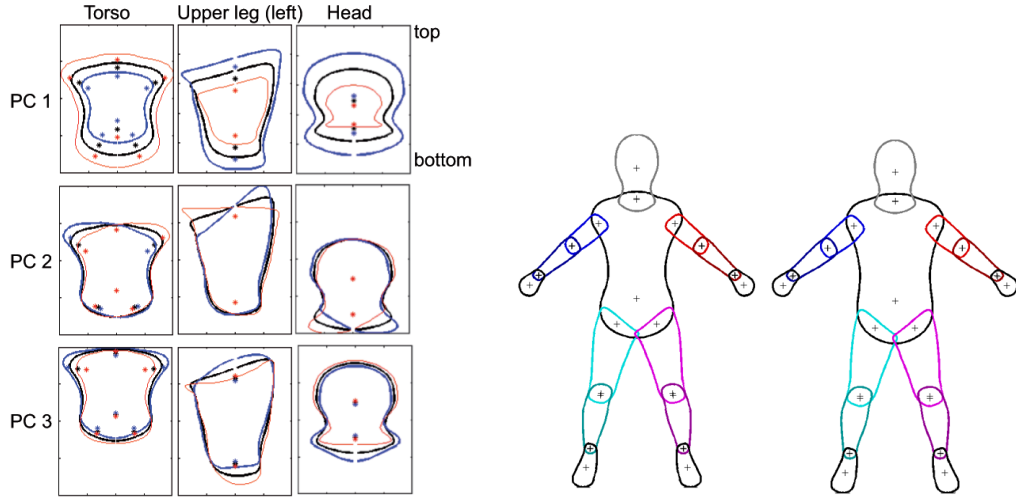


FIGURE 4. DS part deformations. (left) Deformations for three example parts. Black is the mean contour. Red and blue are ± 2 standard deviations from the mean along the first 3 principal component directions. Stars mark the joint locations that deform with the contour. (right) Mean part shapes for the female and male body (14-part model). The dots represent joint points (see text).

4. Pairwise Relationship of Connected Parts

In the DS model the relationship between model variables of connected parts is modeled with pairwise distributions that relate the shape coefficients of a part to the shape and relative orientation of neighboring parts. While these relationships could be quite complex, we find that a reasonable representation is obtained with a simple Gaussian model. Let i and j be two connected parts. The pairwise model between part i and part j is a multivariate Gaussian

$$(10) \quad p_{ij}(\mathbf{I}_i, \mathbf{I}_j | \Theta_{ij}) = \mathcal{N}(\mathbf{z}_j, \sin(\theta_{ji}), \cos(\theta_{ji}), \mathbf{q}_{ji}, t_j, \mathbf{z}_i, t_i; \Theta_{ij})$$

where θ_{ji} is the relative angle of j with respect to i . The vector \mathbf{q}_{ji} defines the distance between the joints of the parts; that is, $\mathbf{q}_{ji} = (\mathbf{y}_{ji} - \mathbf{y}_{ij})$, where \mathbf{y}_{ji} is the joint point of part j connecting j with part i and \mathbf{y}_{ij} is the joint point of part i . The points \mathbf{y}_{ji} and \mathbf{y}_{ij} are both defined in the local coordinate system of the part i , which has its origin \mathbf{c}_i at the midpoint between the joint points, and is aligned with the main axis

of the part. Note that the vector \mathbf{q}_{ji} is analogous to the *spring* that connects two parts in the PS model representation. The scalars t_i and t_j are the length of the two parts, defined as the distance between the part joints. Finally, $\Theta_{ij} = (\mu_{ij}, \Sigma_{ij})$ is the mean and covariance of the Gaussian model, which are computed from the training samples.

5. Generating Model Instances

In order to sample the DS model, we proceed as follows. Starting at the torso, which is the root of the model, we generate a random shape by sampling the Gaussian distribution over the PCA shape coefficients of the part:

$$(11) \quad \mathbf{z}_i \sim \mathcal{N}(\mathbf{z}_i; \mu_{\mathbf{z}_i}, \Sigma_{\mathbf{z}_i}),$$

where the mean vector is a zero-valued vector and the covariance matrix is diagonal. Note that we could start from any other part considered as the root. We then assign global orientation θ_i and location \mathbf{c}_i . This allows generating the torso contour and joint points in a global frame with Eq. 7.

We generate the part j given i exploiting the Gaussian form of the DS model. We can in fact condition the pairwise distribution $p_{ij}(\mathbf{l}_i, \mathbf{l}_j | \Theta_{ij})$ with the shape variables of the part i . This gives a conditional Gaussian distribution over the variables of part j :

$$(12) \quad p_{j|i}(\mathbf{l}_j | \mathbf{l}_i, \Theta_{j|i}) = p_{j|i}(\mathbf{z}_j, \sin(\theta_{ji}), \cos(\theta_{ji}), \mathbf{q}_{ji} | \mathbf{z}_i, \Theta_{j|i}) = \mathcal{N}(\mathbf{z}_j, \sin(\theta_{ji}), \cos(\theta_{ji}); \mu_{j|i}, \Sigma_{j|i}),$$

where we have marginalized the parts lengths t_i and t_j and the joints distance \mathbf{q}_{ji} . The parameters of the conditional distribution are obtained as follows (see [11], sec. 2.3):

$$(13) \quad \mu_{j|i} = \mu_a + \Sigma_{ab} \Sigma_{bb}^{-1} (x_b - \mu_b)$$

$$(14) \quad \Sigma_{j|i} = \Sigma_{aa} - \Sigma_{ab} \Sigma_{bb}^{-1} \Sigma_{ba},$$

where we have indicated with the subscript a the indexes of the conditioned variables in μ_{ij} , with the subscript b the indexes of the conditioning variables, and the marginal covariance matrices derive from writing:

$$\Sigma_{ij} = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$$

From the conditional distribution (Eq. 12) we can sample the variables $\mathbf{z}_j, \sin(\theta_{ji}), \cos(\theta_{ji}), \mathbf{q}_{ji}$. We compute $\theta_j = \theta_i + \arctan(\sin(\theta_{ji}), \cos(\theta_{ji}))$ and $\mathbf{c}_j = R(\theta_j) (-\mathbf{y}_{ji}) + T(\mathbf{y}_{ij} + \mathbf{q}_{ji})$, where \mathbf{y}_{ij} is the joint point on the part i that is connected with the part j , and is computed with Eq. 6; similarly \mathbf{y}_{ji} is the joint point on the part j that is connected with the part i , and is computed with the same equation for part j .

Figure 13 shows examples for 4 torso samples. The colored parts correspond to the mean of the conditional models. Dotted lines represent samples we obtained by first sampling the conditional distributions for the upper arms and legs, and then generating the corresponding lower arms and legs, respectively. Note that, based on the shape of the torso, we obtain different conditional distributions for the relative angle between torso and upper arms. For example, in the cases Figure 5 top row, the torso has the shoulders oriented downward, and the upper arms are correctly oriented also downward. In the cases in the bottom row instead, the shape of the torso sets the upper arms oriented at approximately 90 degrees with respect to the torso.

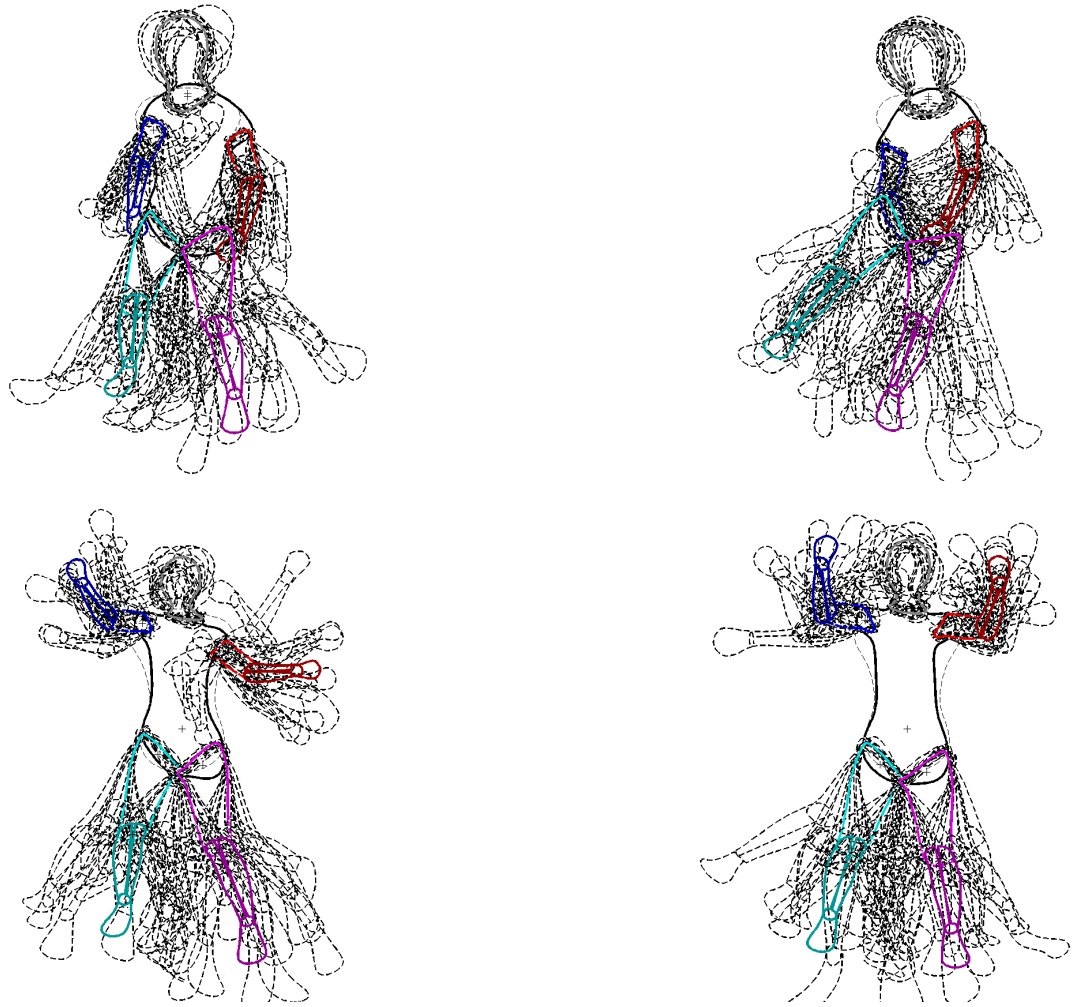


FIGURE 5. Samples from a DS model with 10 shape variables for each part. The colored parts correspond to the mean of the conditional models we obtain given the torso shape. Dotted lines represent samples obtained by sampling the conditional distributions for the upper arms and legs, and then generating the corresponding lower arms and legs, respectively.

Human Pose Estimation from Single Images with the Deformable Structures Model

We apply the Deformable Structures (DS) model to the task of pose estimation from single images. The goal is to estimate the model’s parameters of pose and shape deformations associated with the best alignment of the 2D model to the person in the image. Here we consider a dataset specifically built for the task of upper body pose estimation with the Pictorial Structures (PS) model, the *Buffy the Vampire Slayer* dataset [31]. The dataset includes frames of the Buffy TV show where characters are seen from a mostly frontal view, all the arms are visible, and given the interest in the upper body it is most often the case that the legs are outside the frame. Also, there is an assumption that only a single dominating person is in the frame. Each frame has been manually annotated for one subject, and in case of two or more people in the frame, a selection is done by the annotator. This may not correspond to the subject selected from the pose estimation method, creating ambiguities in the evaluation [108]. Some authors have therefore defined a processed version of the Buffy dataset, where a person is selected, and the frame has been cropped and scaled to include only the chosen subject [85]. More recently a version of the Buffy dataset with original frames and annotations for all the subjects has been introduced [56].

The Buffy dataset is of particular interest for testing our method based on DS, as it has been used by many authors and the characters have a frontal view. The DS model is in fact viewpoint-specific, and while it tolerates some rotations with respect to the frontal view, in case of lateral poses a different DS model, learned from training data with the appropriate camera location, should be used. We have learned DS models from different views, and these have been used to annotate a dataset of actions [49]. However, an alternative option could be to learn a 3D version of the

DS model, where there is no necessity to handle a set of DS models for a number of discrete viewpoints. We will introduce this model in a later chapter of this thesis.

1. Inference

We estimate the human body pose and shape on an image by maximizing the following posterior distribution:

$$(15) \quad p(\mathbf{l}|I, \Theta) \propto \prod_i p_i(I|\mathbf{l}_i) \prod_{(i,j) \in E} p_{ij}(\mathbf{l}_i, \mathbf{l}_j|\Theta_{ij}).$$

Like PS models, the factored form of DS allows doing inference using belief propagation (BP). Unfortunately, efficient BP algorithms assume a discrete (or discretized) space. When the state space of a variable cannot be enumerated, and the potentials do not allow the computation of messages in closed form, sampling approaches can be used [46, 47, 94]. Such non-parametric methods have been applied successfully in human pose estimation in 2D [89] and 3D [90]. We adopt a method inspired by particle belief propagation [46].

In the DS graphical model, the node variables are continuous. Given the large number of variables (we use 4 shape parameters per part), we formulate our inference problem as one of selecting the best configuration among discrete sets of part samples, or particles. This entails first initializing samples at each node, then iteratively performing particle resampling to improve the location of the samples, and using Max-Product BP to derive the most likely configuration. At each iteration, the BP message from node i to node j , defined over the discrete set of N samples, takes the form:

$$(16) \quad \hat{m}_{ij}(\mathbf{l}_j^{(q)}) = \max_{s=1..N} p_{ij}(\mathbf{l}_i^{(s)}, \mathbf{l}_j^{(q)}) p_i(I|\mathbf{l}_i^{(s)}) \prod_{u \in \Gamma(i) \setminus j} \hat{m}_{ui}(\mathbf{l}_i^{(s)})$$

where \mathbf{l}_i is the set of node variables for the part i , $\Gamma(i)$ is the set of neighbours of the part i , s and q are particle indexes at node i and j , respectively. Here we assume all the nodes have the same number of N particles. The product of all the incoming

messages to a node defines the node max-marginal:

$$(17) \quad b_i(\mathbf{I}_i^{(q)}) = \prod_{u \in \Gamma(i)} \hat{m}_{ui}(\mathbf{I}_i^{(q)}).$$

At each iteration, the global best particle configuration can be computed by selecting first the particle that maximizes the max-marginal on the root node and then selecting particles on each node by back-propagation. The Max-Product assignment can also be given as $\mathbf{I}_i^{(\hat{q})}$, where $\hat{q} = \arg \max_{q=1..N} b_i(\mathbf{I}_i^{(q)})$. This latter technique assumes the maximizer is unique [104].

Our simple formulation of the particle-based Max-Product BP as discrete Max-Product BP over a set of particles derives from the following. We have considered the message formulation in the Sum-Product particle belief propagation method (PBP) of Ihler et al. [46]:

$$(18) \quad \hat{m}_{ts}^{(i)} = \frac{1}{n} \sum_{j=1..n} \Phi_{ts}(x_t^{(j)}, x_s^{(i)}) \frac{\Phi_t(x_t^{(j)})}{W_t(x_t^{(j)})} \prod_{u \in \Gamma(t) \setminus s} \hat{m}_{ut}^{(j)}$$

which approximates a message with a set of samples $x_t^{(1)}, \dots, x_t^{(n)}$ drawn from a function W_t . We have replaced the summation with a maximization operation and removed the sampling weights $W_t(x_t^{(j)})$ obtaining:

$$(19) \quad \hat{m}_{ts}^{(i)} = \max_{j=1..n} \Phi_{ts}(x_t^{(j)}, x_s^{(i)}) \Phi_t(x_t^{(j)}) \prod_{u \in \Gamma(t) \setminus s} \hat{m}_{ut}^{(j)},$$

which corresponds to Equation 16. Note that the simplicity of the formulation comes at some risk. The particles represent points in a large dimensional space, and if this space is not covered in an appropriate way, it might be impossible to find a good solution. For example, suppose that in the set of particles of the lower arm there is a particle that represents a very good hypothesis for the lower arm. And a very good hypothesis is also on the torso node, for a torso. If the upper arm node does not contain a good hypothesis to *support* the good ones at the neighboring nodes, both of them can be associated with weak messages, and may be discarded in a successive particles resampling operation. It is important therefore that good hypotheses provide proposals for new parts in neighbouring nodes.

Since the DS model is learned from SCAPE, the joint positions of neighboring parts overlap exactly. This means the learned model variance in the springs is zero, which makes inference difficult, as any configuration with parts that do not connect precisely at the joint would have zero probability. Consequently, for inference, we artificially inflate the variance for the springs connecting parts to give non-zero probability to configurations of parts that are not connected. This creates a *loosely-connected* DS model.

Initialization. To provide a good initialization for the part location and orientation, we run a standard PS inference method [5] as a pre-processing stage. Running a simple PS model to prune the search space is a common strategy [85, 102]. We initialize part samples in three ways: 1) Starting with the PS solution, take the part locations and orientations and draw a shape at random from a prior over part shapes; 2) Draw a random body from our DS prior, 3) Draw parts independently from a prior over locations, orientations, and shapes. These independent part priors are learned from an annotated set of the Buffy training images.

In order to generate a DS instance from the output of a PS-based method, we need to estimate a mapping between the body model of the PS-based method of choice and the DS model. Training sets for PS models are annotated with *stickmen*, which means the annotator draws a segment along the main axis of each body part, to approximately match the human skeleton. We need to estimate regression functions between the ground truth annotations (*stickmen*) and DS, in order to map a solution from a PS method given as stickman to a DS instance. We need therefore to annotate the stickmen training images with the DS model to have stickmen-DS correspondences. We obtain DS annotations using an image annotation tool described below. Then, we learn a regression function to estimate the scale of the DS model and the location of the DS shoulders with respect to the stickmen annotations. For elbows and knees joints the stickmen annotations correspond roughly to DS. In the case of the shoulders joints, the DS model has them located more at the armpit: we need a mapping in

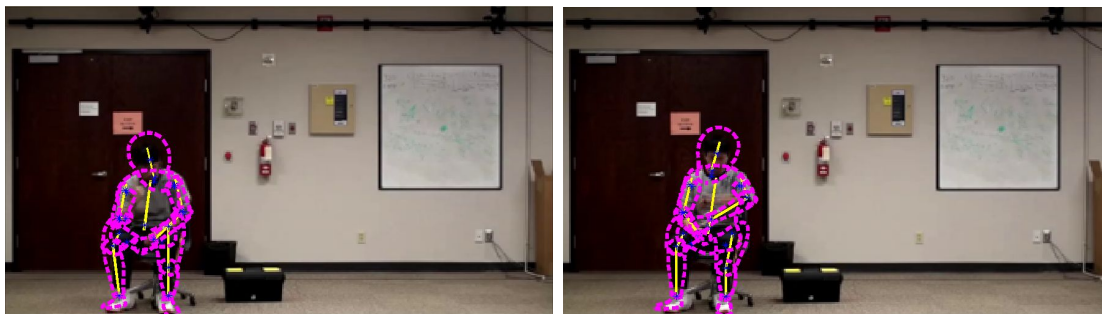


FIGURE 1. PS to DS. Examples of estimates from a PS model (yellow) [107] mapped to DS (magenta).

order to generate proper DS models and also to evaluate our results without bias. To estimate the scale, we build a regression function between the sum of the head size and the distances between neck and shoulder joints and the ground truth DS scale from the annotations. We have in fact observed that the shoulder locations are the most reliable information, while the torso length, which could provide information about the torso scale, is a more error-prone variable. Of course, if a person is seen from a side view the shoulder width is not a reliable parameter. Here we are using a frontal DS model; for dealing with cases where the person is seen from a side a different function should be used.

Figure 1 shows examples of PS estimates (yellow) and corresponding DS estimates (magenta) computed with the stickmen-to-DS mapping, on arbitrary images. The mapping from the PS solution and DS has been learned from the Buffy training set.

We have published two papers, [111] and [67], where we perform pose estimation with DS. For the experiments in [111] we obtain the scale estimate by running a pose detector method [31], which provides the scale estimate for all the particles. For the experiments in [67] we assign the scale by running the method of [107], as in that case we assume the presence of multiple people in the image, and [107] outputs multiple solutions at different scales. In [67] we also augment the node variables with a scale random variable.

After particles have been initialized at the nodes, a first iteration of discrete Max-Product BP estimates the location of the torso. We then build an appearance color

model as a histogram in CIE a^*b^* space [28] from the pixels that correspond to the head and upper part of the torso. This is our appearance model for the upper arms, which we assume can have the same color of the upper torso or can be skin colored like the face. We then perform a number of iterations of discrete Max-Product BP where at each iteration the discrete samples (particles) are resampled in order to improve the solution.

Resampling. During each subsequent iteration of BP, new samples are generated at each node (part) in three ways: 1) by a random walk from the current samples 2) proposed by the neighbors 3) with bottom-up proposals.

The proposal functions to get proposals from neighbors are obtained from the DS model: given a neighbor node, we generate a conditional pairwise distribution for the current node with the same technique explained in the previous chapter for generating samples from DS. Sampling this distribution gives proposals for the current node variables, given a chosen particle on the neighbor. We also consider the state of more than one neighbor for the upper arms: given a likely torso and a likely lower arm, a new upper arm is sampled conditioned on the parent shape and the child location. The conditional proposal in fact exploits the part length parameter in the model formulation (10) to generate a sample that is likely to connect the torso and the lower arm. The bottom-up proposals are derived from the skin color map, from which we sample likely locations for the wrists, assuming the hands are skin colored.

For each new sample, we evaluate its acceptance probability as in PBP [46], with the difference that good samples are never removed from the node. Here, in fact, we are performing MAP estimate, and we are interested in the best solution, while in [46] the goal is to use Monte Carlo methods for modeling marginal distributions of the node variables.

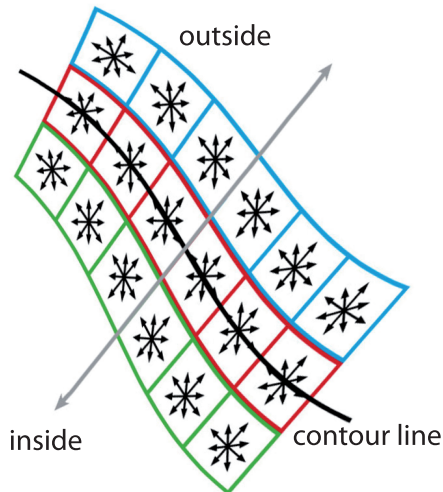


FIGURE 2. Contour likelihood. The image shows the location of the HOG cells along a limb contour. Cells are located on the boundary, just inside, and just outside the part.

2. Likelihood

The likelihood function, $p_i(\mathbf{l}_i)$, represents the probability of the image data assuming a part in a specific location in the image. Since DS defines contour points for each body part, we are able to focus the likelihood computation on the part boundary. Additionally, since we know inside from outside, it is straightforward to formulate likelihood models of skin color or textural appearance. We define the likelihood function as:

$$(20) \quad p_i(I|\mathbf{l}_i) = \phi_i^{\text{contour}}(\mathbf{l}_i)\phi_i^{\text{color}}(\mathbf{l}_i).$$

The contour-based term is given by

$$(21) \quad \phi_i^{\text{contour}}(\mathbf{l}_i) = \frac{1}{1 + \exp(a_i f_i(h_i(\mathbf{l}_i)) + b_i)}$$

where $f_i(h_i(\mathbf{l}_i))$ is the output of a linear SVM classifier applied to the feature vector $h_i(\mathbf{l}_i)$, and a_i and b_i are scalar calibration parameters [73].

The feature vector consists of a set of HOG descriptors computed along the part contour (cf. [60]). For a part i , we take a set of points at fixed locations in the contour

vector $\tilde{\mathbf{p}}_i$, where we compute three HOG descriptors: one at the contour point, one inside and one outside the part contour (Figure 2). All the gradients recorded by the descriptors are steered according to the local contour orientation [88] (Figure 3). We compute the HOG descriptors on a subset of the contour points obtained by subsampling the contour vectors; we used 120 descriptors for the torso, 60 for the head, 40 for the upper arms, 60 for the lower arms including hands, and 80 for the upper and lower legs. We learn the likelihood from the Buffy training set that we have manually annotated with upper-body DS puppets. We also mirrored all the training samples to double the training data. In [111] the HOG descriptors are computed with an adaptive cell dimension, which is set on the basis of the size of the DS puppet. The dimension of the HOG cell is defined with the following equation:

$$(22) \quad c = c_{\text{canonical}} s_{DS} N_y,$$

where $c_{\text{canonical}}$ is what we call the canonical cell dimension and is set at 0.02, s_{DS} is the scale of the DS model, and N_y is the vertical size of the image.

The canonical cell dimension is defined on the training set. We evaluated, for different values of the canonical cell dimension, the score of a linear SVM classifier for positive examples and negative examples. We plot the margin of the classifier versus a quantity expressing contour distance (we considered average distance between contour points and parts overlap score), and selected the canonical cell distance for which ground truth samples score more than negative samples (Figure 4). In [67] and [112] we use a multi-scale approach, and compute the HOG descriptors in an image pyramid with 3 levels and a 0.5 scale factor between one level and the next to the top in the pyramid (Figure 5).

The color likelihood assumes that the lower arms are likely to be skin colored and the upper arms are likely to have the same colors as the upper torso. The color-based probability of a limb is then defined as the mean of the color probability of its pixels. The probability of a pixel being skin is represented by a histogram of skin



FIGURE 3. HOG descriptors are steered to the contour orientation. (a) DS annotation. (b) HOG visualization.

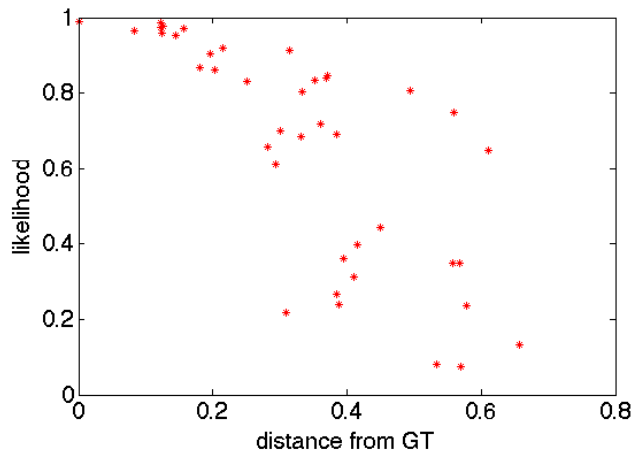


FIGURE 4. Likelihood examples. (a) Random examples for the torso. (b) Plot of the likelihood value for the random samples vs. distance computed as overlap score from the ground-truth annotation; note that here the ground-truth point (for which distance is zero) has the highest likelihood. (c) Sample with highest likelihood.



FIGURE 5. Contour-based likelihood features. HOG descriptors are computed at contour points (red), inside (green) and outside the contour (blue) in a 3-level pyramid.

colors computed from a publicly available data set¹ and from the head regions of our training set. The probability of a pixel having the color of the upper torso is image specific, and is computed using the histogram of the pixels covered by the upper region of the torso and the head once an initial torso estimation has been provided by the inference algorithm.

3. Image Annotation Tool

To generate training data for the contour-based likelihood we use an image annotation tool that allows us to overlap the DS puppet to an image. The annotation tool is written in Javascript, and is executed within a Web browser. The user uploads an image and a DS model. The DS model appears as a transparent layer over the image (Figure 6). Joint points are shown with large red dots, and the user can drag them on the image to pose the puppet.

In the annotation tool the user supplies information about the desired state of the DS model by moving the joint points. This generates a set of constraints for the model variables of pose and part length. Given this information, we estimate shape

¹<http://acouchis.helmholtz-muenchen.de/staff/giovani/colour/>

variables for each part. In an initial version of the annotation tool the user could specify only a subset of the constraints, leaving some of the joints unspecified. Then the model was forced to match the joints the user had specified, and for the remaining joints to provide the most likely pose and shape. We found that this design of the annotation tool was not appreciated by users, as they preferred a simple interface where there is no “suggestion” from the model.

Consider again the DS model representation as a tree-structured graph. We can run inference with message passing techniques in order to estimate the shape variables at each part given joint locations. We convert the joint locations into pose and part length values, then we estimate the shape variables for a DS model given pose and part lengths exploiting the Gaussian form of the pairwise relationships. Consider the message from the node i to the node j :

$$(23) \quad m_{ij}(\mathbf{z}_j) = \int_{\mathbf{z}_i} p_{ij}(\mathbf{z}_i, \mathbf{z}_j, \sin(\theta_{ji}), \cos(\theta_{ji}), t_j) p_i(\mathbf{y}_i | \mathbf{z}_i) \prod_{u \in \Gamma(i) \setminus j} m_{ui}(\mathbf{z}_i) d\mathbf{z}_i,$$

where the relative angle θ_{ji} and the part length t_j are computed given the location of the puppet joints. The likelihood is a Gaussian distribution $p_i(\mathbf{y}_i | \mathbf{z}_i) = \mathcal{N}(\mathbf{z}_i; \mu_{\mathbf{z}_i}, \Sigma_{\mathbf{z}_i})$, where \mathbf{y}_i are the joints of the part i in local coordinates. The mean of the likelihood is $\mu_{\mathbf{z}_i} = (\mathbf{B}_{\mathbf{j}_i}^T \mathbf{B}_{\mathbf{j}_i} + \lambda I)^{-1} \mathbf{B}_{\mathbf{j}_i}^T (\mathbf{y}_i - \mathbf{m}_{\mathbf{j}_i})$, where $\mathbf{B}_{\mathbf{j}_i}$ is the matrix of the PCA components for the joint points, and $\mathbf{m}_{\mathbf{j}_i}$ is the mean of the joint points in the PCA model. The role of this likelihood function is to give high probability to the value of the shape variables that generate the joint points at the position that has been specified by the user. The covariance matrix $\Sigma_{\mathbf{z}_i}$ is set for simplicity to the same value for all parts, and equal the covariance of the PCA shape coefficients of the torso. An exception is made for the torso, for which we set a smaller covariance $\Sigma_{\mathbf{z}_i} = 0.001 I$. This gives more importance to the likelihood term for the torso, which has the effect of keeping the torso joints in place. With a higher variance in fact, the DS prior dominates and the torso tends to assume the mean shape, even if the user wants something different.

In the annotation tool the likelihood is a Gaussian with mean corresponding to the location of the joints specified by the user. We interpret this likelihood as a delta of Dirac function at the mean value of the Gaussian likelihood, as what we want is that the location of the joints is kept fixed. Therefore, we can approximate the message in Eq. 23 with a conditional distribution $p_{j|i}(\mathbf{z}_j|\mathbf{z}_i, \sin(\theta_{ji}), \cos(\theta_{ji}), t_j) = \mathcal{N}(\mathbf{z}_j|\mathbf{z}_i, \sin(\theta_{ji}), \cos(\theta_{ji}), t_j; \tilde{\boldsymbol{\mu}}_{ij}, \tilde{\boldsymbol{\Sigma}}_{ij})$, where the parameters $\tilde{\boldsymbol{\mu}}_{ij}$ and $\tilde{\boldsymbol{\Sigma}}_{ij}$ are computed as described in the previous chapter.

Given the Gaussian form of the messages, message multiplication is performed by summing the precision matrices and the inverse mean vectors of the incoming messages from the neighbors. Let the message from part u to part i be:

$$(24) \quad m_{ui}(\mathbf{z}_i) = \mathcal{N}(\mathbf{z}_i; \tilde{\boldsymbol{\mu}}_{ui}, \tilde{\boldsymbol{\Sigma}}_{ui}).$$

The product of the incoming messages to node i is also a Gaussian:

$$(25) \quad b_i(\mathbf{z}_i) \propto \prod_{u \in \Gamma(i)} m_{ui}(\mathbf{z}_i) = \mathcal{N}(\mathbf{z}_i; \tilde{\boldsymbol{\mu}}_i, \tilde{\boldsymbol{\Sigma}}_i),$$

where

$$(26) \quad \tilde{\boldsymbol{\Sigma}}_i = \tilde{\boldsymbol{\Lambda}}_i^{-1} = \left(\sum_{u \in \Gamma(i)} \tilde{\boldsymbol{\Lambda}}_{ui} \right)^{-1}$$

$$(27) \quad \tilde{\boldsymbol{\mu}}_i = \tilde{\boldsymbol{\Sigma}}_i \tilde{\boldsymbol{\nu}}_i$$

$$(28) \quad \tilde{\boldsymbol{\Lambda}}_{ui} = \tilde{\boldsymbol{\Sigma}}_{ui}^{-1}$$

$$(29) \quad \tilde{\boldsymbol{\nu}}_i = \sum_{u \in \Gamma(i)} \tilde{\boldsymbol{\nu}}_{ui}$$

$$(30) \quad \tilde{\boldsymbol{\nu}}_{ui} = \tilde{\boldsymbol{\Sigma}}_{ui} \tilde{\boldsymbol{\mu}}_{ui}.$$

To estimate the shape variables at each node given pose and part lengths we first compute all the messages in the graph with Equation 23. Then, we compute the belief of each graph with Equation 25, and the estimated value of the shape variables for node i is $\tilde{\boldsymbol{\mu}}_i$.



FIGURE 6. Examples from the DS annotation tool.

4. Experiments

Our hypothesis is that a more accurate model of body shape results in a more discriminative likelihood model and consequently more precise estimation of body pose in images. To test this hypothesis, we perform several experiments, published in [111], and compare with state of the art at the time of the publication. Experiments are run on the Buffy images used in [5]; we use a test set of 235 images for consistency with Sapp et al. [85]. The independent part priors (over location and shape) that are used for initialization are learned from annotated Buffy images not present in the test set.

We run the iterative inference method for 6 iterations and 80 particles per part. Given the stochastic nature of the inference, we run the method 3 times with different random seeds, and select the solution with highest posterior probability.

Error is computed using the percentage of correctly estimated body parts (PCP) used in previous work. A body part is correctly estimated if its end points lie within half of the ground truth segment’s length from the ground truth end points [25]. The likelihood training set consists of DS puppets from the annotation tool and corresponding images, none of which are from the Buffy series.

We test 3 different versions of the DS model: 1) The full model but with a *uniform likelihood* function (NL). This tests “chance” performance that, in this case, is quite high since the inference uses fairly good proposal functions. If the proposal functions were perfect, the problem would already be solved. 2) A model with no shape variation (NS). This uses the *mean shape* of the DS puppet and tests the value of the shape model. 3) The full DS model (DS), which has 4 shape coefficients per part.

Since our inference relies on a PS model for initialization, we take this to be the baseline. A more discriminative likelihood should improve performance. A fully fair comparison is not possible. It is not possible to simply interchange the three key components: inference, model (prior), and likelihood. They interact in ways that make separate analysis difficult. The PS model, for example, uses a discretized state space and optimal inference. Our method is at a disadvantage in that the inference is stochastic.

Performance results are reported in Table 1. In our inference framework the DS model effectively refines the results of the baseline method. Additionally the value of the shape space is seen in the significant improvement over the mean shape model that is only slightly more accurate than the PS model.

Figure 7 shows several representative examples where the DS model (solid red) improved over the baseline PS method (dashed green). The performance of the inference is dependent on the performance of the baseline PS model in providing a good initialization, and on the correctness of the scale estimation. The method, however, does recover from some fairly bad failures of the scale estimation. Figure 8 shows some representative examples of failures.

Method	Torso	Head	U. Arms	L. Arms	Total
Baseline (PS)	97.0	92.3	86.3	52.1	77.7
Our (NL)	99.2	97.9	90.9	10.4	66.6
Our (NS)	99.2	97.5	94.0	50.4	80.9
Our (DS)	99.6	99.2	94.7	62.8	85.6
Eichner et. al.	98.7	97.9	82.8	59.8	80.1
CPS	100	96.2	95.3	63.0	85.5
Y&R	100	99.6	96.6	70.9	89.1

TABLE 1. Results (PCP) from [111] for DS model without likelihood (NL), DS model with a fixed shape (NS), and full DS model (DS), with shape variation. For comparison with other methods we report results from [85] (CPS) and Yang and Ramanan (Y&R) [107]. The results reported for Eichner et. al. are taken from [85]. Baseline (PS) is [5].

In more recent work [67], we have applied an improved version, named D-PMP, of the Max-Product particle belief propagation algorithm previously described, also exploiting a more recent and accurate method for the initial estimate of the pose [107] and treating the scale of the DS model as a random variable at each node. Moreover we have employed a multi-scale version of our HOG-based contour likelihood. A detailed description of the D-PMP algorithm can be found in [67]. We also have used a more recent version of the Buffy dataset, with *stickmen* annotations for all figures [56]. We have partitioned images into single- and multi-person groups, and results are reported on each set, separately in Table 2. The values in Table 2 are PCP scores with threshold 0.5 for lower arms and the average over all parts. In case of the experiment for multiple people we report also a detection rate that corresponds to the proportion of people for which the head or torso is correctly detected. Examples of results are shown in Figure 9. Note that the puppets in Figure 9 have a more variable shape with respect to the ones in Figure 7 as for the latest results we have used 5 shape PCA coefficients per part instead of 4.



FIGURE 7. Estimated body pose from [111]— examples where the DS model improves on the PS baseline. DS is solid red, PS is dashed green.

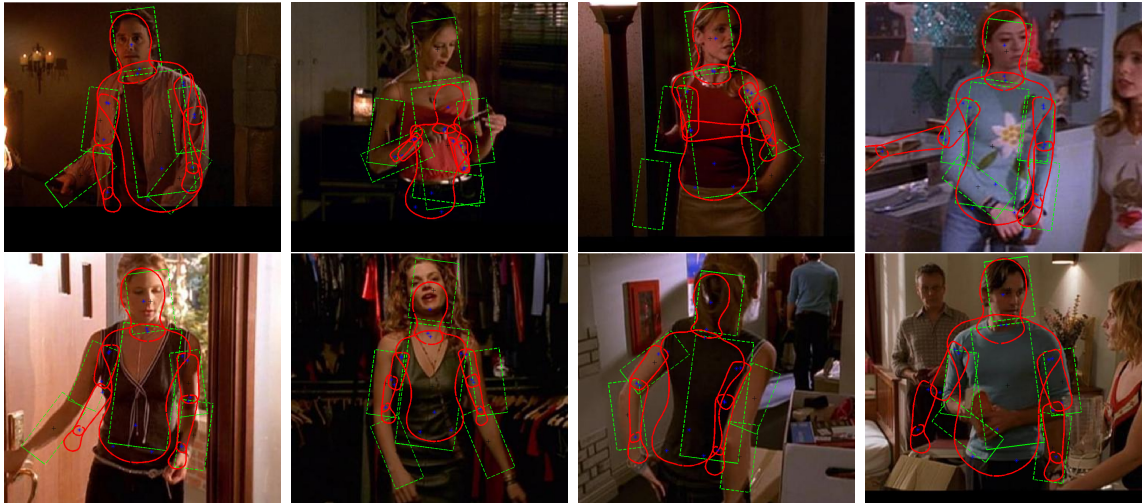


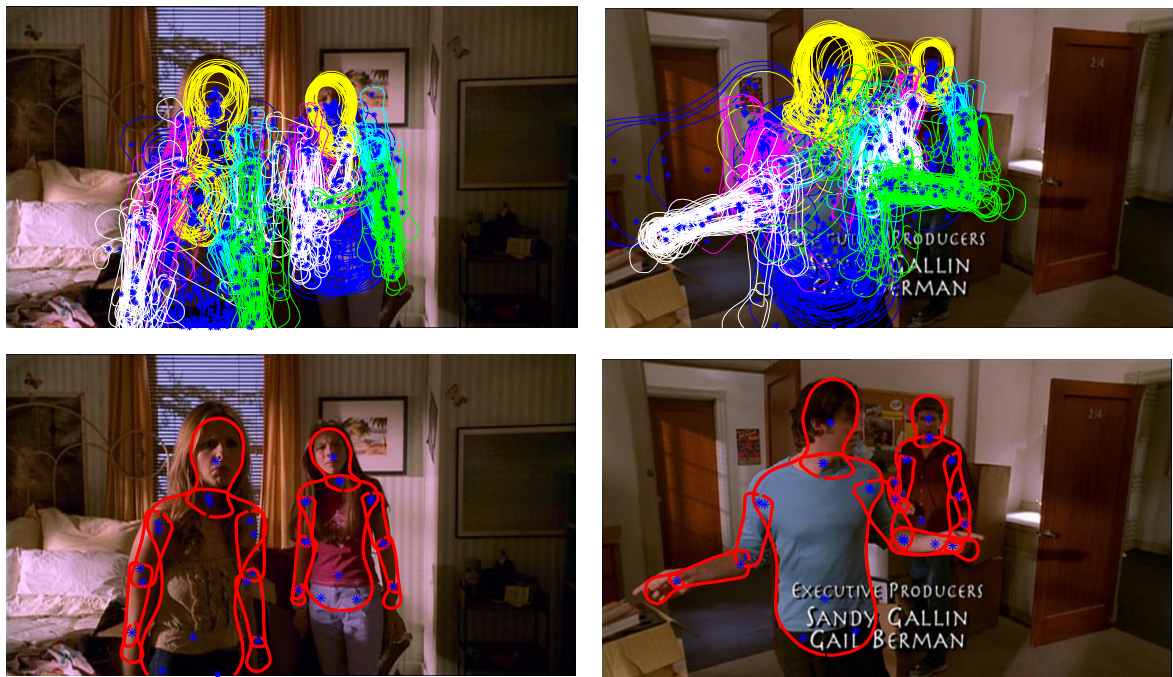
FIGURE 8. Estimated body pose from [111]– representative failure cases. DS is solid red, PS is dashed green.

Method	Single Person			Multiple People			
	L.R. Arm	L.L. Arm	Total	L.R. Arm	L.L. Arm	Total	Det. Rate
Y&R	0.61	0.69	0.87	0.65	0.61	0.83	0.72
D-PMP	0.67	0.77	0.89	0.66	0.68	0.85	0.72

TABLE 2. Results from [67]. PCP scores for lower arms and average PCP with threshold 0.5 on single and multi-person images. Detection rate for the multi-person partition is the proportion of people for which the head or torso is correctly detected.



Single Person



Multiple People

FIGURE 9. Results from [67] on Buffy Dataset. (top) Single person images showing a MAP estimate (red) with different solutions for the arm. (bottom) The full set of particles at the final iteration of D-PMP (top). Best pose in the set of retained hypotheses (bottom, red).

Pose Estimation on Video Sequences with the Deformable Structures Model

In this chapter we address the problem of estimating the 2D pose of a person in monocular video sequences captured under uncontrolled conditions, without manual initialization.

We consider the case of a video sequence with a person moving in front of the camera and we use the DS model to represent the foreground. In our approach we exploit visual cues from the video frames and also the optical flow between frames, that we consider here an observation.

1. Exploiting Optical Flow

Optical flow is the motion of the pixels between adjacent video frames. Computing the optical flow between two frames is hard, as often what it is useful is not the motion of pixels but the motion of the objects in the scene, and this may not correspond to pixel motions due to changes in illumination, occlusion, or shape changes of the objects. In recent years, many advances have been made for computing optical flow in the presence of fast motion, and today computed optical flow can be considered as an useful source of information for higher level tasks. In fact, if we look at the optical flow for a set of frames of a video sequence containing a moving person, we can clearly see that useful information for pose estimation is present (Figure 1).

Single frame pose estimation is challenging and current methods tend to do poorly at estimating the pose of the limbs. Arms and hands are relatively small and can be difficult to localize due to occlusion, motion blur, accidental alignment, ambiguities, poor contrast, etc. Incorporating information from multiple frames may alleviate some of these problems. Previous approaches for human pose estimation in video



FIGURE 1. Optical Flow. The second row shows the forward optical flow of the sequence of frames in the first row. In the optical flow images the hue corresponds to the direction as indicated in the image on the right. The saturation corresponds to flow magnitude.

sequences implement a *tracking by detection* approach: they use image evidence in individual frames to estimate pose and then try to infer a coherent sequence of poses by imposing priors that encode smooth motion over time [4] [6] [22]. A common problem of tracking-by-detection is that images with weak pose cues, for which pose estimation returned a wrong estimate, can deteriorate the whole solution. Stronger cues would come from better appearance models, but this is feasible only for *tracking* where an initial pose is given. A more general problem formulation entails building graphs (usually trees) representing the pose in each frame and connecting corresponding nodes on neighboring frames. The resulting graph has many loops and can only be solved with approximate methods (i.e. loopy belief propagation), which can be slow for long sequences. We observed that learning and applying motion priors for articulated object like the human body is not easy.

Like previous work we want to exploit the inherent consistency of appearance and motion over time. However, we take a novel approach that does not rely on human motion priors. Instead, we exploit optical flow in three ways: 1) to exploit image evidence from adjacent frames, 2) to propagate information over time, and 3) to provide richer cues for pose estimation.

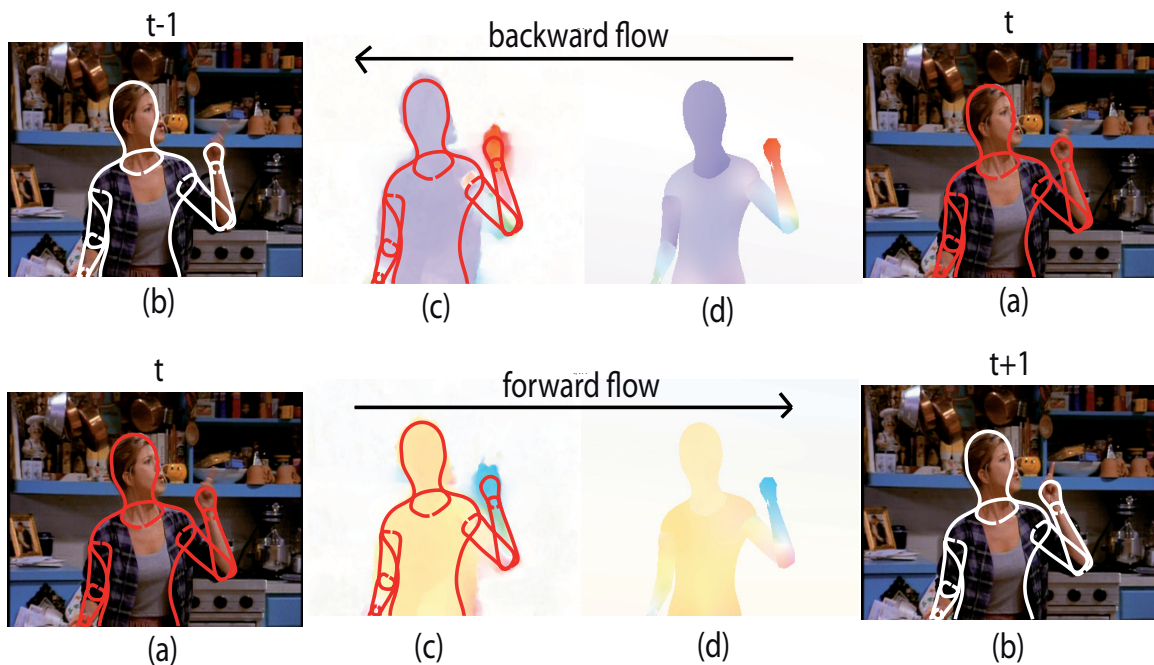


FIGURE 2. Flowing puppets. (a) Frame with a hypothesized human “puppet” model; (b) The prediction of the puppet from (a) into the adjacent frame using the puppet flow; (c) Dense flow between frame (a) and its neighboring frames; (d) The flow is approximated by an affine background layer and the motion of a foreground puppet (the puppet flow).

Our approach is enabled by recent advances in methods for dense optical flow computation. The availability of accurate estimates of dense optical flow allows us to consider the flow as an observation, while the articulated DS model provides a tool for modeling the regions of motion of a moving person. The question is: How can optical flow be incorporated to make the pose inference problem simpler and more accurate?

Consider the problem of estimating body pose in a set of frames for which we also have reliable flow information. Assume that we have a hypothesis for the body at frame t (Figure 2(a), red). In any given frame, the image evidence may be ambiguous and we would like to combine evidence from multiple frames to more robustly infer pose. Due to the complexity of human pose, we perform inference using a distribution



FIGURE 3. Puppet flow. (left) Frame with overlapped a DS model; (center) dense optical flow; (right) puppet flow.

of “particles” at each frame, where each particle represents the pose of the body. If we are lucky and have particles at frame t and $t + 1$ that are both correct, then the poses in each frame explain the image evidence and the change in pose between frames is consistent with the flow. This should increase our confidence in the solution.

Unfortunately such an approach is not practical. Estimating the pose of the body in two frames simultaneously effectively doubles the size of the state space that, for articulated body models, is already high. Alternatively, if we independently estimate the pose in both frames then, given the high-dimensional space and a small set of particles, we will have to be extremely lucky to have two poses that are consistent with the image evidence in both frames and the optical flow. A different approach is needed.

Our first solution is to estimate the pose of the body only at one frame (keeping the dimensionality under control) and to use the optical flow to check how good this solution is in neighboring frames. Given a pose at frame t (Figure 2(a), red) we use the computed dense optical flow (Figure 2(c)) to predict how the DS puppet should move into the next frame, forwards and backwards in time. Our prediction is based on what we call the **puppet flow**, a part-based affine-motion flow associated with a DS model that we estimate by fitting an affine motion model to the computed dense optical flow within each part (Figure 3). The puppet flow (Figure 2(d)) provides the prediction of the puppet in the next frame (Figure 2(b)). We then extend our image likelihood to take into account evidence from the neighboring frames. The advantage is that inference takes place for a single puppet at a time but we are able

to incorporate information from multiple frames. The image evidence is computed in each frame with a multi-scale image likelihood that captures image statistics along the contour of the puppet (see previous chapter).

Our second use of optical flow is in search. Given that we want the foreground model to be consistent with the optical flow, our foreground representation must be able to represent occlusion. Pure part-based models have no notion of what is in front of what. Consequently, we use the DS model but, instead of a distributed collection of parts, our state space represents the full pose of the model; this allows us to model occlusions. Also, we augment the DS model with a scale parameter to capture the overall size of the person in the image.

Our optimization uses a particle-based stochastic search method [48], where each particle represents a whole body configuration. We initialize particles on each frame of the video sequence using a state-of-the-art single-frame pose estimation method [107]. We take the most likely particles in a given frame and use the puppet flow to predict their poses in adjacent frames. This enriches the particle set at neighboring frames. Inference always happens in a single frame but the two strategies described above serve to incorporate information from adjacent frames. We generate additional pose proposals that incorporate information about the possible location of hands based on image and flow evidence; this is our third use of flow.

To summarize, our approach for pose estimation on video sequence is to exploit DS in new ways of integrating information over time. The key idea is to use the optical flow field to define “puppets” that “flow” from one time to the next, allowing us to integrate image evidence from multiple frames in a principled and effective way and to propagate good solutions in time. A good pose is one that is good in multiple frames and agrees with the optical flow. We call these DS puppets that flow in time flowing puppets.

2. Flowing Puppets

As described in Chapter 3, DS is a gender-specific part-based probabilistic model, where contour points of body parts are represented in local coordinate systems by linear models of the form:

$$(31) \quad \begin{bmatrix} \mathbf{p}_i \\ \mathbf{y}_i \end{bmatrix} = \mathbf{B}_i \mathbf{z}_i + \mathbf{m}_i,$$

where \mathbf{p}_i are contour points, \mathbf{y}_i are joint points, \mathbf{z}_i are PCA (Principal Component Analysis) coefficients, \mathbf{B}_i is a matrix of principal components, and \mathbf{m}_i is the mean part contour and joints. Let $\mathbf{l}_i = (\mathbf{c}_i, \theta_i, \mathbf{z}_i)$, where \mathbf{c}_i is the center of the part i and θ_i is the part orientation. The correlation between the *shape coefficients*, \mathbf{z}_i , and body pose parameters captures how shape varies with pose and is modeled with pairwise Multivariate Gaussian distributions over the relative pose and shape coefficients of connected body parts. The probability of a model instance is factored as:

$$(32) \quad p(\mathbf{I}|\Theta) \propto \prod_{(i,j) \in E} p_{ij}(\mathbf{l}_i, \mathbf{l}_j | \Theta_{ij})$$

where E is the set of pairs of connected parts and Θ represents the model parameters. The DS model does not include a scale variable in the potentials, but a scale factor can be specified and it is used to convert the model from the DS model space to image pixel coordinates.

Let \mathbf{x}_t be a vector of DS model variables and the scale at time t (i.e. $\mathbf{x}_t = [\mathbf{l}_t, s_t]$), let I_t be the image frame at time t , and $U_{t,t+1}$ the dense optical flow between images I_t and I_{t+1} . We define the posterior distribution over the DS model variables and scale for each frame in the sequence of N frames as:

$$(33) \quad p(\mathbf{X}|\mathbf{I}, \mathbf{U}, \Theta) \propto \prod_{t=1}^{N-1} p(I_{t+1}|\hat{\mathbf{x}}_{t+1}) p(\hat{\mathbf{x}}_{t+1}|\mathbf{x}_t, U_{t,t+1}) \prod_{t=1}^N p(I_t|\mathbf{x}_t) \prod_{t=1}^N p(\mathbf{l}_t|\Theta) \prod_{t=1}^N p(s_t|\pi_s)$$



FIGURE 4. DS puppet layer. (1) Frame; (2) Corresponding puppet layer with parts ordered by fixed order. The warmer the color, the closer to the camera.

where $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$, $\mathbf{I} = [I_1, \dots, I_N]$, $\mathbf{U} = [U_{1,2}, \dots, U_{N-1,N}]$, $p(\mathbf{I}|\Theta)$ is from Eq. (32), $p(s_t|\pi_s)$ is a prior on scale, $p(I_t|\mathbf{x}_t)$ is the static image likelihood for the frame at time t , $p(I_{t+1}|\hat{\mathbf{x}}_{t+1})$ is the static image likelihood for the frame at $t + 1$, evaluated for $\hat{\mathbf{x}}_{t+1}$, which is the “flowing puppet” of \mathbf{x}_t given the flow $U_{t,t+1}$ (see below). Here our likelihood uses flowing puppets in the forward direction, but our formulation is general and can be extended to consider flowing puppets generated with backward flow and for more than one time step.

Given a DS puppet defined by the variables \mathbf{x}_t , and given the dense flow $U_{t,t+1}$, the corresponding flowing puppet for frame $t + 1$ is generated by propagating \mathbf{x}_t to \mathbf{x}_{t+1} through the flow. The conditional probability distribution $p(\hat{\mathbf{x}}_{t+1}|\mathbf{x}_t, U_{t,t+1})$ expresses the noisy generative process for the flowing puppet $\hat{\mathbf{x}}_{t+1}$.

Exploiting the part-based representation of the DS model, we define a layered model of the body parts with a fixed depth ordering. We assume the torso is the most distant part, then comes the head, the right and the left upper arms, then the right and left lower arms. Figure 4 shows an example of the layer map, where warm colors indicate parts that are closer to the camera. Given the visibility mask for each body part, we consider the corresponding pixels in the optical flow map $U_{t,t+1}$. This is where the DS body shape representation becomes important. Figure 2(c) shows a puppet, \mathbf{x}_t , overlaid on the forward and backward optical flow fields (i.e., computed from t to $t + 1$ and from t to $t - 1$). We fit an affine motion model to the optical

flow vectors within each body part. The resulting puppet flow field is illustrated in Figure 2(d); this is our estimate for how the puppet should move from frame to frame. We then apply the estimated affine motion to the joints of each part, resulting in predicted puppets, $\hat{\mathbf{x}}_{t-1}$ and $\hat{\mathbf{x}}_{t+1}$, at the adjacent frames (Figure 2(b), white). Our process of generating the flowing puppet does not include a noise model, thus the probability distribution $p(\hat{\mathbf{x}}_{t+1}|\mathbf{x}_t, U_{t,t+1})$ is simply a delta function centered on the predicted puppet.

3. Image Likelihood

The static image likelihood is defined by three terms: a contour-based term $p_c(I_t|\mathbf{x}_t)$ that encourages alignment of the DS puppet contour with the edges in the image, a color term $p_s(I_t|\mathbf{x}_t)$ that encodes the knowledge that lower arms and hands are likely to be skin, and a hand likelihood $p_h(I_t|\mathbf{x}_t)$ computed from a hand probability map generated from a hand detector (Figure 6):

$$(34) \quad p(I_t|\mathbf{x}_t) = p_c(I_t|\mathbf{x}_t)p_s(I_t|\mathbf{x}_t)p_h(I_t|\mathbf{x}_t).$$

The DS model we use is learned from a 3D model that does not include hand pose variations, therefore we use a 2D model without separate hand parts with their own articulation parameters. Instead, hands are included in the model of the shape of the lower arm. To define a region-based likelihood for hand color, we simply consider the image area surrounded by the points in the lower arm that correspond to the hand contour.

The contour-based likelihood $p_c(I_t|\mathbf{x}_t)$, as in Chapter 4, is the output of an SVM classifier with a feature vector of HOG descriptors [23] that are steered to the contour orientation and computed along the model contour. In order to obtain a likelihood model that is more robust to scale variations, we compute the features at different scales. Figure 5 shows an example for the upper arm: We use a 3-level pyramid; HOG cells are placed at contour points (blue), inside the contour (red), and outside (green). The contour-based image likelihood is learned from the Buffy dataset [31].



FIGURE 5. Contour-based likelihood features. HOG descriptors are steered to the contour orientation and computed at contour points (blue), inside (red) and outside the contour (green) in a 3-level pyramid.

The skin-color likelihood $p_s(I_t|\mathbf{x}_t)$ is defined by a histogram of skin chroma and hue color coordinates in the CIE L*C*h* color space [28]. We define a log probability map for the image, and compute log skin color likelihood for the hands, and the lower arms without hands, as the average log probability value in the part region. The total skin-color likelihood is then the product of the skin-color likelihood of these parts.

The hand likelihood $p_h(I_t|\mathbf{x}_t)$ is based on a hand probability map generated by a hand detector using optical flow. The hand detector is defined as in [86] by computing the gradient magnitude of the flow, then learning an SVM classifier. The hand probability map is built as the max response from the detector at each image location over a discrete set of hand orientations. The detector is learned from training images from the *VideoPose2.0* dataset [86] where we have manually annotated hands with oriented bounding boxes. Figure 6 shows an example of the hand probability map. We exploit also image cues for hand detection. We train and use a hand detector based on the method described in [63], which uses image features like statistics of image gradients and colors (Figure 7).



FIGURE 6. Hand detection. Example of output from the hand detector trained on optical flow. Image (left), optical flow (center), and hand probability map defined from running a flow-based hand detector on the flow (right).

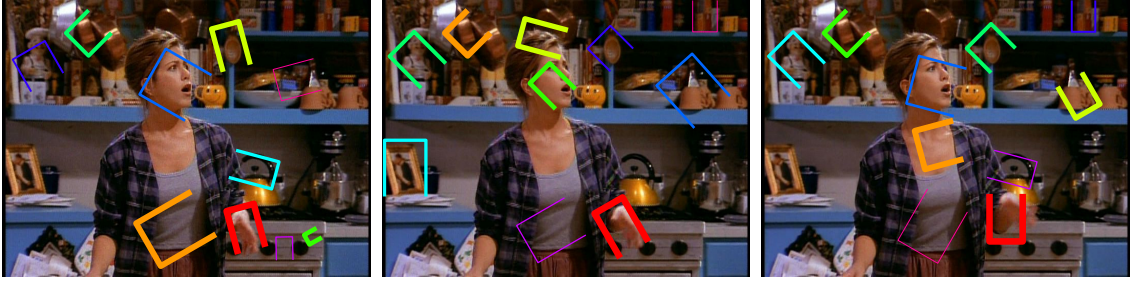


FIGURE 7. Hand detection. Examples of output from hand detector trained on image cues. The red box is the detection with top score; the missing side indicates the position of the wrist.

4. Inference

The DS model state space consists of pose and shape parameters. To reduce the number of variables during inference, we define the configuration of a body as only the location of the joint points \mathbf{y}_t , thus reducing the number of variables. We use a set of 11 joints points, namely nose, neck, right and left shoulders, belly button, right and left hips, elbows and wrists. From these joints, we can easily compute the DS puppet parameters, $\mathbf{l}_i = (\mathbf{c}_i, \theta_i, \mathbf{z}_i)$, where the shape coefficients \mathbf{z}_i are computed with the method described in the previous chapter when describing the annotation tool. In fact, the problem is the same: how to estimate a DS model given the location of the joints. The state space for a puppet in a frame is then $\mathbf{x}_t = [\mathbf{y}_t, s_t]$, where s_t is the puppet scale.

We convert the probabilistic formulation, Equation (33), into an energy:

$$\begin{aligned}
E(\mathbf{X}) = & \lambda_{DS} \sum_{t=1}^N E_{DS}(\mathbf{y}_t) + \lambda_c \sum_{t=1}^N E_c(\mathbf{x}_t) + \\
& \lambda_s \sum_{t=1}^N E_s(\mathbf{x}_t) + \lambda_h \sum_{t=1}^N E_h(\mathbf{x}_t) + \lambda_c \sum_{t=1}^{N-1} E_c(\hat{\mathbf{x}}_{t+1}) + \\
(35) \quad & \lambda_s \sum_{t=1}^{N-1} E_s(\hat{\mathbf{x}}_{t+1}) + \lambda_h \sum_{t=1}^{N-1} E_h(\hat{\mathbf{x}}_{t+1}),
\end{aligned}$$

where $E_{DS}(\mathbf{y}_t) = -\log p(\mathbf{l}_t|\Theta)$ (see Equation (32)), $E_c(\mathbf{x}_t)$, $E_s(\mathbf{x}_t)$ and $E_h(\mathbf{x}_t)$ are the energy terms associated with the contour-based, the skin-color based, and the hand-detector-based image likelihood on the current frame, respectively. $E_c(\hat{\mathbf{x}}_{t+1})$, $E_s(\hat{\mathbf{x}}_{t+1})$ and $E_h(\hat{\mathbf{x}}_{t+1})$ are the negative log likelihoods of the puppet in frame t propagated to the frame $t + 1$ through the dense optical flow $U_{t,t+1}$. We use a uniform prior for scale, as bounds for the scale parameters are set in the optimizer. The weights for the energy terms are estimated with grid search on a validation set.

We seek a maximum a posteriori estimate and minimize the energy using a novel iterative approach based on frame optimization and propagation. For the frame optimization, we adopt Particle Swarm Optimization (PSO) [48].

PSO searches the parameter space by perturbing the particles. Perturbing the vertices can produce implausible puppets so we first convert the pose into an angle representation, do this perturbation in angle space, convert back to joint positions, and then to the expected DS model to obtain contours and regions. Let $\mathbf{x}_i^{(k)}$ be the i -th particle at iteration k . The particle is composed as $\mathbf{x}_i^{(k)} = [\theta^T, \mathbf{l}^T, \mathbf{z}^T, \mathbf{c}_x, \mathbf{c}_y, s]$, where θ is the vector of rotations for each body part, \mathbf{l} is a vector of parts lengths, or sticks, \mathbf{z} is a vector of the shape variables for the torso, \mathbf{c}_x and \mathbf{c}_y are the image coordinates of the torso, and s is the puppet scale.

At each iteration particles are moved according to a computed velocity:

$$(36) \quad \mathbf{x}_i^{(k)} = \mathbf{x}_i^{(k-1)} + \mathbf{v}_i^{(k)},$$

where the velocity is computed as:

$$(37) \quad \mathbf{v}_i^{(k)} = \mathbf{w} \otimes \mathbf{v}_i^{(k-1)} + \mathbf{c}_1 \otimes (\mathbf{x}_{i,best}^{(k)} - \mathbf{x}_i^{(k)}) + \mathbf{c}_2 \otimes (\mathbf{x}_{best}^{(k)} - \mathbf{x}_i^{(k)}),$$

where $\mathbf{x}_{i,best}^{(k)}$ is the best value that particle i has assumed in all previous iterations, and $\mathbf{x}_{best}^{(k)}$ is the best value assumed from all particles in all previous iterations. Here *best* value means that the particle minimizes the energy (Eq. 35). \mathbf{c}_1 and \mathbf{c}_2 are random vectors in $[0, 1]$. \mathbf{w} is a vector of weights, that are set at 1 for θ and \mathbf{l} , 0.5 for \mathbf{z} , 0.1 for the torso location coordinates and 0.5 for the puppet scale.

If a value of a particle exceeds a limit, the velocity is inverted in sign and the particle value is set to be the limit value. These bounds for scale, limb angles, and limb lengths are set based on the VideoPose2.0 training set.

Inspired by [50] we employ a hierarchical strategy optimizing first the torso, head and right arm, then the whole puppet.

We also add a data-driven move for the wrists: with probability $p = 0.5$ instead of moving the particle according to the computed velocity, we keep the current particle state but we sample the location of the wrists from a probability map over hand location. To do this we randomly use the hand map computed with static features or the hand map computed with optical flow features.

The process of optimization and propagation is visually described in Figure 8. We start by initializing a set of P particles on each frame (Figure 8, first row). Then the video sequence is scanned forward and backward to propagate the best M particles from a frame to the next using the flow (Figure 8, second row). Each frame in the sequence is then optimized using PSO. Figure 8, third row, shows examples of particles after PSO. After optimizing pose in each frame, the best M particles are propagated to the neighbors, forward and backward, using the flow (Figure 8, fourth row). After propagation, each frame has $P+2M$ particles, but only the best P particles are retained for the frame. This process of optimization and propagation iterates for a defined number of R runs.

To initialize particles for PSO we run the Flexible Mixtures of Parts (FMP) model [107] and from its solution we generate particles adding noise and placing wrists at sampled locations obtained from the hand probability maps. We use the FMP code provided by the authors with their model trained on the Buffy dataset. The FMP model generates a “stickman” as output. In order to map the FMP stickman to the DS model at a proper scale, we learn a regression function between the scale of the stickman and DS from manually annotated frames of the VideoPose2.0 training set.

After each run of PSO all the particles are resampled according to their energy, that we interpret as negative log posterior.

In our experiments we used $P=40$, $M=5$, $R=8$. We run the optimization with 3 different seeds for the random number generator and select the solution with the minimum energy.

The DS model is learned for random poses and cameras, and the camera parameters used to generate the training samples might not be consistent with those of the camera that was used to shoot the video. We introduce prior knowledge on camera location and pose in TV shows by redefining the mean DS shape as the average shape in the Buffy training set [31] annotated with the DS model.

5. Experiments

The *VideoPose2.0* dataset [86] contains 1225 frames from two popular TV shows (Friends and Lost) corresponding to 44 clips. The dataset is divided into 706 training frames and 519 test frames in 18 clips. For consistency with [86], we use the dataset with only every other frame of the original video sequences. It contains frames at the original size, and frames that have been cropped and rescaled to have the person in the middle of the frame to meet the needs of the pose estimation method of [86]. In contrast to [86] we use the original frames, since we model scale explicitly and estimate it during optimization. We annotate the clips for gender and use a DS model of the

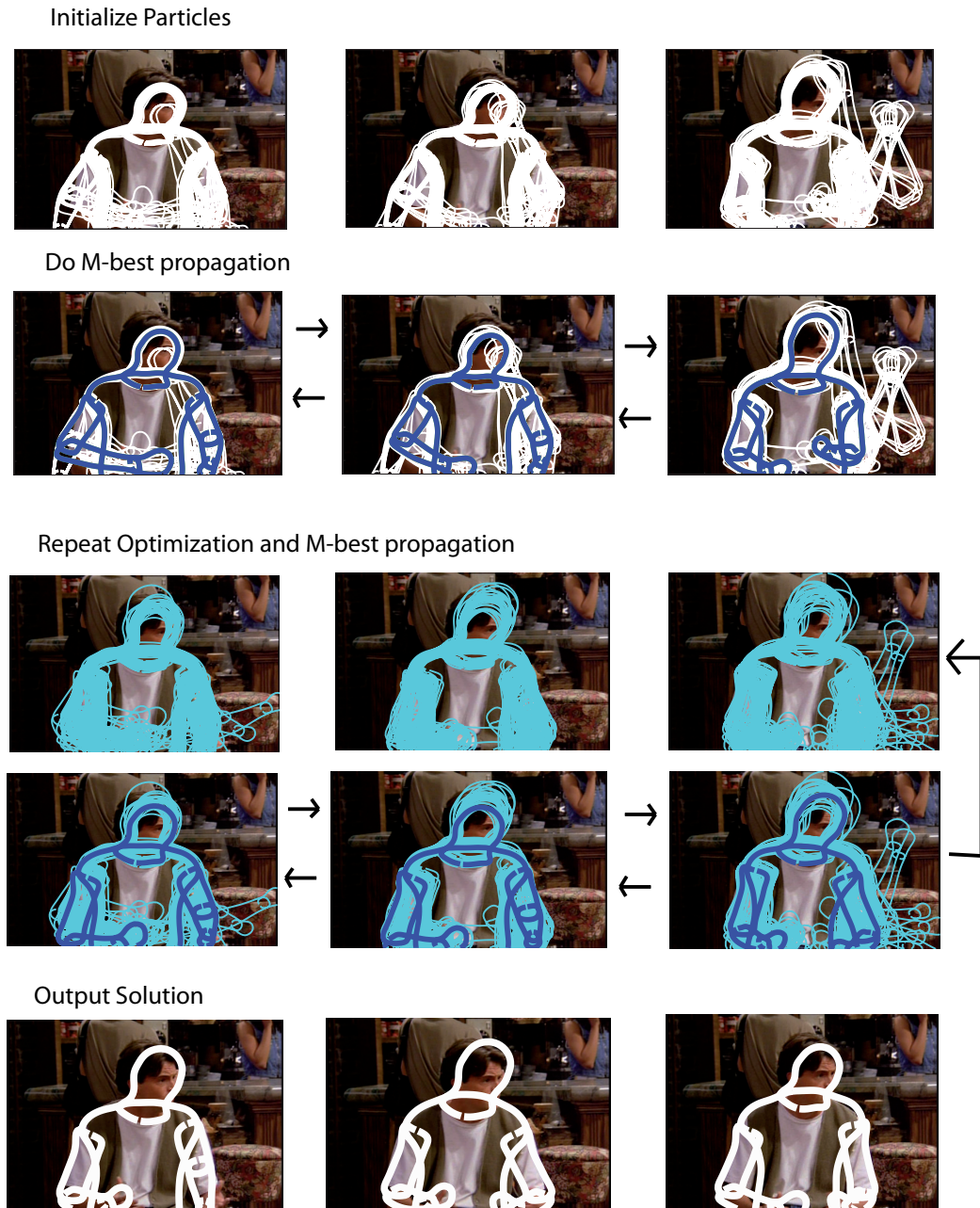


FIGURE 8. Particle-based optimization. Particles are initialized on each frame (first row), then the M best are propagated through the flow forward and backward (second row). For a defined number of iterations particles are then locally optimized, then the M best are propagated to the neighbors (third and fourth row). Then the best particle on each frame is returned as the solution (last row).

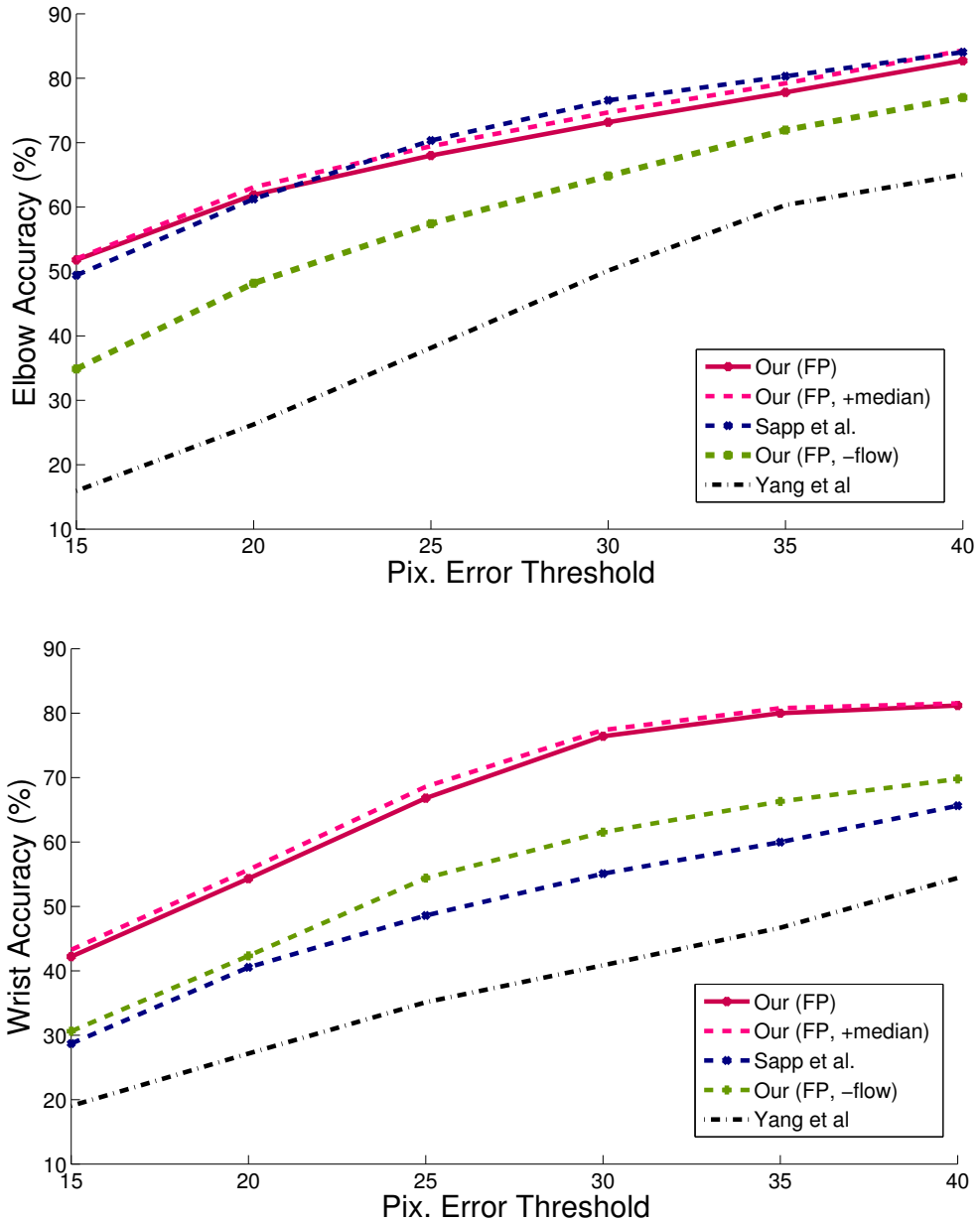
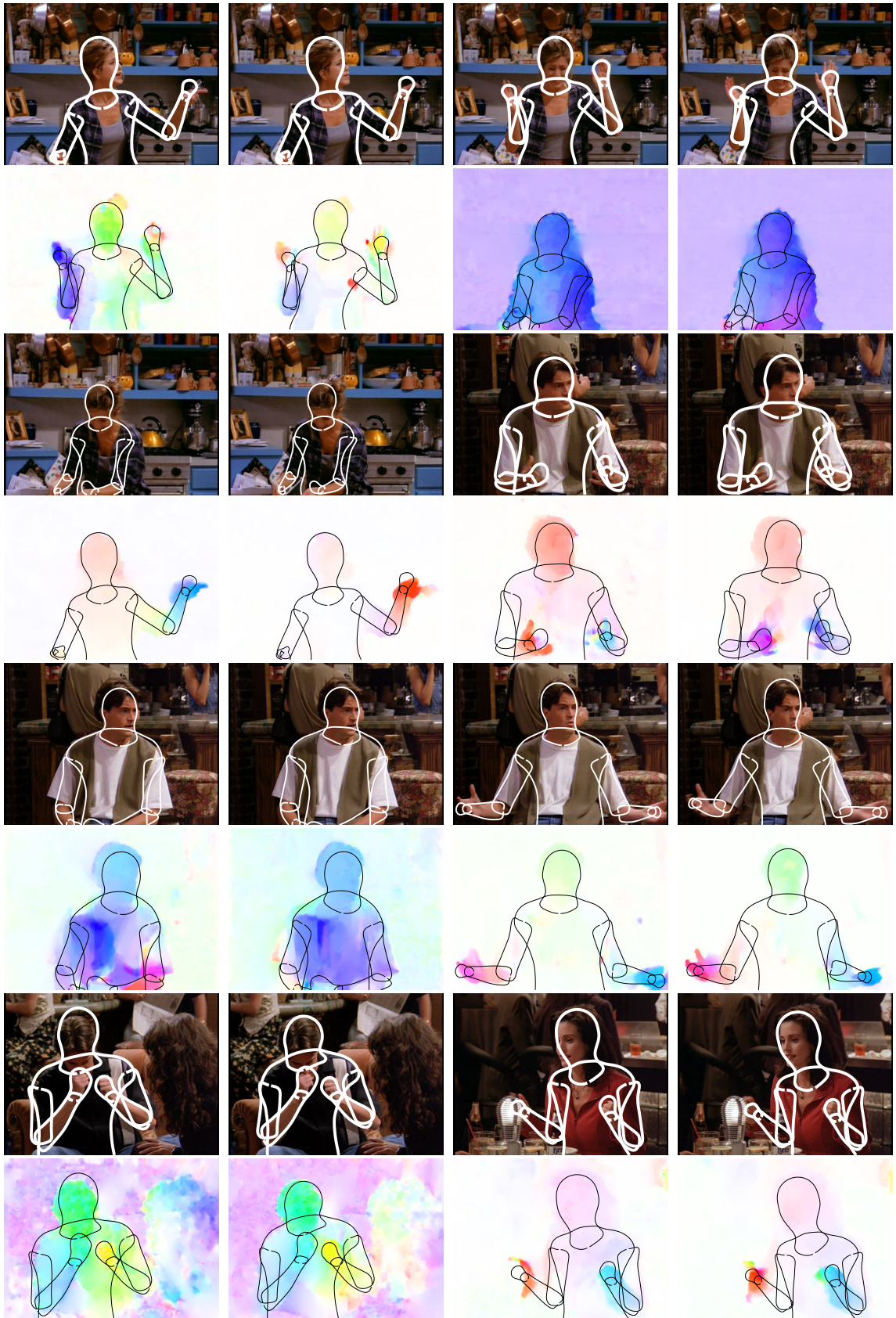


FIGURE 9. Results. Accuracy of elbow (left) and wrist detection (right) for different threshold distances from ground truth. We significantly improve over our baseline (Yang and Ramanan [107]) and over the state-of-the-art (Sapp et al. [86]) in wrist detection. FP stands for Flowing Puppet.

appropriate gender. The dense flow is computed with the method of [106] in both the forward and backward time direction.

Results are reported as in [86] as the percentage of joints that have a distance lower than a threshold in pixels from the ground truth (Figure 9). As a baseline we report results for Yang et al. [107], which we use during initialization. Note that it performs rather poorly but was not trained for this dataset. We compare different variants of our method (FP for Flowing Puppets). First, we report results without and with median filtering as applied in [86], where a median filter is applied to the location of the joints over time with a 3-frame temporal window. Second, to show the benefit of our optimization strategy, we show results obtained without exploiting the dense flow for propagation and likelihood (FP, -flow). We significantly improve over [107], and are at the performance level of [86] for elbows. We can observe that our approach performs significantly better than the static image detector of [107]. Fragkiadaki et al. [33] also perform pose estimation on the VideoPose2.0 dataset, but for testing they select a set of the clips that is different from the one specified in the dataset; a direct comparison is not possible. Recently Cherian et al. [22] have proposed a method that reports state-of-the-art performance on the VideoPose2.0 dataset.

Figure 10 shows several examples of correctly predicted body pose, with the DS puppet overlaid on the image and on the optical flow. Figure 11 shows some representative failure cases.



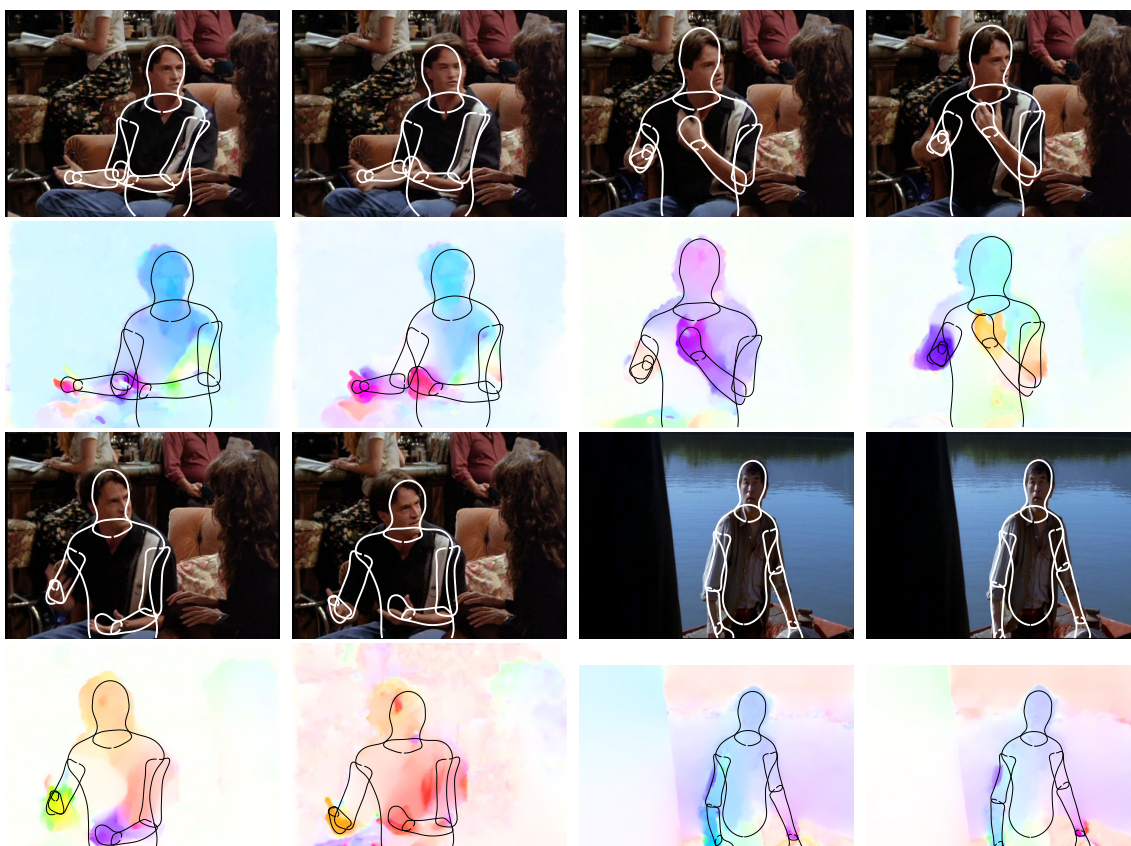


FIGURE 10. Estimated body pose. Successful detection results from 9 test clips are shown (2 frames per clip). Images are shown with the estimated puppet overlaid in white. Below each image is the estimated forward flow field color coded as in [10] with the puppet overlaid in black.



FIGURE 11. Estimated body pose. Examples of failure cases. In all cases the image evidence supports incorrect poses.

Limitations of the Deformable Structures Model

The DS model is view-specific: we have learned a model representing the 2D shape of a person facing the camera, and we have used it to infer pose from images and videos where subjects were seen in frontal view. In a more general setting, one should be able to estimate the pose of people in arbitrary, unknown, viewpoints.

We have learned a set of DS models corresponding to different camera viewpoints, and these have been used to annotate videos with people performing various actions in [49].

While using a set of models would constitute a reasonable approach to the pose estimation problem, there is a redundancy in the representation: we can consider the torso and shape to be viewpoint-specific, but the shape of the other limbs, maybe with the exception of the head, would be partially independent from the viewpoint, as for example we can have a pose with an arm pointing to the camera from a frontal view and also from a lateral view. We have therefore explored the definition of a single multi-view DS model, where the torso shape is viewpoint-dependent, and other parts shapes are instead independent from the camera orientation.

1. Multi-view DS Model

In a multi-view DS model the shape of the torso is defined as a mixture model, with a mixture component for each of the camera viewpoints in a discrete set of cases considered. The shape of the other limbs is view-independent, as given the articulated nature of the body, limbs can be oriented with respect to the camera in arbitrary ways, independently from the torso orientation, with some exceptions for example for upper arms pointing to the camera when the viewpoint is from the back, as this pose would be not possible for anatomical constraints.

In order to learn a multi-view DS model, we generated training data for nine uniformly spaced views between the West and East orientation, that we labeled as W,WSW,SW,SSW,S,SSE,SE,ESE,E. We then augmented the DS model’s set of variables with a view label v that indicates the orientation of the torso among the nine considered.

Given the view label of each training sample, we clustered the samples by view to define a Gaussian mixture model over the PCA coefficients that define the torso shape. Let s be the index of the torso part; the distribution of the torso shape coefficients and the conditional distribution given a view are expressed respectively as:

$$(38) \quad p(\mathbf{z}_s) = \sum_{V=1}^9 \pi_v p(\mathbf{z}_s | v = V),$$

$$(39) \quad p(\mathbf{z}_s | v = V) = \mathcal{N}(\mathbf{z}_s | \boldsymbol{\mu}_V, \Sigma_V),$$

Figure 1 shows the torso shapes generated from the mean PCA coefficients $\boldsymbol{\mu}_V$ for each view. Figure 2 shows the corresponding DS puppets, which have been generated specifying the view only for the torso. Given the torso shape coefficients, the upper arms and upper legs are defined from the corresponding conditional pairwise potentials $p_{t|s}([\mathbf{z}_t, \sin(\theta_{ts}), \cos(\theta_{ts}), \mathbf{q}_{ts}] | \mathbf{z}_s = \boldsymbol{\mu}_V)$, for s being the torso index and t the index of a children. The part lengths have been marginalized out. Similarly lower arms and lower legs are defined from their parent shape. Note that (Figure 2) the information about the torso viewpoint propagates to the lower legs to set the right direction for the feet.

We define view-specific pairwise potentials for the torso and its neighbors. The probability of the multi-view model is then expressed as:

$$(40) \quad p([\mathbf{l}, v] | \Theta) = \sum_{V=1}^9 \mathbb{I}(v = V) p(\mathbf{l} | \Theta_V),$$

where $\Theta = (\Theta_1, ..\Theta_9)$ and

$$(41) \quad p(\mathbf{l} \mid \Theta_V) = \frac{1}{Z(\Theta_V)} \prod_{s=\text{torso}, t>s, (t,s) \in \mathcal{E}} p_{stV}(\mathbf{l}_s, \mathbf{l}_t) \prod_{s>\text{torso}, t>s, (t,s) \in \mathcal{E}} p_{st}(\mathbf{l}_s, \mathbf{l}_t)$$

View-specific pairwise potentials are defined as:

$$(42) \quad p_{stV}(\mathbf{l}_s, \mathbf{l}_t) = \mathcal{N}(T_{st}(\mathbf{l}_s, \mathbf{l}_t) \mid \boldsymbol{\mu}_{stV}, \Sigma_{stV}),$$

where the distribution is defined over relative quantities given by:

$$(43) \quad T_{st}(\mathbf{l}_s, \mathbf{l}_t) = [\mathbf{z}_t, \sin(\theta_{ts}), \cos(\theta_{ts}), \mathbf{q}_{ts}, \mathbf{z}_s, d_t, d_s],$$

and $\boldsymbol{\mu}_{stV}$ and Σ_{stV} are the mean and covariance of the training samples for the specified view.

The multi-view DS model still has some limitations, as while different orientations of the person with respect to the camera have been allowed, the model remains specific for the other camera parameters that were assumed to generate the training samples as projections of a 3D body model. In particular the height of the camera can be an application-dependent parameter of some importance. In fact, if we have learned a DS model for a camera approximatively at the height of the subjects, we have a proper model for TV shows, but a bad model for surveillance cameras, that have typically a much higher position.

What we have done with the multi-view model is to try to "fix" the 2D model to recover some of the invariances we lost with the projection from 3D to 2D. In doing this we have created a more complex model with the additional viewpoint variable.

The question then is: Would it be possible to build a 3D version of the DS model without increasing too much the number of variables so we would have a body model that is part-based, with realistic deformations, but independent from the camera? We answer positively this question with the work described in the next thesis chapter, where we introduce a 3D part-based model.

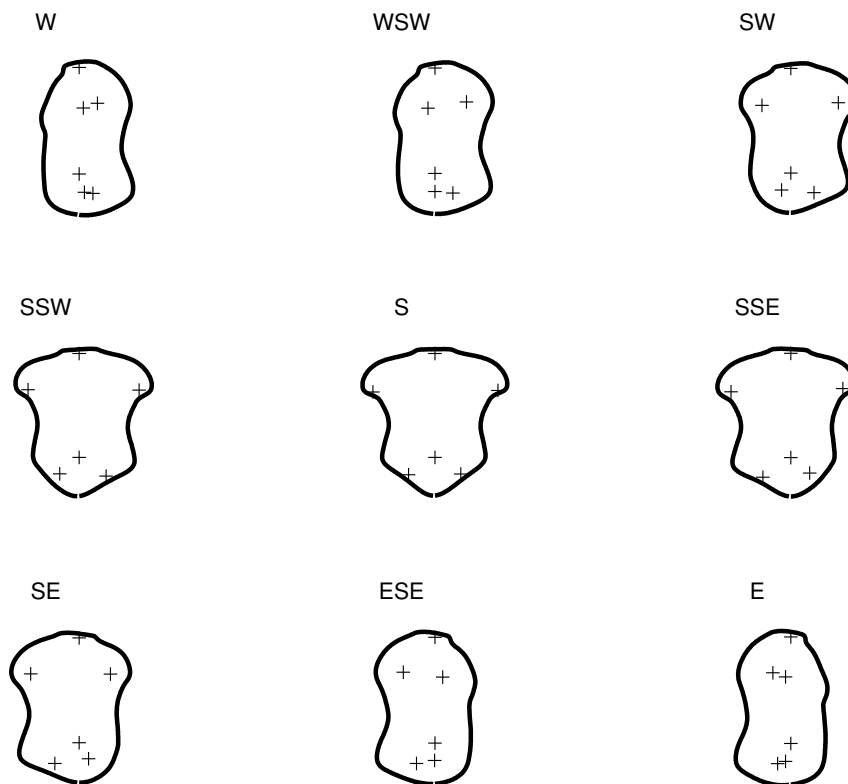


FIGURE 1. Multi-view DS model. Each figure shows the mean torso for each mixture component, corresponding to 9 discrete camera viewpoints uniformly spaced between the West and East direction (S = South).

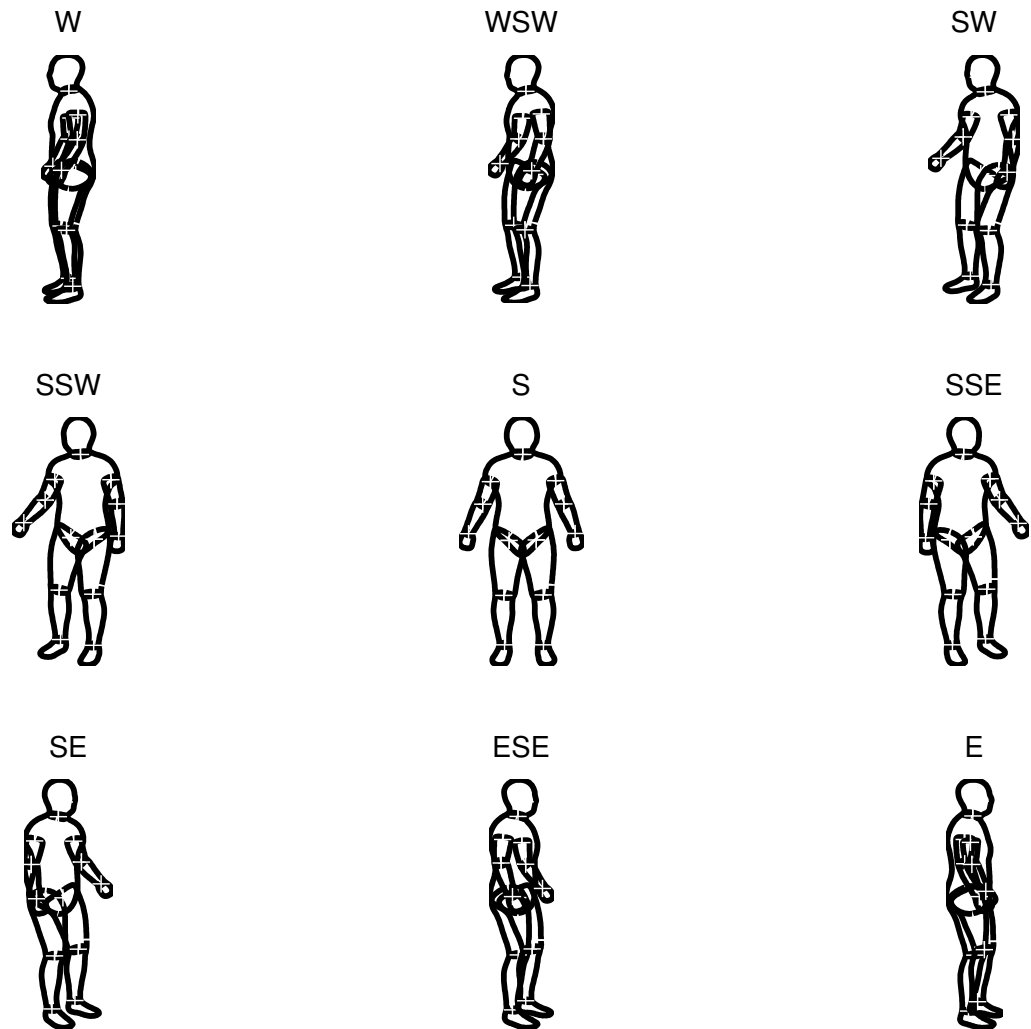


FIGURE 2. Multi-view DS model. DS puppet generated from the mean torso of each mixture component. Note how the orientation of the feet are correctly generated, even if the viewpoint is only accounted for at the torso level, with a view-dependent shape (Figure 1). The shape of the torso determines the appropriate shape for the upper legs (see text).

Stitched Puppet Model

In this chapter we introduce the second contribution of this thesis in terms of new models of the human body: the Stitched Puppet (SP) model, a part-based, 3D model of the human body with a realistic representation of body shape.

In the context of 3D models, there are two main classes in use. The first represents 3D body shape and pose-dependent shape variation with high realism (Figure 1(a)) [3, 8, 21, 42, 70]. Such models have been introduced mostly to the computer graphics community, and are described using a relatively high dimensional state space, combining shape and pose parameters, making inference computationally challenging [75]. The second class of models, which are instead popular in computer vision, is based on simple geometric parts that can be estimated independently from data (Figure 1(b)) [90, 91]. This approach breaks the global state space into smaller ones describing the shape and pose of each part separately. Such models have a graph structure that connects the parts via potential functions and inference is done using message passing algorithms such as belief propagation. These models are advantageous for inference but have a crude geometric structure that does not make it possible to recover body shape and that does not match well to image evidence.

Part-based models are not totally absent in computer graphics, where a segmentation into parts can help in creating more accurate 3D models. An example is the work of Tena et al. [95], where 3D face models are built breaking the mesh of a face into regions, and then learning low-dimensional representations over deformations for each region with Principal Component Analysis (PCA). Regions can deform independently under user’s control, and are then blended to stitch at their interfaces; the advantage of this representation is that the local deformation spaces allow a localized control for the user to specify mesh deformations (Figure 2). Local deformation spaces are

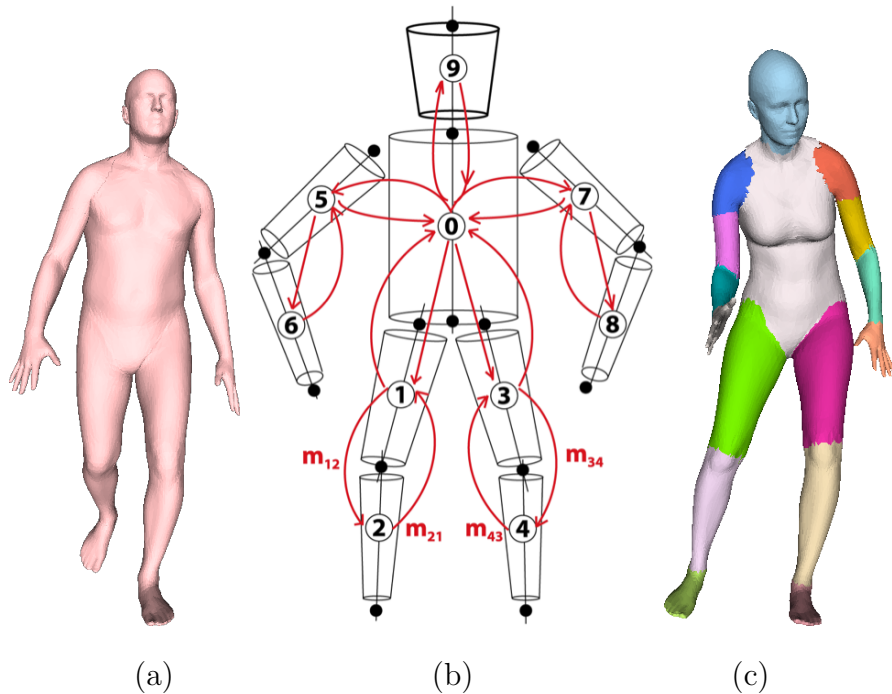


FIGURE 1. 3D Body Models. (a) A SCAPE body model [8] realistically represents 3D body shape and pose using a single high-dimensional state space. (b) A graphical body model composed of geometric primitives connected by pairwise potentials (image reproduced from [91]). (c) The **stitched puppet model** has the realism of (a) and the graphical structure of (b). Each body part is described by its own low-dimensional state space and the parts are connected via pairwise potentials that “stitch” the parts together.

also used in [65], with the difference that the localization is obtained by enforcing sparsity of the components, learned over the whole mesh, rather than throughout a segmentation of the mesh into regions. As a consequence, there is no need of stitching mesh regions.

The SP model offers the best features of both approaches in that it is both part-based and highly realistic (Fig. 1(c)). It is learned from a detailed 3D body model based on SCAPE [8]. Each body part is represented by a mean shape and a subspace of shape deformations learned using PCA. These shape variations allow SP to capture and fit a wide range of human body shapes. Each part can also undergo

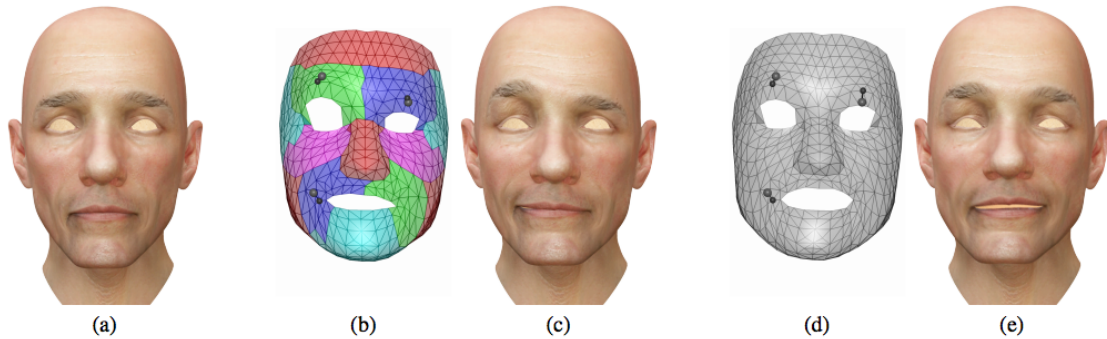


FIGURE 2. Region-based face model (image and caption reproduced from [95]). Face posing using interactive region-based (b) and holistic (d) face models. The models drive the human character shown in (a). User-given constraints (black markers) create a wink with a smirk, when issued to the region-based model (b and c). In contrast, the same constraints produce uncontrolled global deformations when the holistic model is used (d and e).

translation and rotation in 3D. As with other part-based models, the parts form a graph with pairwise potentials between nodes in the graph. The SP potentials represent a “stitching cost” that penalizes parts that do not fit properly together in 3D to form a coherent shape.

Unlike the SCAPE model, parts can move away from each other but with some cost. This ability of parts to separate and then be stitched back together is an important property that is exploited during inference to better explore the space of solutions. As a result, SP can robustly fit body shape and pose given low-resolution and noisy data, without a good initialization.

1. Definition

The Stitched Puppet (SP) model is a part-based 3D model of the human body parameterized by pose, intrinsic shape, and pose-dependent shape deformations. Intrinsic shape is the body shape that varies between people due to gender, age, height, weight, fitness, etc. The model is composed by 16 body parts: head, torso, shoulders,

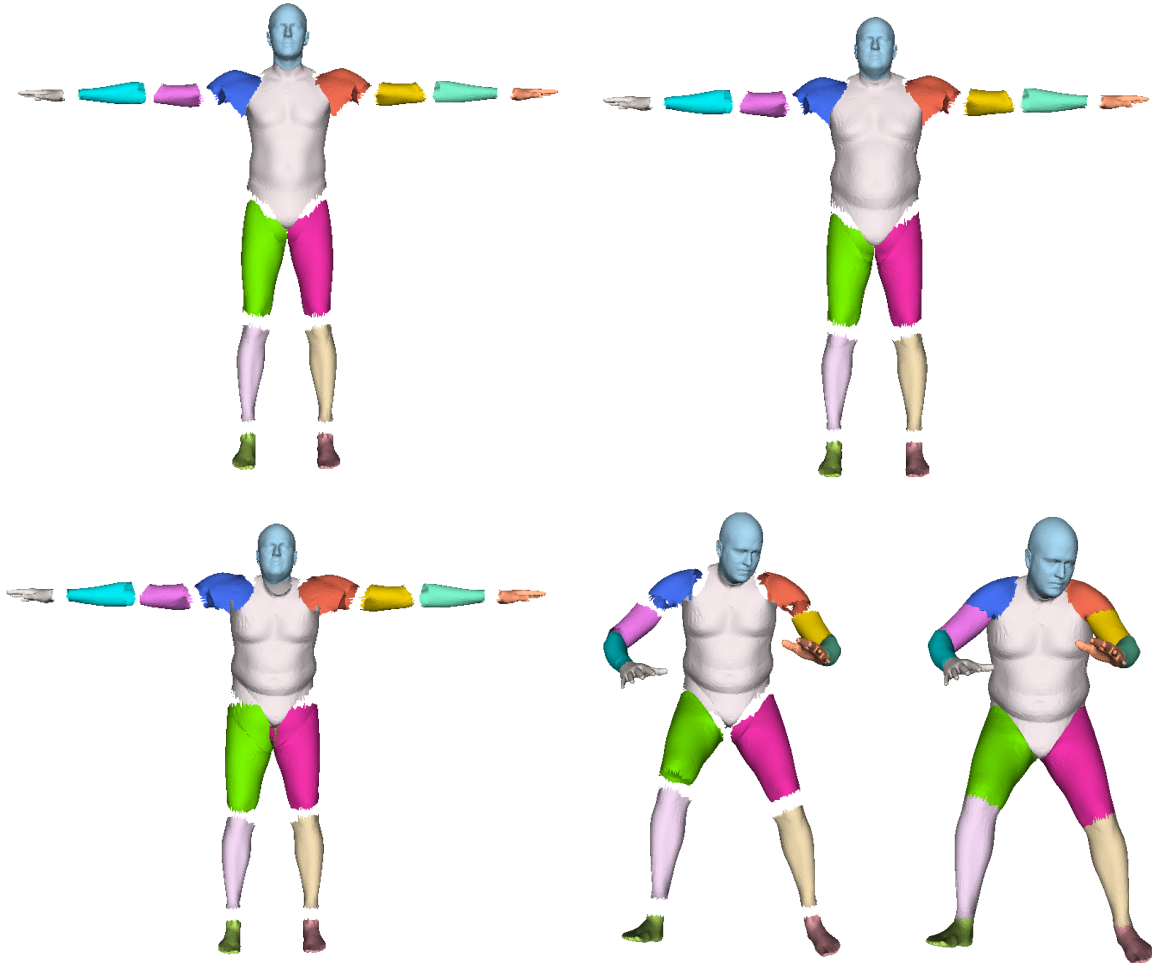


FIGURE 3. Stitched Puppet Model. To generate an instance of SP we start with the template body (top left), which is segmented into parts. We apply the intrinsic shape deformation to change the body shape (top right). We generate pose deformations for each body part (see text) (bottom left). The pose of the body is defined by the rotation and translation that stitches the parts together (bottom, middle and right).

upper arms, lower arms, upper legs, lower legs, hands and feet (see color coding in Fig. 1(c)).

SP is a tree-structured graphical model in which each body part corresponds to a node, with the torso at the root. Each part is represented by a triangulated 3D mesh in a canonical, part-centered, coordinate system. Let i be a node index, with

$i \in [0..15]$. The node variables are represented by a random vector:

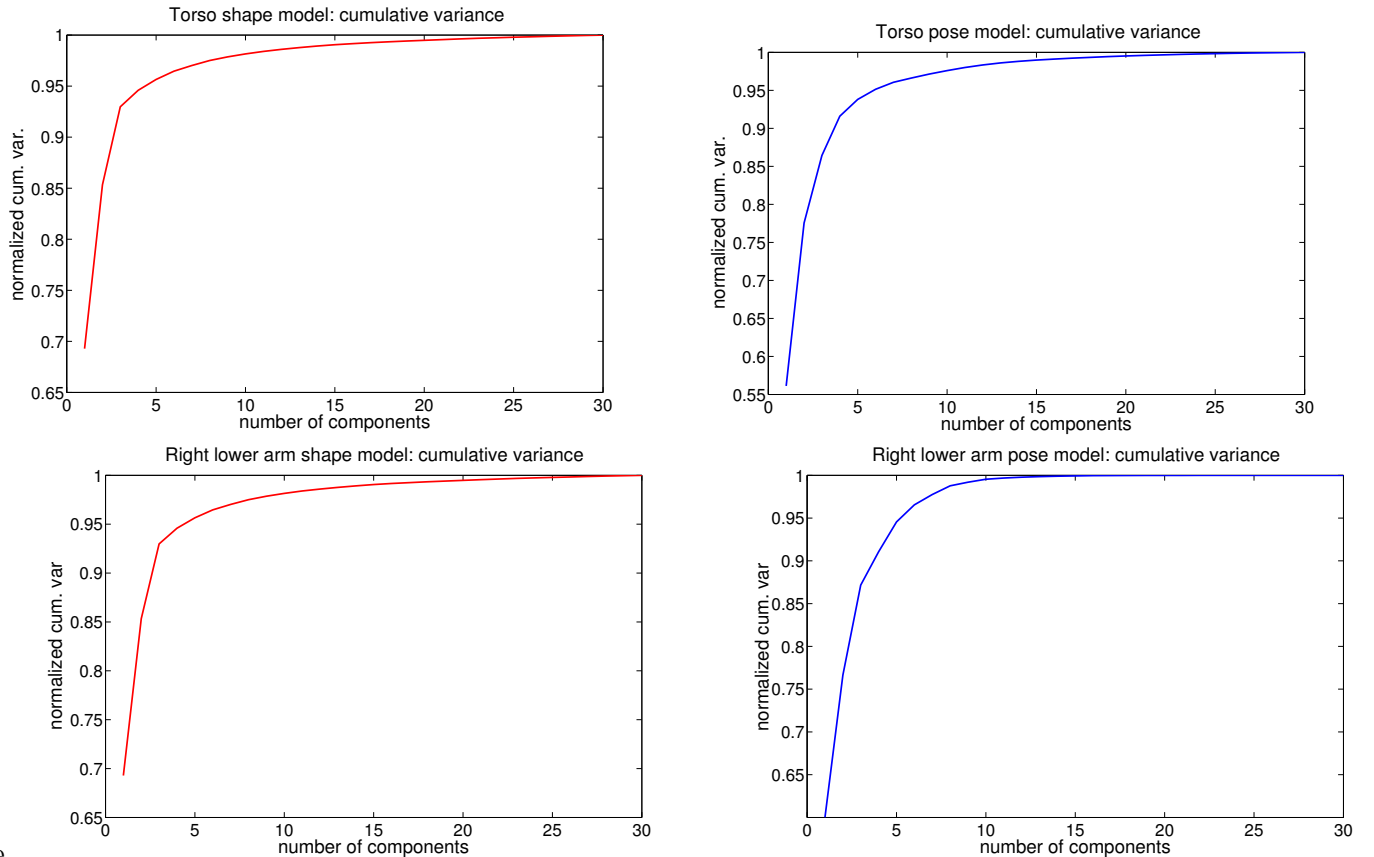
$$(44) \quad \mathbf{x}_i = [\mathbf{o}_i^T, \mathbf{r}_i^T, \mathbf{d}_i^T, \mathbf{s}_i^T]^T$$

where \mathbf{o}_i is a three-dimensional vector representing the location of the center of the part in a global frame and \mathbf{r}_i is a three-dimensional Rodrigues vector representing the rotation of the part with respect to a reference pose. The reference pose is the pose of the part in the *template* mesh (Fig. 3(a)). The parts also have two vectors of linear shape coefficients, \mathbf{d}_i and \mathbf{s}_i , that represent pose-dependent deformations and intrinsic body shape, respectively. We learn these shape deformation models using Principal Component Analysis (PCA).

From model variables to meshes. Given a set of node variables \mathbf{x}_i (Eq. 44), the mesh vertices for the part i are generated as follows. First, we use the intrinsic shape parameters \mathbf{s}_i to generate a deformed template mesh \mathbf{q}_i with the desired intrinsic shape (Fig. 3(b)):

$$(45) \quad \mathbf{q}_i = B_{s,i} \mathbf{s}_i + \mathbf{m}_{s,i},$$

where $\mathbf{m}_{s,i}$ is a vector of 3D vertices of part i in a local frame with origin in the part center corresponding to a part with mean intrinsic shape across all training body shapes. The 3D coordinates of the vertices are stacked in the $\mathbf{m}_{s,i}$ vector as $[x_1, y_1, z_1, x_2, y_2, z_2, \dots]$, therefore $\mathbf{m}_{s,i}$ is a column vector of $3N_i$ vertex coordinates, where N_i is the number of vertices of the part i . $B_{s,i}$ is the matrix of PCA basis vectors of size $(3N_i, n_s)$, where we take $n_s = 4$ for all the parts. Figure 4 shows the cumulative variance of the PCA spaces for the female torso: with 4 principal components for the shape model we capture about the 95% of the variance we would represent with 30 components. The PCA subspace for the intrinsic shape, $B_{s,i}$, is learned over meshes in a template pose, where we assume there are no pose-dependent deformations.



Figure

FIGURE 4. PCA models. Normalized cumulative variance for the PCA models of the female torso (top) and lower arm (bottom) for shape (left) and pose (right).

Unlike SCAPE [8], where shape deformations are transformations operating on triangles, SP is much simpler¹. Since each body part has its own coordinate system, the deformations can be applied directly to *vertices* in this coordinate frame. The resulting vector \mathbf{q}_i represents the vertex coordinates in the local frame of the mean part deformed to represent some intrinsic body shape.

We next apply a pose-dependent deformation to the modified template, \mathbf{q}_i (Figure 3(c)):

$$(46) \quad \mathbf{p}_i = B_{p,i} \mathbf{d}_i + \mathbf{m}_{p,i} + \mathbf{q}_i,$$

¹Unlike SCAPE, SP has no need for a least-squares optimization to stitch triangles into a coherent mesh. This has significant computational advantages and results from the fact that part shapes are defined in their own coordinate systems.

where $B_{p,i}$ is the matrix of PCA basis vectors for the pose-deformation model of part i , $\mathbf{m}_{p,i}$ is the mean of the deformation in the training set with respect to the template, and the resulting \mathbf{p}_i represents the local coordinates of the part after shape deformations have been applied. For the PCA model we use 12 principal components for the torso and 5 principal components for the other body parts. Now, given the part center \mathbf{o}_i and the Rodrigues vector \mathbf{r}_i , a rigid 3D transformation is applied to the vertices in local frame \mathbf{p}_i to convert them into a global coordinate frame, $\tilde{\mathbf{p}}_i$ (Figure 3(d,e)). The global coordinates are computed as:

$$(47) \quad \tilde{\mathbf{p}}_{i,k} = R(\mathbf{r}_i) \mathbf{p}_{i,k} + \mathbf{o}_i,$$

where k is an index in the set of vertices of the part i , and R is a rotation matrix computed from the Rodrigues vector \mathbf{r}_i .

In the SP model the pose-dependent deformations are independent from the intrinsic shape. In reality, the pose-dependent deformations of a large body are different from those of a small body for the same pose. A more appropriate model for accurate pose-dependent deformations would be a bilinear model, where the displacement to add to the vertexes to model deformations depends from the intrinsic shape. As explained in the following section, we learn SP from computer generated training data obtained from a model (SCAPE) that also considers intrinsic shape and pose deformations as independent factors. However, in the SCAPE model the pose-dependent deformations are expressed over triangles, and are also applied before the deformations for intrinsic shape. Using triangles provides a more scale-invariant way of applying the pose-dependent deformations with respect to modeling displacements. After the deformations are applied to the triangles, the triangles are disconnected. An optimization step then brings them in place for representing a connected mesh. From the perspective of vertexes displacements, the transformation applied to a vertex is likely to result in a intrinsic shape deformation and a pose-dependent deformation that is not independent from shape. In this version of the SP model we consider the

intrinsic shape and pose deformations as independent. In future work, we plan to learn a bilinear SP model, directly from 3D scans.

2. Learning

We learn SP from instances of a SCAPE model [8]. Using SCAPE we have the advantage that the training data is a set of training meshes in a wide range of body shapes and poses that are in complete correspondence, and we can also exploit an existing segmentation of SCAPE into parts. The SCAPE model we consider has a template mesh in a T-pose, and is segmented into 19 parts. For SP we take the same template mesh but segment it into fewer parts; in particular, we treat the torso as a single part (Figure 5), merging the upper torso and the pelvis.

To create training data, we sample a set of 600 poses from motion capture data² and for each pose we generate 9 more by adding small amounts of noise to the part rotations. This gives 6000 SCAPE meshes in different poses for training. We also generate 600 samples of bodies with different intrinsic shapes in the template pose by sampling from the SCAPE body shape model. From this set of samples, where we only vary shape, we learn the intrinsic shape model of our parts. We learn separate models for men and women.

We define the SP mesh topology as a “chopped” version of SCAPE, where each part is an independent mesh with a locally defined assignment of vertices to faces. For neighboring body parts, we duplicate the vertices that are in common. We call the set of vertices that are duplicated “interface points” and these are important for stitching SP parts together. The SCAPE model represents pose dependent deformations as a linear function of relative part rotations in the model. Here, our distributed graphical model requires a different solution where part deformations are only functions of variables local to the node, thus each part has its own pose-dependent shape deformation space.

²The data was obtained from <http://mocap.cs.cmu.edu>. The database was created with funding from NSF EIA-0196217.



FIGURE 5. Pose-dependent part deformation. An example of the torso PCA model is shown (first three principal components from top to bottom). The center mesh in each row is the mean shape and the left and right meshes correspond to ± 3 standard deviations from the mean.

Given the part segmentation of the training meshes, we take each part and transform it to a canonical coordinate frame. Figure 6 shows some training samples.

The vertices in this frame for each part form one training example. We compute the mean shape as the mean location of each vertex, subtract this, and perform PCA. For the pose-dependent shape deformation we learn 16 independent PCA models, $B_{p,i}$, one for each part. We use 5 shape basis vectors to represent the pose-dependent

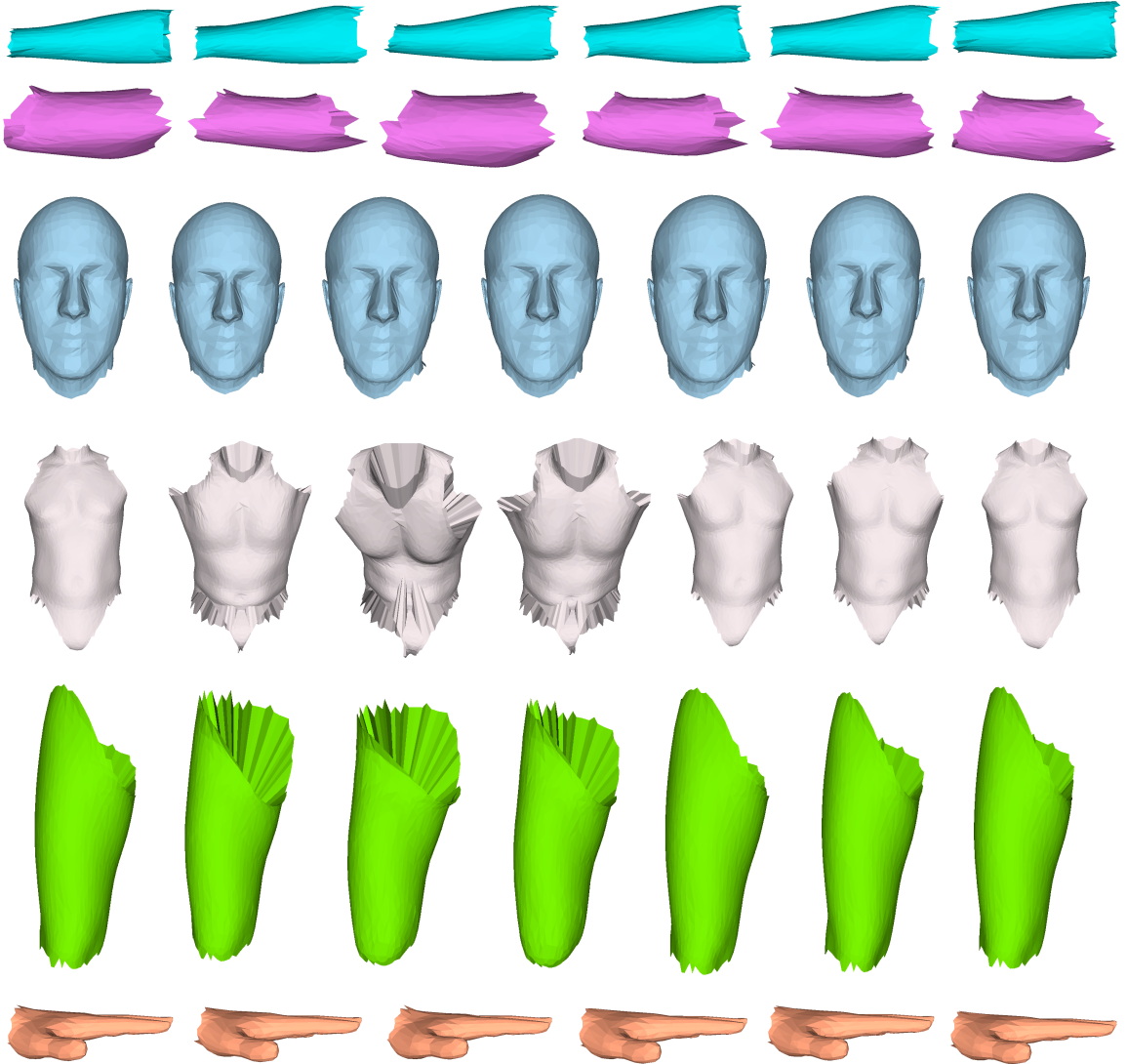


FIGURE 6. Training samples. Examples of few training samples from the dataset with variation in pose. Meshes are all defined in a local coordinate system with origin in the middle of the body part.

deformations of each part, except for the torso for which we use 12. Figure 5, 8, 10 illustrate the pose PCA space of the torso, head and upper left leg, respectively. The range of shapes of the torso varies much more than the other parts due to the flexibility of the spine.

For the intrinsic shape model we learn the PCA model over the full body, obtaining a single matrix of PCA components B_s of size $(3 \sum_{i=0:15} N_i, n_s)$. For the intrinsic shape \mathbf{s}_i , we use 4 basis vectors for each part, $n_s = 4$. Subsets of rows in the

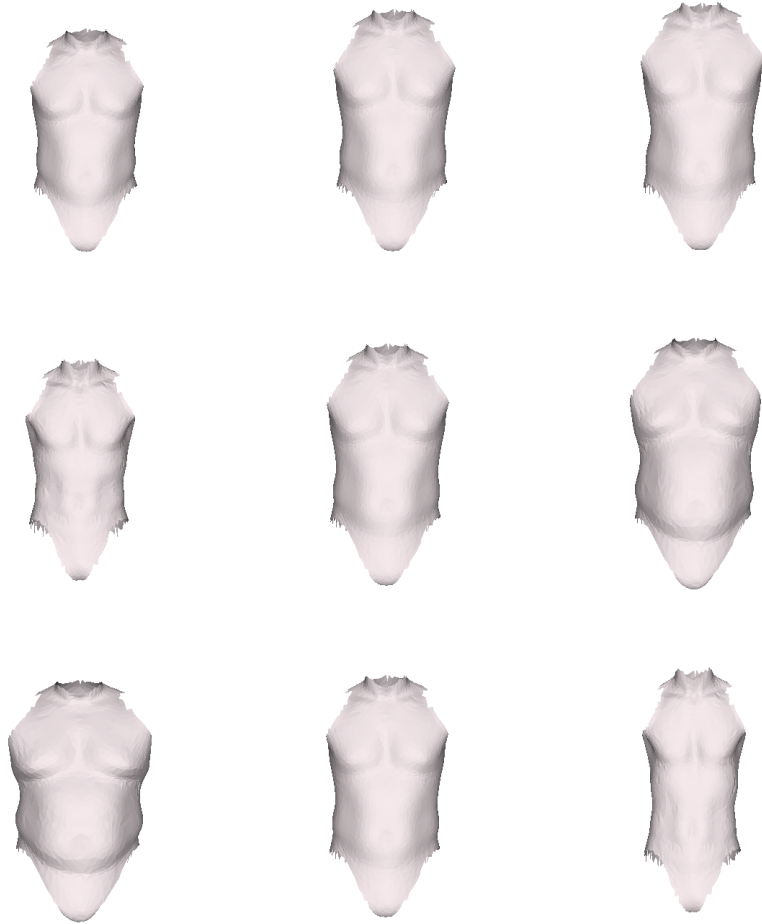


FIGURE 7. Intrinsic shape part deformation. An example of the torso PCA model is shown (first three principal components from top to bottom). The center mesh in each row is the mean shape and the left and right meshes correspond to ± 3 standard deviations from the mean.

global shape basis matrix B_s correspond to different body parts and define the PCA component matrix, $B_{s,i}$, for each body part. This approach means that, if each node in SP has the same intrinsic shape coefficients, this will correspond to a coherent body shape. This, however, is not enforced by SP during inference and parts can take on different shapes as needed. Figure 7, 9, 11 illustrate the shape PCA space of the torso, head and upper left leg, respectively.

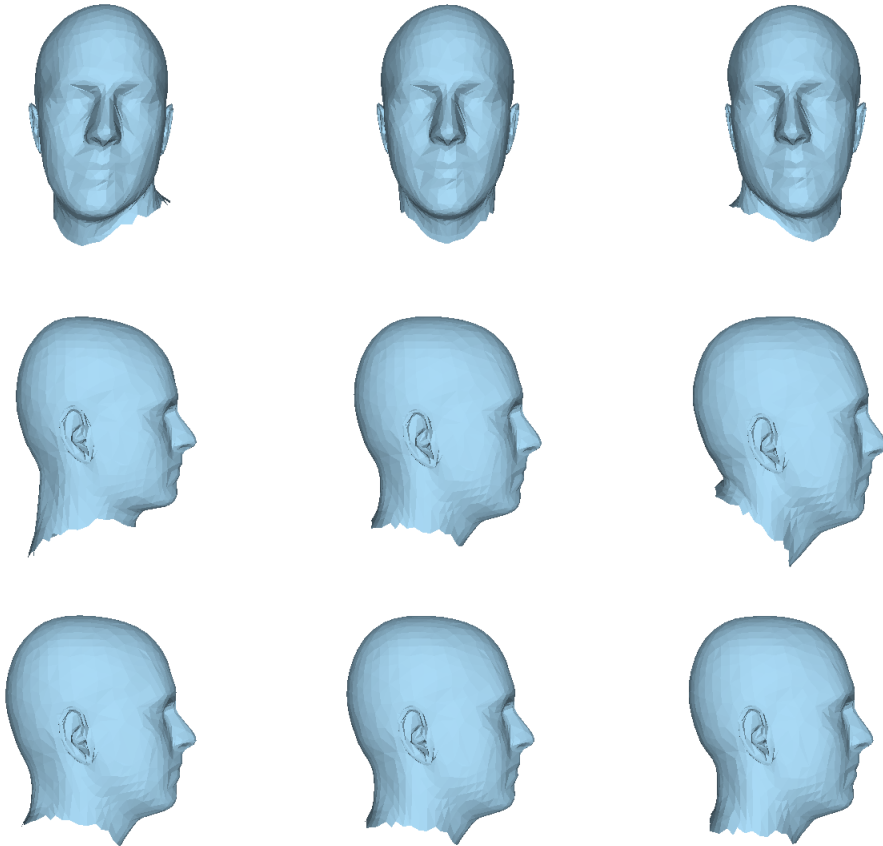


FIGURE 8. Pose-dependent part deformation. An example of the head PCA model is shown (first three principal components from top to bottom). The center mesh in each row is the mean shape and the left and right meshes correspond to ± 3 standard deviations from the mean.

3. Stitching Potentials

We define the SP as a tree-structured graphical model with pairwise potentials. Implicit in the idea of SP is a stitching potential to glue parts together. This potential cannot be learned from the training set, since the training parts are already stitched together. Consequently we define it manually to allow body parts to be loosely connected (Figure 12); cf. [90]. We define the stitching potential as:

$$(48) \quad \Psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) = -\frac{1}{\|\mathbf{u}_{ij}\|_1} \sum_{k=1..N_{ij}} u_{ij}(k) \|\tilde{\mathbf{p}}_{i,I_{ij}(k)}(\mathbf{x}_i) - \tilde{\mathbf{p}}_{j,I_{ji}(k)}(\mathbf{x}_j)\|^2,$$

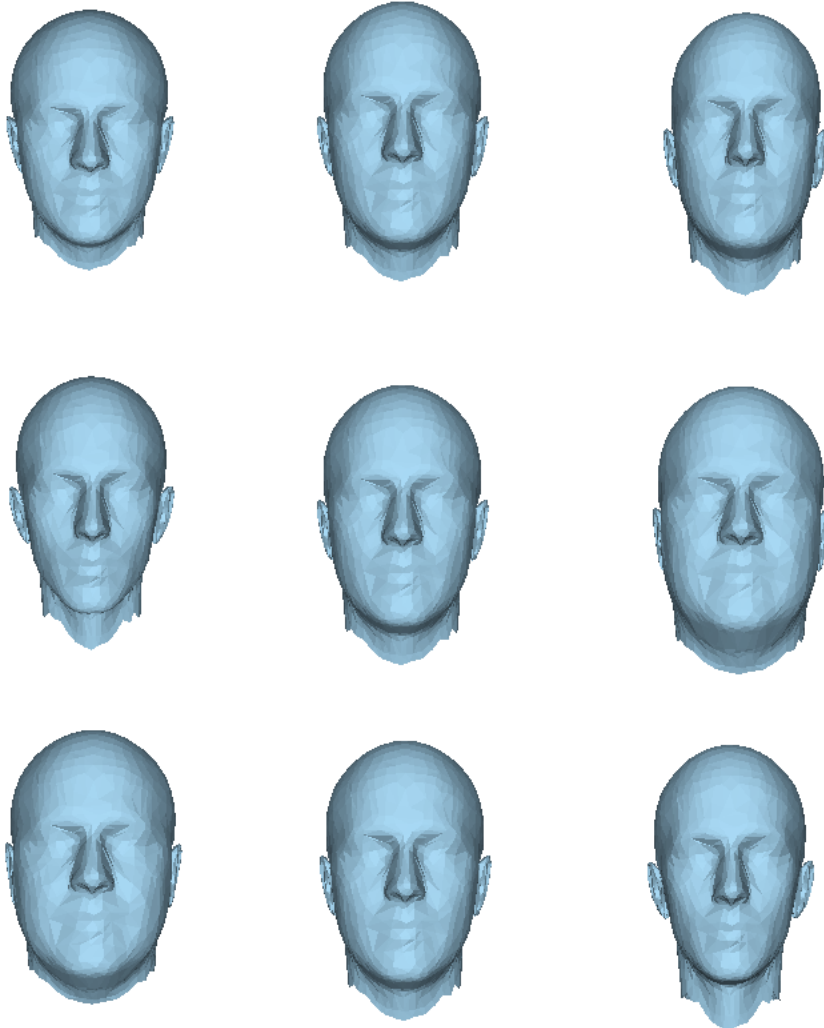


FIGURE 9. Intrinsic shape part deformation. An example of the head PCA model is shown (first three principal components from top to bottom). The center mesh in each row is the mean shape and the left and right meshes correspond to ± 3 standard deviations from the mean.

which is a weighted sum of squared distances between the interface points of the adjacent parts. Assume there are N_{ij} interface points between part i and part j . Let $I_{ij}(k)$ and $I_{ji}(k)$ denote the index of the k -th corresponding interface point shared by part i and part j respectively. $\tilde{\mathbf{p}}_i(\mathbf{x})$ indicates the part points in the global frame after applying parameters \mathbf{x} for each part. $u_{ij}(k)$ defines a stitching weight that is

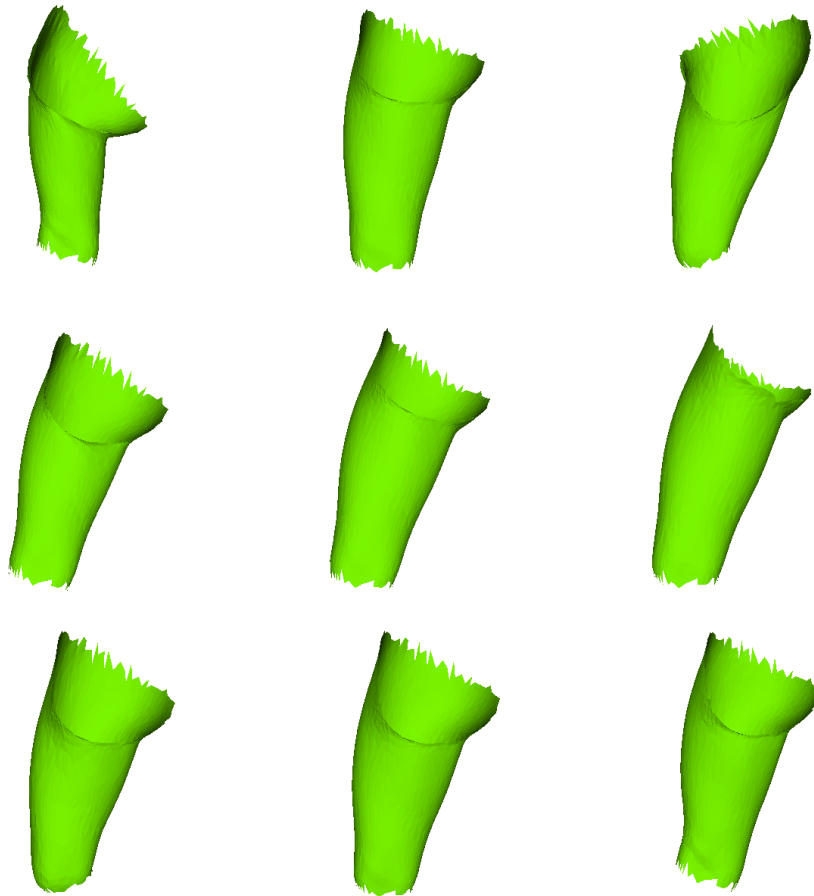


FIGURE 10. Pose-dependent part deformation. An example of the upper left leg PCA model is shown (first three principal components from top to bottom). The center mesh in each row is the mean shape and the left and right meshes correspond to ± 3 standard deviations from the mean.

set to 0.8 for points that are allowed to stretch more, like the front of the knee or the back of the elbow, and is set at 1.0 otherwise.

4. Generating Model Instances

It is useful to propose bodies during inference and for this we use a simple proposal process which is illustrated in Fig. 3. We define multivariate Gaussian distributions

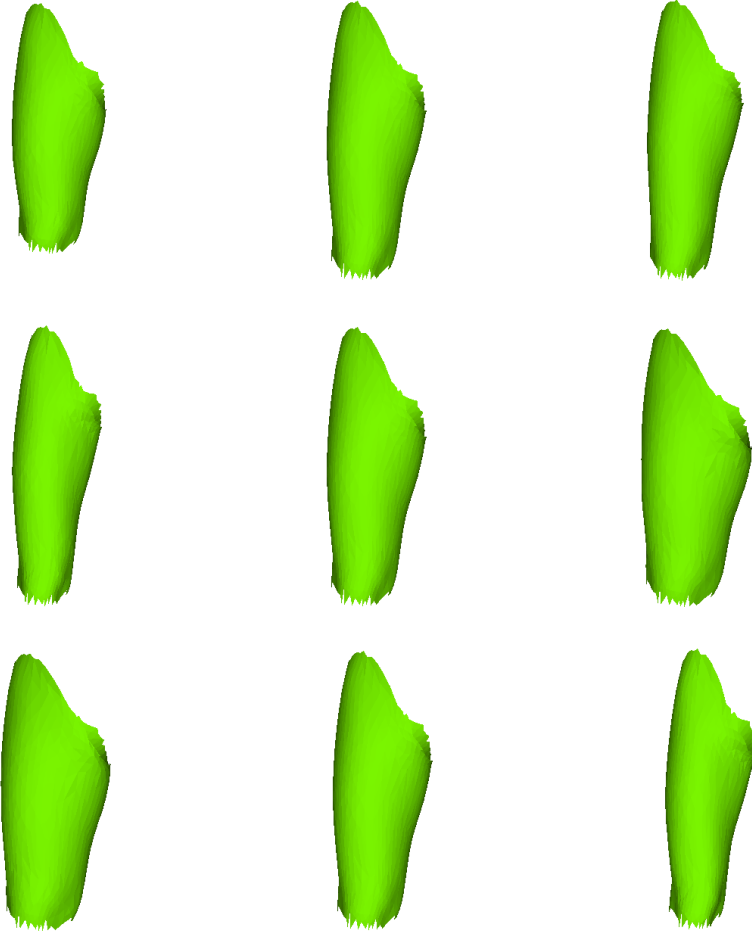


FIGURE 11. Intrinsic shape part deformation. An example of the upper left leg PCA model is shown (first three principal components from top to bottom). The center mesh in each row is the mean shape and the left and right meshes correspond to ± 3 standard deviations from the mean.

over the pose deformation variables and relative rotations of neighboring parts:

$$(49) \quad p_{ij}(\mathbf{r}_{ij}, \mathbf{d}_i, \mathbf{d}_j) = \mathcal{N}(\mathbf{r}_{ij}, \mathbf{d}_i, \mathbf{d}_j; \mu_{ij}, \Sigma_{ij}),$$

where \mathbf{r}_{ij} is the relative rotation of part j with respect to part i , and \mathbf{d}_i and \mathbf{d}_j are PCA coefficients for the pose deformation models of part i and j , respectively. We learn these functions from the training set with pose variations, for each combination i, j of connected parts.

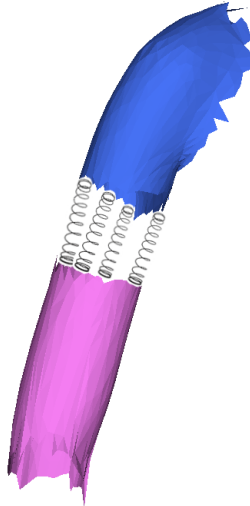


FIGURE 12. Stitching parts illustration. Parts can be thought of as being connected by springs between the interface points. When the model fits together seamlessly, this stitching cost is zero. During inference, the parts can move apart to fit data and then the inference method tries to infer a consistent model.

In order to generate an instance of SP, we first sample a vector of intrinsic shape variables \mathbf{s}_i (we sample from a Gaussian distribution over the PCA coefficients given the variance estimated when learning the PCA model). The intrinsic shape variables are replicated for each node, and generate body parts for the template mesh with the desired intrinsic shape (Figure 3(b)). We then sample a vector of pose deformation variables for the torso. These define the pose of the torso: given in SP the torso part also includes the pelvis, poses with the torso bent or twisted with respect to the pelvis are modeled as pose deformations (Figure 5). We then assign a global rotation and generate the torso mesh in the global frame. Recursively in the tree, starting at torso, for each node i : we get the pose-dependent deformation variables of the parent, $\mathbf{d}_{pa(i)}$; we condition the pairwise Gaussian $p_{pa(i)i}$ with $\mathbf{d}_{pa(i)}$, and marginalize the relative rotation vector $\mathbf{r}_{pa(i)i}$. This gives a Gaussian distribution over \mathbf{d}_i ; we sample this conditional distribution to get part deformations, and generate the part mesh in the local frame. The effect of the part deformations applied to each body part



FIGURE 13. Example of SP bodies. Several bodies generated using the SP model. Note the realism of the 3D shapes.

is shown in Figure 3(c). We finally compute the rotation and translation that stitches the parts together at their interface (Figure 3(d,e)) using the orthogonal Procrustes algorithm. Figure 13 shows samples of bodies generated using this procedure. Note that the described procedure does not prevent inter penetration of the parts. During inference, when we generate samples, we add noise to the part locations, creating disconnected bodies. This is described in the next chapter.

3D Mesh Alignment with the Stitched Puppet Model

We use SP to infer the pose and shape of people in the FAUST dataset [12]. FAUST contains high resolution 3D scans of people with different body shapes in a wide variety of poses. The goal is to align all the body scans so that points on each scan are in full correspondence with the other scans. The pose variation, noise, self contact, and missing data make the dataset very challenging.

The FAUST dataset is associated to a *challenge*, which requires participants to align 3D scans of the same person in different poses (*intra-subject* challenge), and of different people in different poses (*inter-subject* challenge). Figure 1 shows 20 scans of one of the 10 subjects in the dataset.

Our approach is model-based. In the literature many model-free approaches have been proposed to align generic meshes, but here, in particular for the inter-subject challenge, a model-based approach seems to be a better approach, as we know all

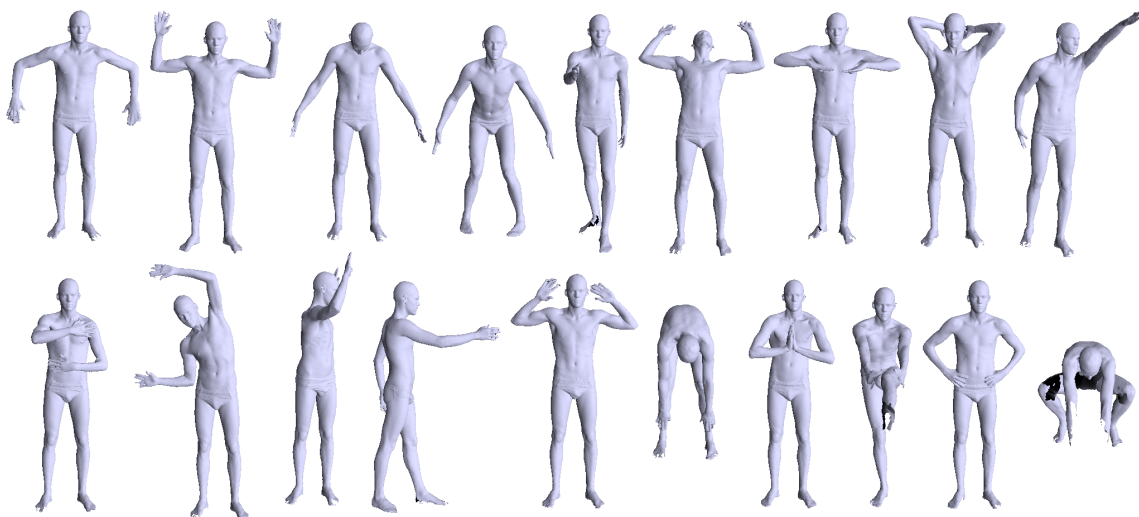


FIGURE 1. FAUST test scans for one subject. Goal of the *intra-subject* challenge is to bring pairs of these scans into correspondence.

scans are human subjects. We align the SP model to each scan, estimating pose and intrinsic shape. We can then create point-point correspondences between scans and the SP model, and based on these, create point-point correspondences across scans, as required from the challenge.

We face a complex inference problem: the state space for each part of SP includes continuous random variables representing the part shape coefficients and the 3D pose of the part and cannot be easily discretized. This is similar to the previous 3D part-based models [90] that use inference with continuous random variables. To deal with this, we exploit the D-PMP (Diverse Particle Max-Product) algorithm [67], which is suited to solve our optimization problem for the SP distributed model with continuous variables in high dimensional spaces. As illustrated in [67] and in Chapter 4, we have already applied the method to a similar inference problem of 2D human pose estimation.

1. Method

Consider the task of aligning SP to a 3D mesh S . We optimize the following energy:

$$(50) \quad E(\mathbf{x}, S) = E_{\text{stitch}}(\mathbf{x}) + E_{\text{data}}(\mathbf{x}, S),$$

where $\mathbf{x} = [\mathbf{x}_0, \dots, \mathbf{x}_{15}]$ are the model’s variables. The energy is the sum of a stitching term and a data term. The stitching term is the cost for parts to be disconnected, plus a penalty for penetration:

$$(51) \quad E_{\text{stitch}}(\mathbf{x}) = \sum_{i=0..15} \sum_{j \in \Gamma(i)} \alpha_{ij} (-\Psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) + Q_{ij}(\mathbf{x}_i, \mathbf{x}_j)),$$

where $\Gamma(i)$ is the set of neighbors of part i , α_{ij} is a weight for the joint between part i and part j , $\Psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ is the stitching potential (Equation 48), and Q_{ij} is a penalty for parts that intersect. We use a simple test on the location of the part centers, setting Q_{ij} to zero if $\|\mathbf{o}_i - \mathbf{o}_j\|_2$ is more than 0.05, and 1.0 otherwise. This has the effect of avoiding parts of similar length and shape, like upper and lower arms, to overlap in 3D. More sophisticated penalty terms could be also used. The node variables are

$\mathbf{x}_i = [\mathbf{o}_i^T, \mathbf{r}_i^T, \mathbf{d}_i^T, \mathbf{s}_i^T]^T$, and include the part center, the part rotation, expressed as a Rodrigues vector, pose-dependent deformations expressed as PCA coefficients, and the intrinsic shape deformations, also expressed as PCA coefficients.

Data term. The data term varies depending on the problem and here we consider the problem of fitting SP to 3D data. Specifically we consider fitting it to high-resolution scan data, which has the characteristic of being relatively noise-free, with no outliers, while missing data is possible.

We define a matching cost between the 3D model and the 3D data as the distance between model vertices and data vertices, S , plus a penalty for a mismatch in the normal direction. We want indeed to penalise cases where there is a small distance between model and scan, but very different orientation of the corresponding surfaces:

$$(52) \quad E_{\text{data}}(\mathbf{x}) = \sum_{i=0..15} \beta_i (D_i(\mathbf{x}_i, S) + R(\mathbf{x}_i, S)),$$

where β_i is a weight for part i , and $D_i(\mathbf{x}_i, S)$ is a model-to-data distance term given as:

$$(53) \quad D_i(\mathbf{x}_i, S) = \frac{1}{N_i} \sum_{k=1..N_i} (d_{i,k}(S)^2 + b)^\gamma$$

where

$$(54) \quad d_{i,k}(S) = \min_{\mathbf{v}_s \in S} \|\tilde{\mathbf{p}}_{i,k}(\mathbf{x}_i) - \mathbf{v}_s\|_2$$

is the distance from the model's point $\tilde{\mathbf{p}}_{i,k}(\mathbf{x}_i)$ to the data. The parameters are set as $b = 0.001$ and $\gamma = 0.45$. The penalty for normals mismatch penalizes cases where the normal at the model's point and the normal at the closest data point is larger than a threshold, $R_{ij} = \eta \sum_{k=1..N_i} \mathbb{I}(\theta_{i,k} > \frac{3}{4}\pi)$. Here \mathbb{I} is the indicator function, $\theta_{i,k}$ is the angle between the normal at $\tilde{\mathbf{p}}_{i,k}$ and the normal at $\mathbf{v}_{i,k}$, where $\mathbf{v}_{i,k}$ is the minimizer for $d_{i,k}(S)$ and $\eta = 0.005$.

Optimization. To minimize the energy we use the D-PMP algorithm [67]. As mentioned in Chapter 4, D-PMP is a particle-based method for MAP estimation in graphical models with pairwise potentials. In contrast to Markov Chain Monte Carlo (MCMC) methods, where particles represent distributions, in D-PMP particles are

used to represent locations of modes of the posterior. This implies that even if the model is very high dimensional, it is not necessary to use a very large number of particles. D-PMP is an iterative method where belief propagation (BP) is applied at each iteration over the set of particles. At each iteration, particles are resampled with the aim of creating new particles in good locations. A key component of the algorithm is the selection step. During resampling, the number of particles is doubled in order to place particles in new locations without removing any of the current ones. Then, a selection step based on preserving the BP messages is applied. During resampling, different strategies can be considered. Typically new particles are created by sampling the prior, with random walks from the current particles, or exploiting data-driven proposals.

We initialize particles by generating SP sample bodies with mean intrinsic shape. Each particle represents a body part, and is a vector of node variables. To place the samples in global frame, we set the position of the torso at the origin, in an upright posture but with random orientation about the vertical axis. To provide a rough alignment with the input 3D data, we also align it to the origin of the global frame by computing the mean of the input data points. We add a small random noise to the location of each particle, obtaining disconnected sets of body parts. If we could assume the subject is always in the same global orientation (for example was always facing the same camera in the scanner), in the particles initialization we could assume only a unique view, and this would help the inference as left and right body parts would be always on the correct side, without ambiguities for poses that differ for 180 degrees in torso rotation. However, in the FAUST test set some of the subjects have a different orientation with respect to most of the examples, thus we cannot assume a preferred global torso orientation. FAUST contains only static poses, therefore subjects are never upside down, and thus we assume an upright posture.

As an estimate of the global translation of the scan, we take the average value of the scan points. We make this point coincide with the centre of the torso of the SP models that we generate for initialization. Note that this simple assumption creates

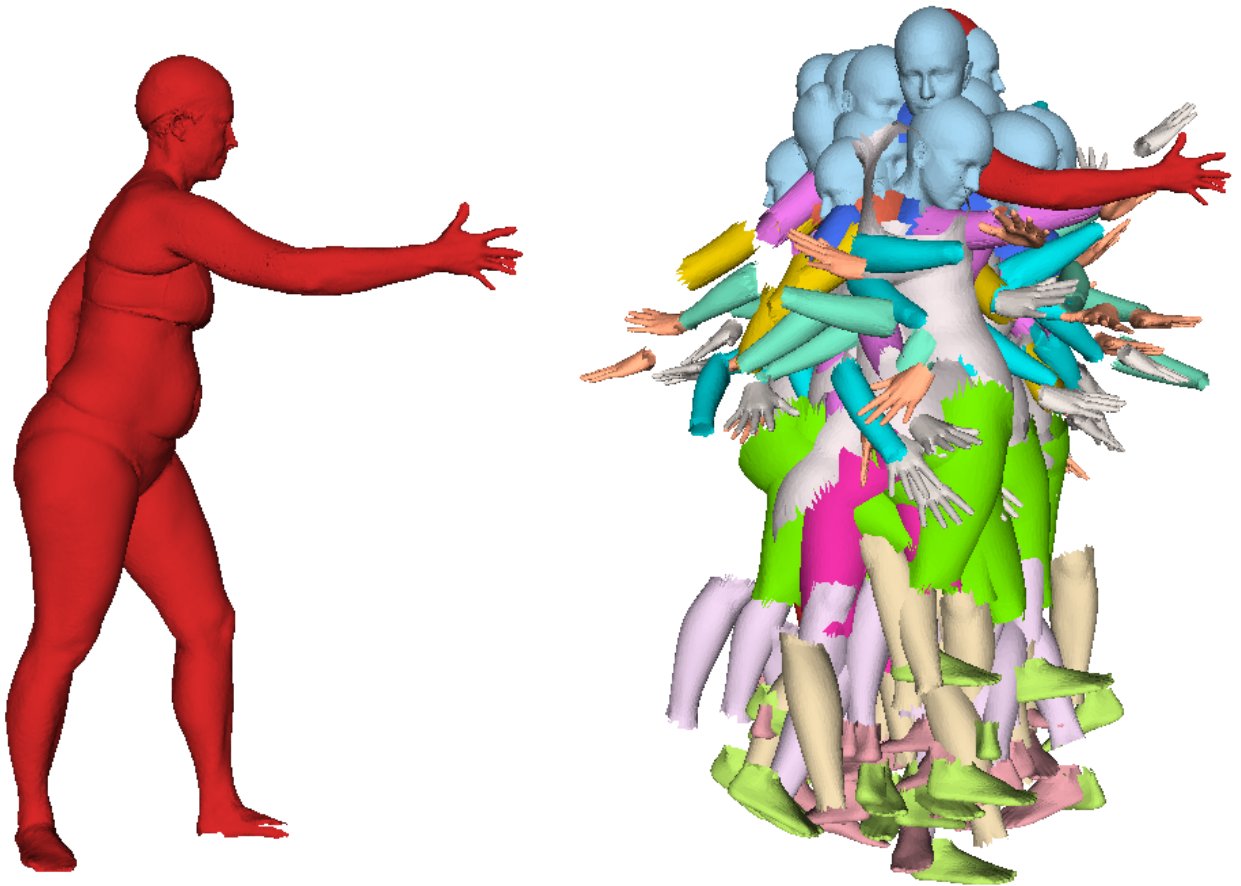


FIGURE 2. Example of particles initialization: note that particles are initialized from disconnected SP models randomly sampled. The red arm on the right belongs to the FAUST scan around which the random puppets are generated.

a bias between model and scan, but the optimization algorithm can deal with large uncertainty in the particles locations, also due to the fact that the scan data does not contain outliers. Figure 2 shows the set of initial particles in an example where the optimization uses 30 particles.

During optimization, we use an adaptive scheme for assigning the weights α and β in the energy.

Given a budget of N iterations, we split them into 4 stages. At the beginning we set the weights in a way that lowers the influence of the distal parts (lower limbs, hands and feet), to which we assign small weight for the stitching and the data terms.

Namely, for the distal parts we set the weights for the stitching term α_i to 0.001 for the hands and feet interfaces. We set α_i to 0.5 for the interfaces at the hips: we verified that tighter hip interfaces help to prevent overlap between the legs. For the other interfaces we set α_i to 0.25. The weights for the data term, β_i , are set to 1.0 for all the parts, with the exception of the lower parts, hands and feet, for which we set β_i to 0.001.

In a second stage we increase these weights to bring in more influence from the distal parts. We set all the data term β_i to 1.0, with the exception of hands and feet that have β_i at 0.1. We set the stitching weights, α_i , to 0.25 for all the interfaces, with the exception of the hips, that are at 0.5.

In a third stage we set all the stitching weights, α_i , to 0.5. We also reduce the variance of the distribution we use to add random noise to the particles, in order to perform a more local exploration of the solution space.

At a final stage we apply a greedy PMP (Particle Max-Product) algorithm (also used in [67] for a comparison with D-PMP) where, at each iteration, all the particles are resampled with random noise around the current best location. This has the effect of refining the solution.

Resampling and refinement. At each iteration, for each node i in the graphical model (body part), and for each particle $\mathbf{x}_i^{(s)}$, we define a resampling strategy as follows. With probability 0.5 we resample a new particle $\hat{\mathbf{x}}_i^{(s)}$ with a random walk from $\mathbf{x}_i^{(s)}$. The sampling is performed over all the node variables or only over the pose deformation variables $\mathbf{d}_i^{(s)}$ with equal probability. Alternatively, we generate a new particle as a proposal from a neighbor node. First, we select a neighbor j for the node i , then a random particle from node j , $\mathbf{x}_j^{(t)}$. We use $\mathbf{x}_j^{(t)}$ to condition the pairwise Gaussian distribution between node j and node i , $p_{ji}(\mathbf{r}_{ji}, \mathbf{d}_j, \mathbf{d}_i)$ (Equation 49). With probability 0.5 the conditioning variables are the pose deformation variables $\mathbf{d}_j^{(t)}$, otherwise we also condition the pairwise Gaussian with a random relative angle uniformly sampled within joints limits. We sample pose deformation variables from the conditional Gaussian, and obtain $\hat{\mathbf{d}}_i^{(s)}$. We then set the intrinsic shape parameters

$\hat{\mathbf{s}}_i^{(s)} = \mathbf{s}_j^{(t)}$. The location and orientation are computed as those that stitch the mesh of $\hat{\mathbf{x}}_i^{(s)}$ to the mesh of the neighbor’s particle $\mathbf{x}_j^{(t)}$. After each particle is resampled, we run a few steps of Levenberg-Marquardt (LM) optimization over the location, rotation and pose-deformation parameters to locally improve alignment to the scan data. Since this local optimization is applied only to the generated particle, it has the effect of making it disconnected from its neighbor.

Local optimization with the Levenberg-Marquardt method. The local gradient-based optimization brings the part closer to the closest points in the 3D scan. Let $\tilde{\mathbf{p}}$ be the points of a part in the global frame and \mathbf{s} the set of points on the scan closest to $\tilde{\mathbf{p}}$.

Here we use a KD-tree to quickly determine the closest points on the scan to the model, and we use only a subset of the part points. In the Levenberg-Marquardt method, one performs a gradient-based optimization to reduce the residuals between the model prediction and the data, where the cost function is a sum of squared residuals, and the residuals are differences between model points and data. The points have a non-linear relationship with the variables being optimized, and the method requires the computation of the first derivatives of the model points with respect to the problem variables. Here we optimize over rotation, translation and pose-dependent shape deformations. We indicate the optimization variables as $\mathbf{q} = [\mathbf{r}^T, \mathbf{c}^T, \mathbf{d}^T]^T$.

Let $\mathbf{e}(\mathbf{q}) = \tilde{\mathbf{p}}(\mathbf{q}) - \mathbf{s}$ be a matrix of $(N, 3)$ residuals, that we indicate also as $\mathbf{e} = [e_x^T, e_y^T, e_z^T]$. The cost function is then:

$$(55) \quad E(\mathbf{q}) = \sum_{j=[0..(N-1)]} \sum_{k=[x,y,z]} \mathbf{e}_{j,k}(\mathbf{q})^2.$$

Let J be the Jacobian matrix, where each element is a partial first derivative of the cost function with respect to each of the optimization variables. We write $J = [J_{\mathbf{r}_0}^T, J_{\mathbf{r}_1}^T, J_{\mathbf{r}_2}^T, J_{\mathbf{c}_x}^T, J_{\mathbf{c}_y}^T, J_{\mathbf{c}_z}^T, J_{\mathbf{d}_0}^T, J_{\mathbf{d}_1}^T]^T$. Here we write the equations for 2 pose-dependent deformations variables, but we use 12 for the torso and 5 for the other parts. Let J_r be also a matrix of partial first derivatives of the elements of the Rodrigues vector with respect to each of the element of the corresponding rotation matrix. J_r has

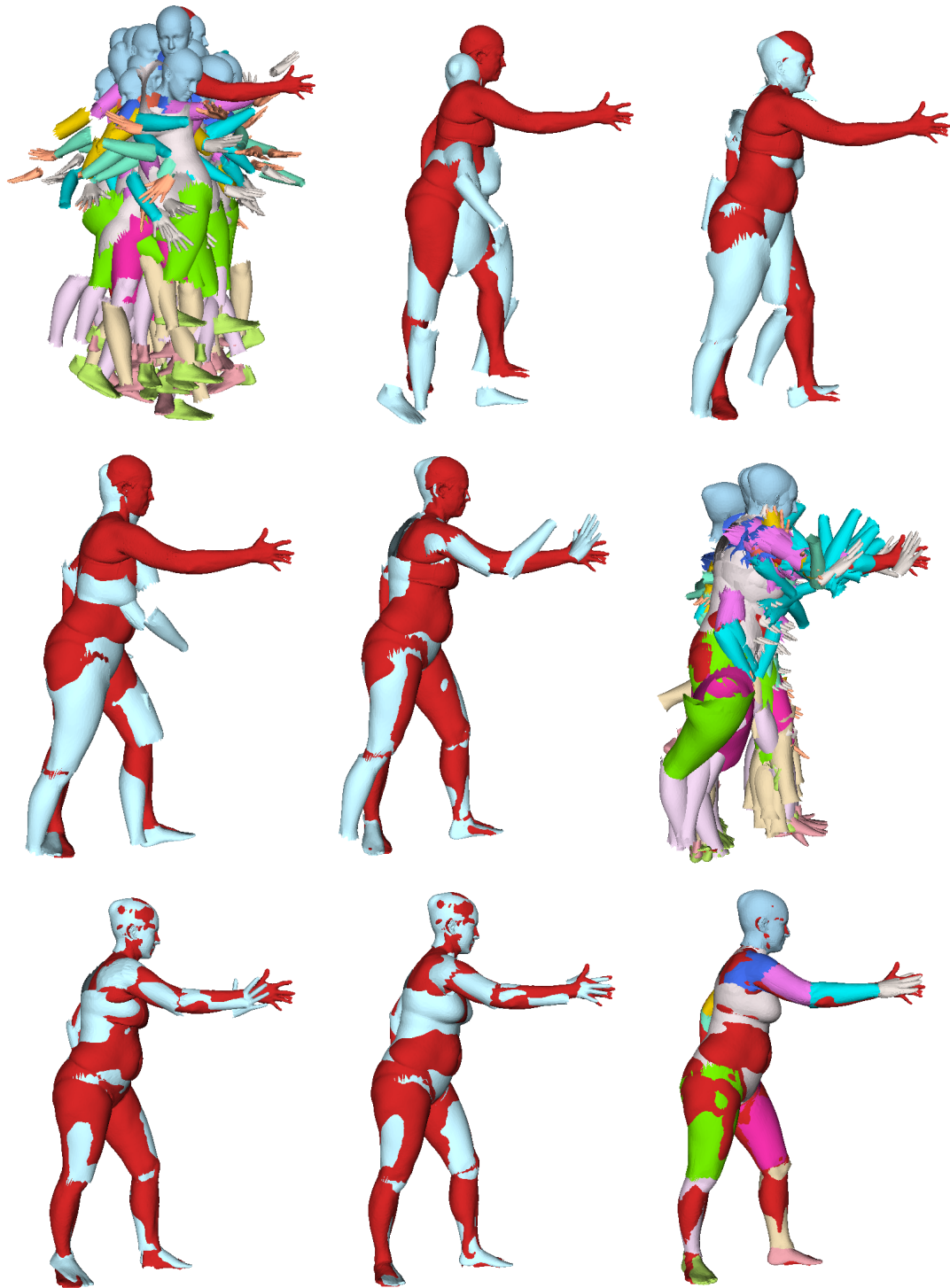


FIGURE 3. D-PMP optimization. Inference with 30 particles for 60 iterations. From top to bottom, left to right: initial particles; scan (red) and current solution (light blue) at various steps; the final set of particles. At the end a greedy algorithm resamples all the particles around the current solution.

dimension (3, 9), and the element $J_r(h, l)$ is the partial derivative of \mathbf{r}_h with respect to the l -th element of the rotation matrix, indexed as a 9-element vector.

The elements of the Jacobian are:

$$\begin{aligned}
J_{\mathbf{r}_0} &= 2 e_x (J_{r_{0,0}} \tilde{\mathbf{p}}_x + J_{r_{0,1}} \tilde{\mathbf{p}}_y + J_{r_{0,2}} \tilde{\mathbf{p}}_z) + \\
& 2 e_y (J_{r_{0,3}} \tilde{\mathbf{p}}_x + J_{r_{0,4}} \tilde{\mathbf{p}}_y + J_{r_{0,5}} \tilde{\mathbf{p}}_z) + \\
(56) \quad & 2 e_z (J_{r_{0,6}} \tilde{\mathbf{p}}_x + J_{r_{0,7}} \tilde{\mathbf{p}}_y + J_{r_{0,8}} \tilde{\mathbf{p}}_z)
\end{aligned}$$

$$\begin{aligned}
J_{\mathbf{r}_1} &= 2 e_x (J_{r_{1,0}} \tilde{\mathbf{p}}_x + J_{r_{1,1}} \tilde{\mathbf{p}}_y + J_{r_{1,2}} \tilde{\mathbf{p}}_z) + \\
& 2 e_y (J_{r_{1,3}} \tilde{\mathbf{p}}_x + J_{r_{1,4}} \tilde{\mathbf{p}}_y + J_{r_{1,5}} \tilde{\mathbf{p}}_z) + \\
(57) \quad & 2 e_z (J_{r_{1,6}} \tilde{\mathbf{p}}_x + J_{r_{1,7}} \tilde{\mathbf{p}}_y + J_{r_{1,8}} \tilde{\mathbf{p}}_z)
\end{aligned}$$

$$\begin{aligned}
J_{\mathbf{r}_2} &= 2 e_x (J_{r_{2,0}} \tilde{\mathbf{p}}_x + J_{r_{2,1}} \tilde{\mathbf{p}}_y + J_{r_{2,2}} \tilde{\mathbf{p}}_z) + \\
& 2 e_y (J_{r_{2,3}} \tilde{\mathbf{p}}_x + J_{r_{2,4}} \tilde{\mathbf{p}}_y + J_{r_{2,5}} \tilde{\mathbf{p}}_z) + \\
(58) \quad & 2 e_z (J_{r_{2,6}} \tilde{\mathbf{p}}_x + J_{r_{2,7}} \tilde{\mathbf{p}}_y + J_{r_{2,8}} \tilde{\mathbf{p}}_z)
\end{aligned}$$

$$(59) \quad J_{\mathbf{c}_x} = 2 e_x$$

$$(60) \quad J_{\mathbf{c}_y} = 2 e_y$$

$$(61) \quad J_{\mathbf{c}_z} = 2 e_z$$

$$(62) \quad J_{\mathbf{d}_0} = 2 e_x \tilde{A}_{x,0} + 2 e_y \tilde{A}_{y,0} + 2 e_z \tilde{A}_{z,0}$$

$$(63) \quad J_{\mathbf{d}_1} = 2 e_x \tilde{A}_{x,1} + 2 e_y \tilde{A}_{y,1} + 2 e_z \tilde{A}_{z,1}$$

where to compute $\tilde{A}_{x,l}, \tilde{A}_{y,l}, \tilde{A}_{z,l}$ we first compute $A = B_{\mathbf{p}}\mathbf{d}$, a vector of size (3, N), with N the number of contour points. This is a vector of the components of the coordinates of part points in the local frame that depend on \mathbf{d} . We then rotate these coordinates into the global frame and obtain \tilde{A} , from which we extract matrices of size ($N, 2$) for each of the three spatial coordinates (here we are considering 2 pose-dependent deformation variables).

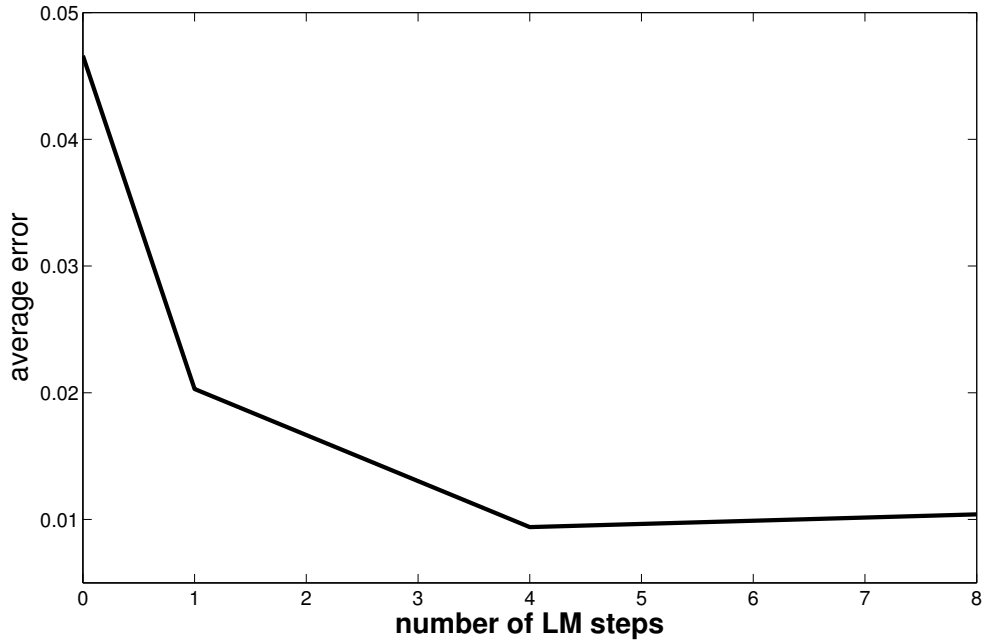


FIGURE 4. Local Levenberg-Marquardt (LM) optimisation. Plot of the average alignment error over a subset of the FAUST training set for different values of the steps in the local LM optimization.

Given the Jacobian, we can compute an increment for the current value of \mathbf{q} as;

$$(64) \quad \Delta \mathbf{q} = -(J J^T + \lambda I)^{-1} J e,$$

where e is a column vector of dimension N with elements $e_j = \sum_{k=\{x,y,z\}} \mathbf{e}_{j,k}(\mathbf{q})^2$. The LM method computes a variable update at each iteration. If with the new value of the variables the cost function decreases, then the move is accepted and the value of λ is decreased. Otherwise the move is rejected and the value of λ is increased. The initial value of λ is 0.1. Here we run the method for 4 iterations, and we increase/decrease the value of λ by multiplying or dividing by 10. Note that the cost function we minimize is not the same as the data term. The local LM optimization is very effective for better performance. Figure 4 shows the average alignment error over a subset of the FAUST training set for different numbers of LM iterations.

2. Experiments

We align the SP model to each of the test scan in the FAUST dataset, applying the iterative particle-based optimization method described in the previous section. We first run the optimization with 80 particles per node for 120 iterations, obtaining an average error of about 2 cm for the intra-challenge, with a running time of about 30 min. To improve accuracy, we run our method with 160 particles for 240 iterations, for 3 different random initializations, among which we select the result with lowest energy. For each pair of scans in the intra-subject and inter-subject FAUST challenges, we need to provide an association between scan points. We compute this mapping with a procedure that estimates the closest point from the scan to the model, effectively exploiting the model as a "bridge" to connect points in the two scans. These point-point associations are then submitted to the FAUST website (<http://faust.is.tue.mpg.de/>) for evaluation.

Our final results are an average error of 1.57 cm for the intra-subject challenge and 3.13 cm for the inter-subject challenge. Figure 5 and 6 show the histograms of the errors for the intra-subject challenge and the inter-subject challenge, respectively. Figure 7 and 8 shows two cases with the worst errors, which are located where there are contact points, and scan vertices are assigned to the wrong body part in the model.

At this time, the method described is the first to our knowledge to compete on FAUST, so we have no comparison with other methods. The authors of the benchmark report average errors on the intra-subject test of 28 cm for MÖBius voting [59] and 12 cm for Blended Intrinsic Maps (BIM) [54], which are two model-free registration techniques. But these two methods did not return any results for 6 and 12 cases, respectively. No performance numbers are available for model-based methods, apart from the average errors for the method used to build the ground truth data without the appearance term for texture matching. These errors are 0.7 cm and 1.1 cm for the intra-subject and inter-subject tasks, respectively.

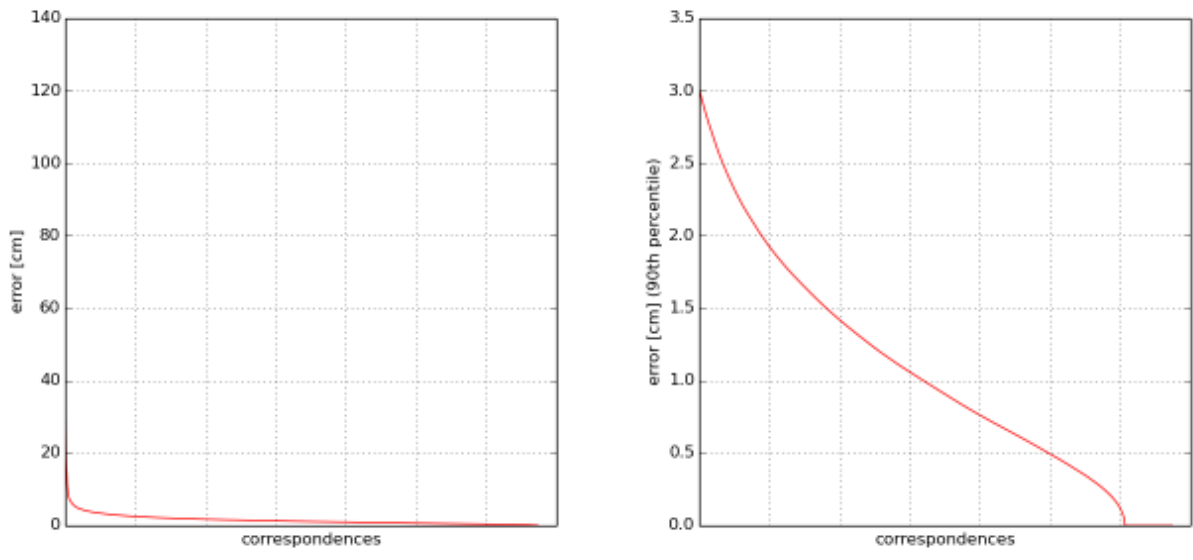


FIGURE 5. Intra-subject challenge. Histograms of the errors.

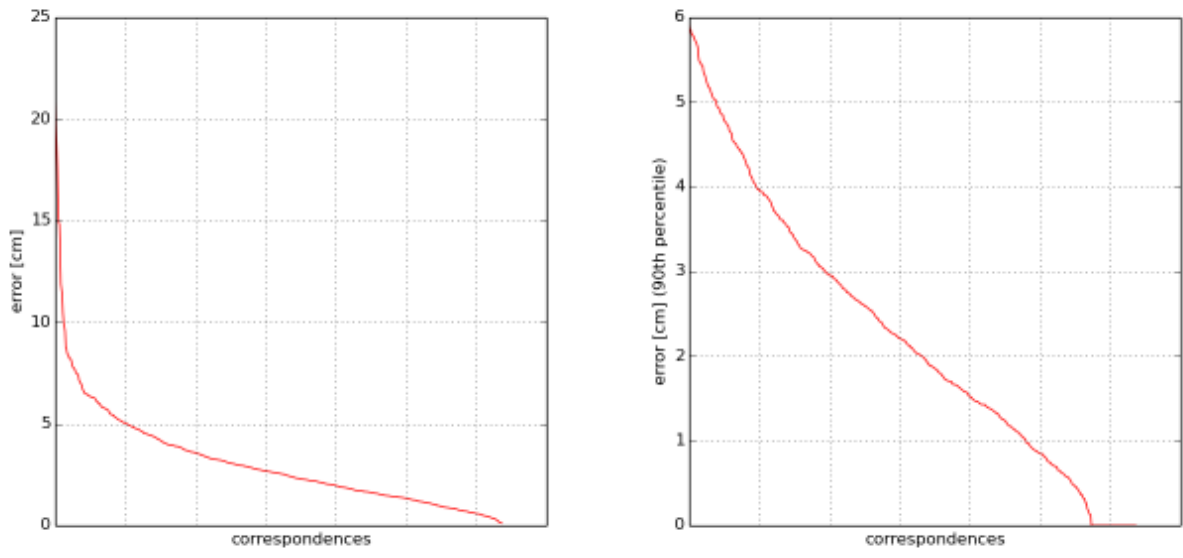


FIGURE 6. Inter-subject challenge. Histograms of the errors.

Figure 9 shows examples of problem cases. We found that the major source of error is when hands are in contact and vertices of one hand are assigned to the other hand (Figure 9(right)), creating large errors where the mesh to be aligned has hands

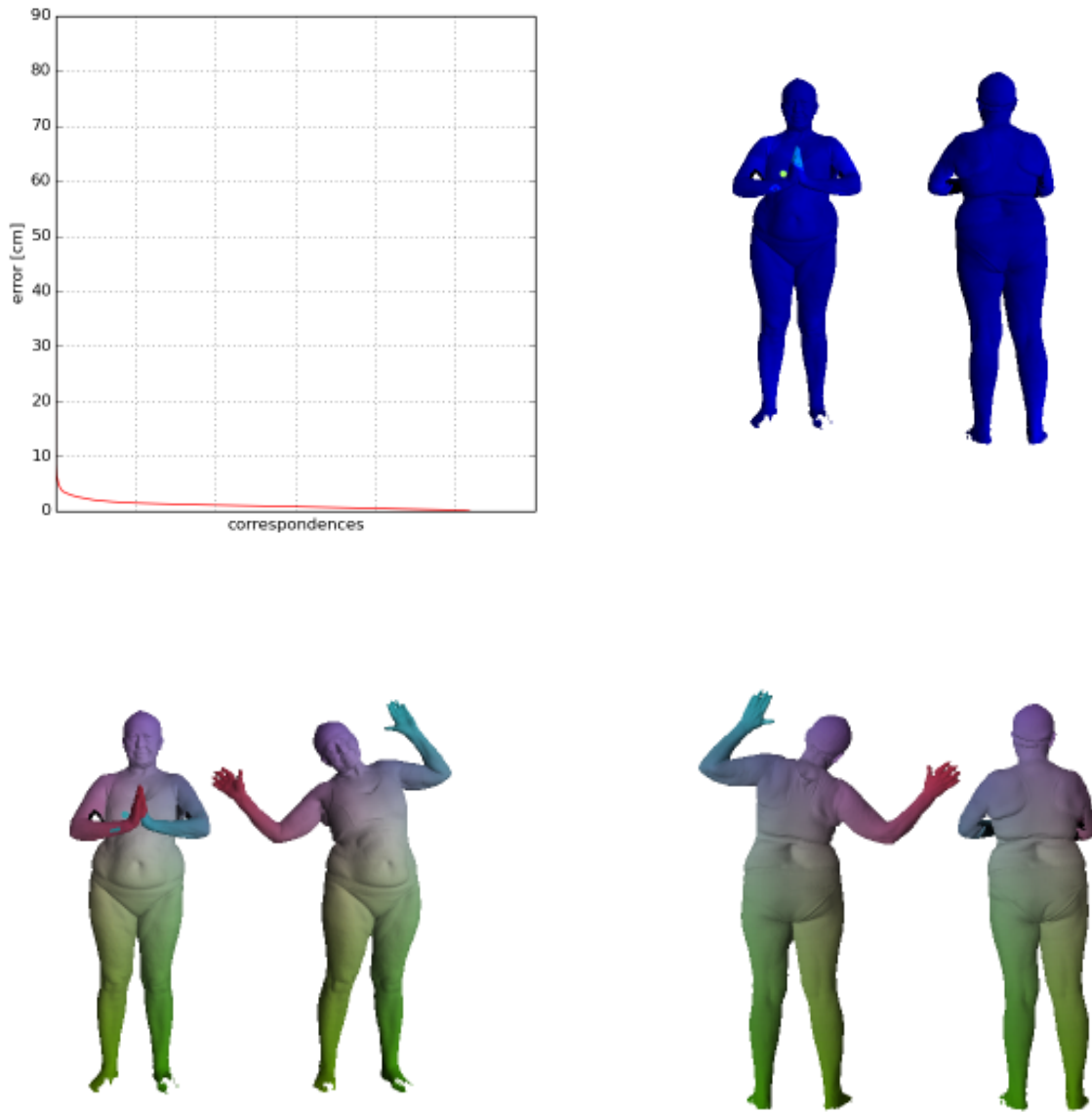


FIGURE 7. Intra-subject challenge. One of the worst results, which is due to a mistake in associating scan points to the model in case of contact points. Note in the figure at the bottom left how some points on the right lower arm have been associated with the left hand.

very far away (as illustrated in Figure 7). In a couple of cases we missed a lower arm (Figure 9(center)), here our simple test for interpenetration was not enough to prevent

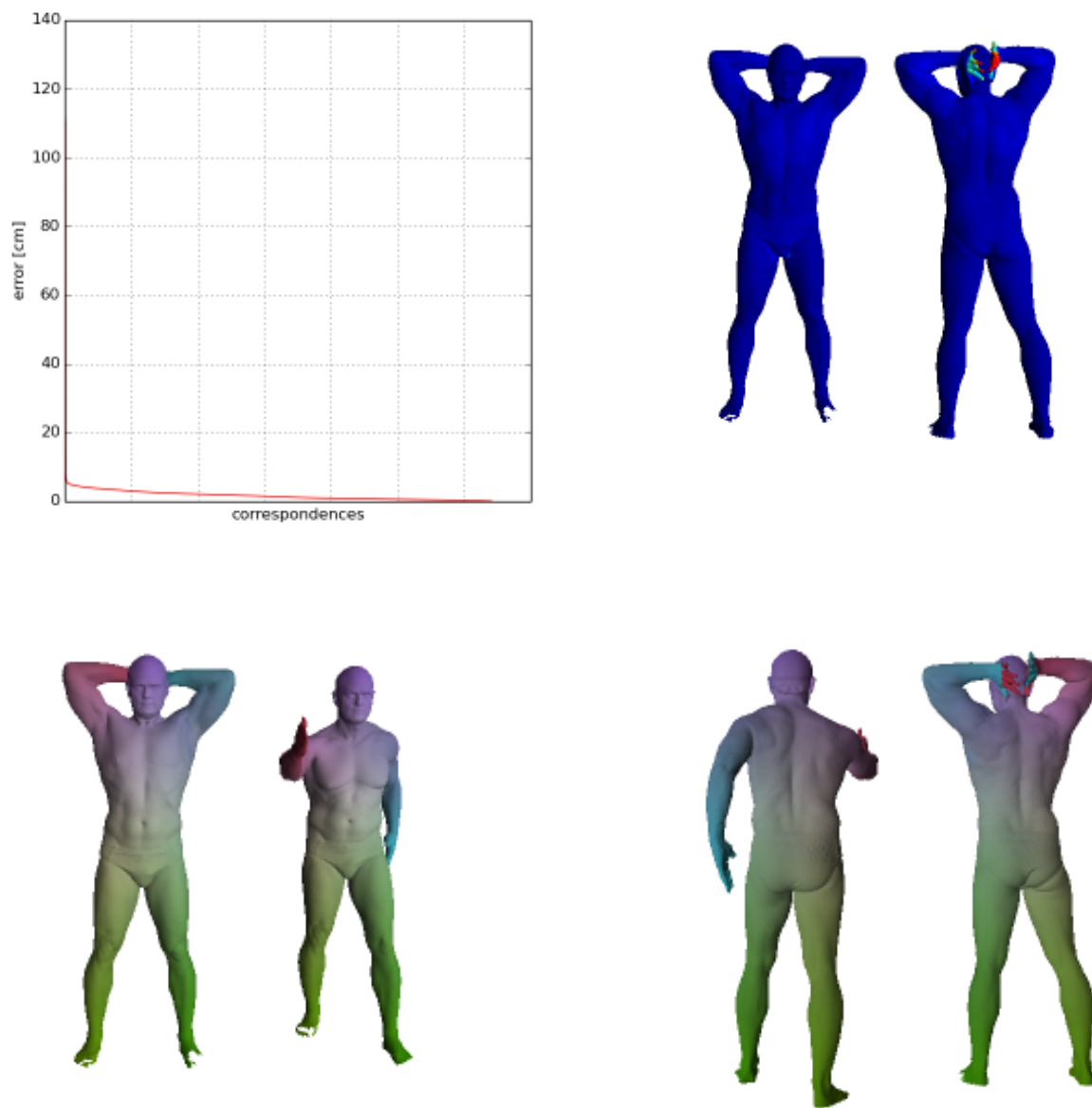


FIGURE 8. Intra-subject challenge. Another bad result, which is due to a wrong association between scan points and model vertexes when there are contact points. Note in the figure at the bottom right how the hands have different colors, illustrating the confusion between left and right hands.

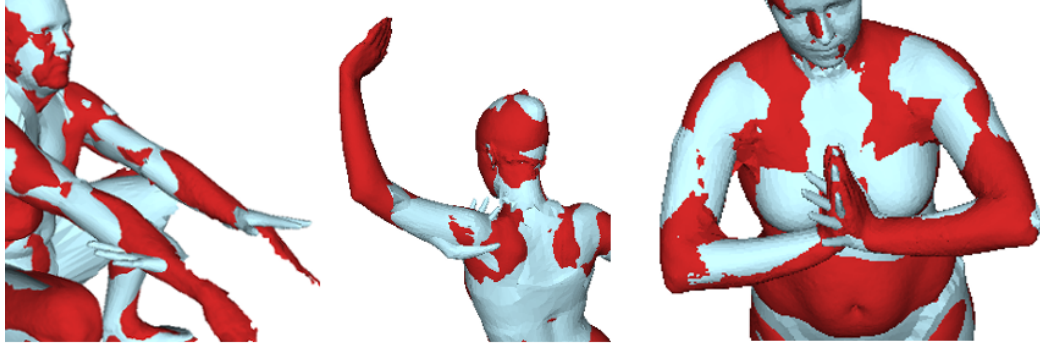


FIGURE 9. FAUST errors. Examples of mistakes in the alignment of the scan (red) to the model (light blue). (left) The hands are matched to the lower arms; (middle) the lower arm and hand are matched to the upper arm; (right) the hands are not accurately matched.

the error. Also, inaccuracies come from the scans with missing data (Figure 9(left)). The FAUST training set contains no examples with significant amounts of missing data, therefore the parameters in our energy could not be optimized to deal with such cases.

Figure 10 and 11 show examples of alignment: the scan is visualized in red and the model in light blue.

We show the results for the same pose for all the subjects in the test set to illustrate the quality of the shape estimation in Figure 12, where the scan is rendered in red and the model in rendered in light blue.



FIGURE 10. Alignment on FAUST. We show the test scan in red and SP in light blue.

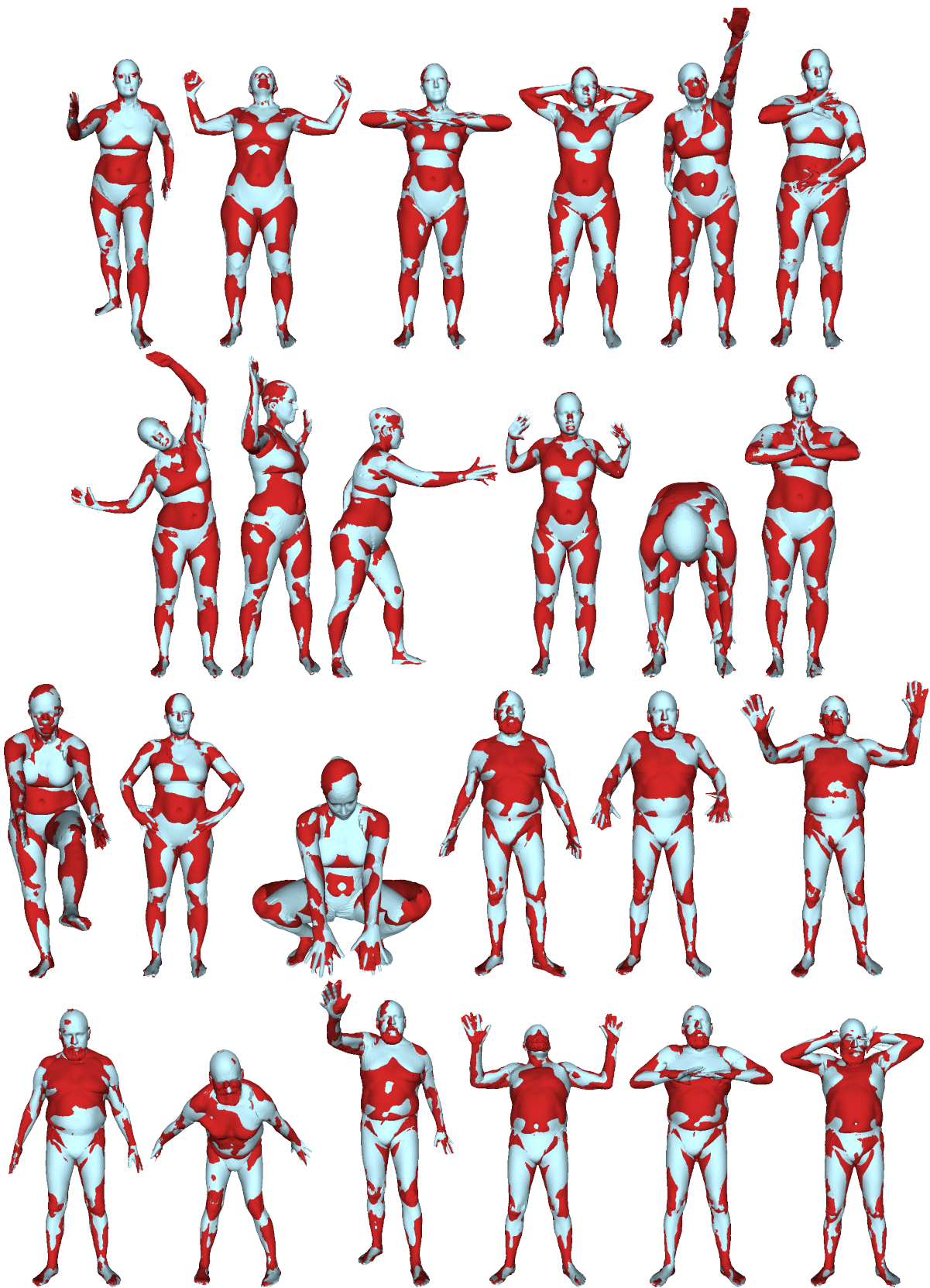


FIGURE 11. Alignment on FAUST. We show the test scan in red and SP in light blue.

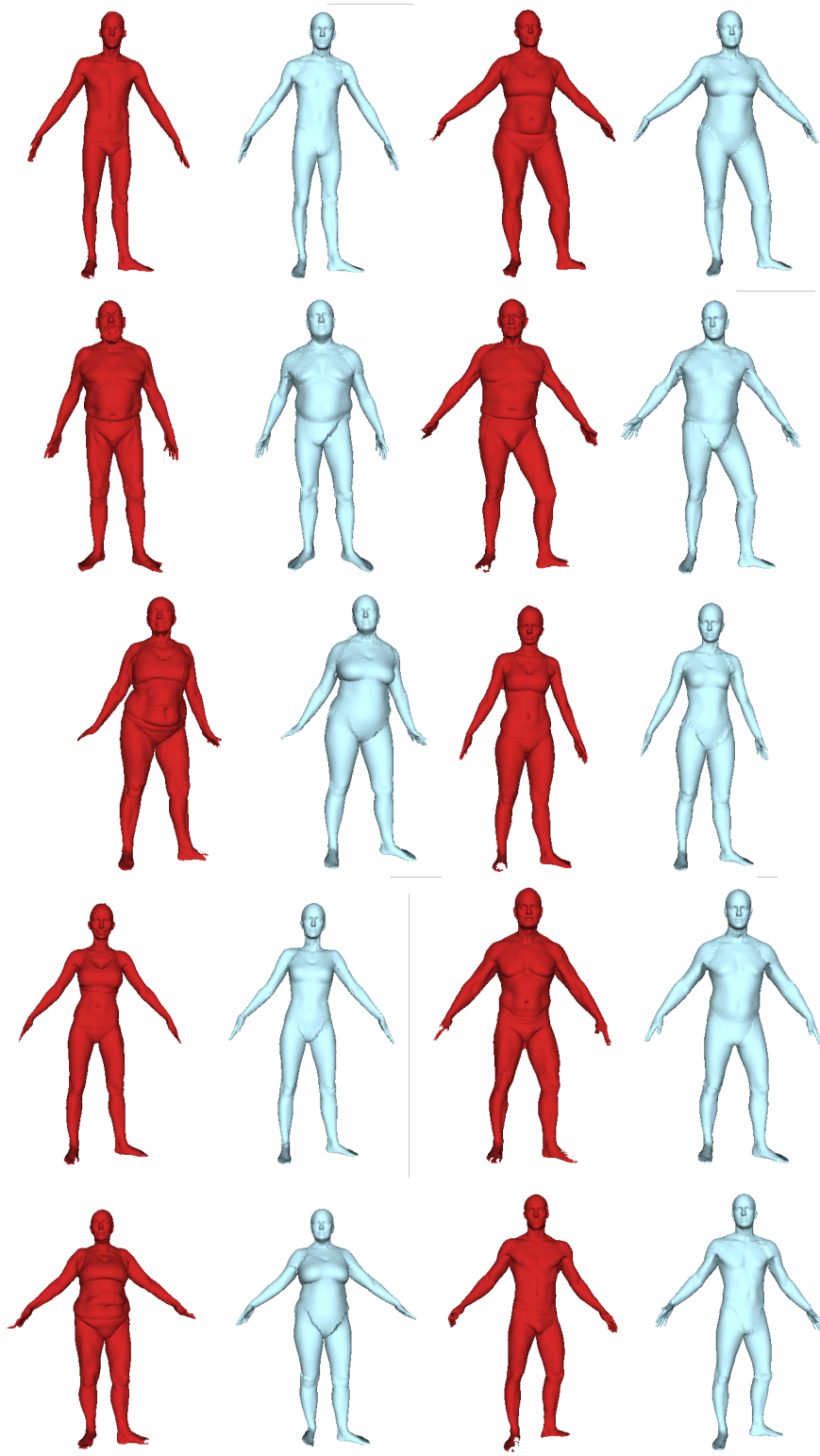


FIGURE 12. Alignment on FAUST. We show the scan (red) and model (light blue) for the different subjects in the test set to illustrate the quality in the estimation of the body shape.

3. Pose and Shape from Visual Hull

We perform a further experiment on a different type of 3D data to illustrate that our method can deal also with very noisy, approximate, 3D information. We consider a set of frames from the MPI08 dataset [74], from which we extracted the visual hull. We then aligned the SP model, independently on each frame. We used the same set of parameters we defined for the initial stage of the optimization for the previous experiment on FAUST, with the difference that we only perform the first stage of the optimization, as we found that our settings for the refinement stages were decreasing the quality of the solution. This is no surprise given the different quality of the input data, as in the refinement stages more weight is given to the model. In this case we also did not use the penalty for normals mismatch in the data term. We run our method with 200 particles for 120 iterations, for 3 different random initializations, independently for each frame. Figure 13 shows a subset of the results. We show the sequence up to the frame where our model performed correctly. After the last frame the actor turns upside down and our algorithm failed to align to the data, giving preference to a standing position. Figure 14 instead shows the last frames of the sequence, with failure cases.

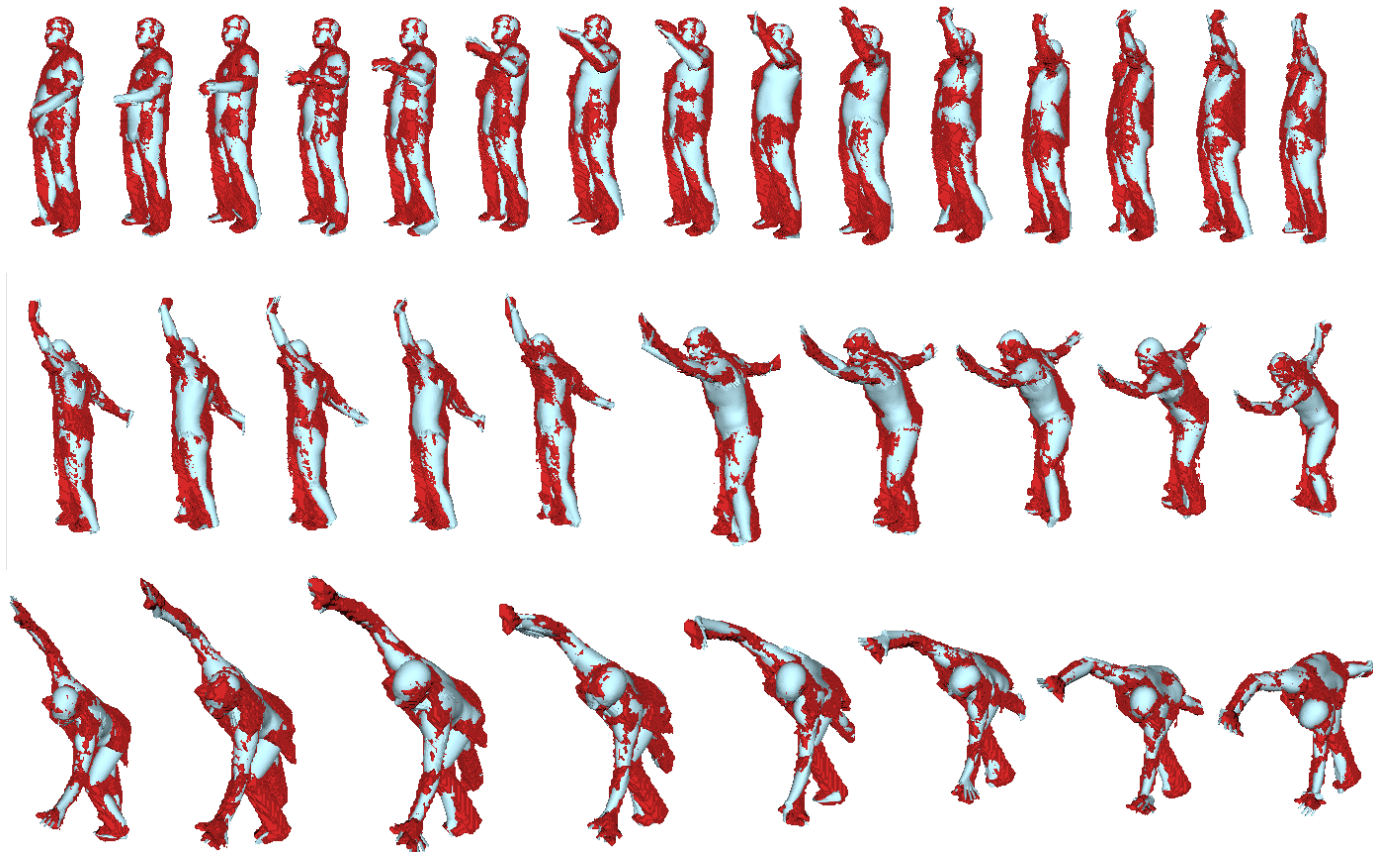


FIGURE 13. Alignment to visual hull data. We show the visual hull data in red and SP in light blue.

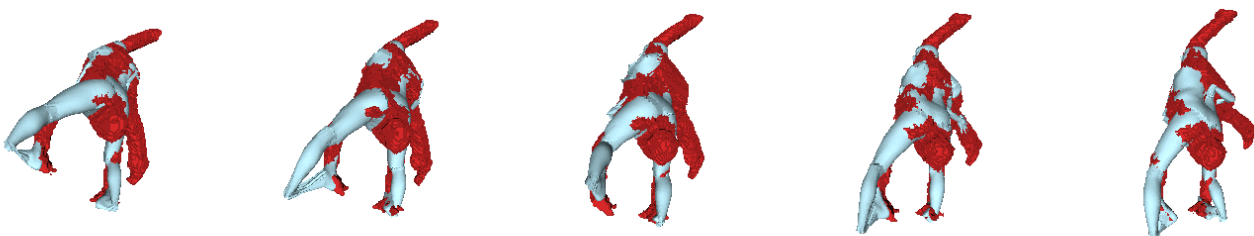


FIGURE 14. Alignment to visual hull data. Failure cases; here the subject is almost upside-down and the optimization cannot estimate a proper orientation for the body.

CHAPTER 9

Conclusion

In this thesis we developed novel models of the human body to solve problems like pose and shape estimation from images, videos and 3D data. In particular, we have focused our attention on part-based models that can support inference and optimization with distributed techniques based on message passing algorithms.

Models of the human body typically used in computer vision are very simple assemblies of geometric structures, like rectangles in 2D or cylinders in 3D. In computer graphics, models of the human body are mesh-based and very realistic, but computationally challenging to apply to computer vision problems.

In this thesis our aim was to fill the void between these two extremes by proposing models that can accurately represent body shape, in particular its variations with pose, and at the same time admit a tree-structured graphical model representation, where efficient algorithms based on belief propagation can be applied.

We encountered several issues in designing our models. First, we had to generate suitable training data. We followed the direction of generating the training data from another model, a very realistic representation learned from real data. This indirect approach gave us the advantage of a great control on the number and types of training data, and of an existing segmentation of the human body into parts.

Second, we had to define a discriminative learned likelihood to use our 2D model, Deformable Structures (DS), for pose estimation on uncontrolled images. DS is not a full model of human appearance in images, as it represents the contour of a minimally clothed body. In uncontrolled images people dress in various ways and can have voluminous hair, therefore a simple likelihood function based on contour matching was not a feasible choice. We exploited the successful application of HOG descriptors

for learning discriminative scoring functions for part detectors and introduced a novel contour-based image likelihood based on *steered* HOG descriptors.

Learning the likelihood involved challenges to define an easy-to-use annotation technique for generating many pairs of images with a DS model aligned to the depicted person. We designed a Web-based annotation tool where the DS puppet can be easily overlapped to images by dragging skeleton joints.

A further challenge came from inference. Adding shape to part-based models that are usually parameterized by pose, we increase the number of random variables considerably, and we cannot use the simple message passing methods used for discrete problems. We then applied a modified version of a particle-based Max-Product belief propagation method, that we integrated with novel model-based and data-driven proposals to resample particles.

We applied the 2D body model to the task of pose estimation from images and from video sequences. We followed the classical framework of Pictorial Structures in defining our 2D model and its application to the pose estimation problem from images, where the model plays the role of a prior over pose and is independent from data, while the data-dependent constraints come from the likelihood function. With DS we can model pose and pose-dependent shape deformations for a fixed intrinsic body shape.

Our approaches have produced state-of-the-art results at the time of their publications, but today the most successful methods for pose estimation are based on defining mixture models conditioned by image data, and on learning models or likelihood functions with deep convolutional neural networks that can also include context. This suggests that a local evaluation of image evidence is too noisy and unreliable, and pose models that are not conditioned by image data are too weak for consistent good solutions in a dataset with images that are very different from each other, with people of different appearance, pose, backgrounds, and scale. Also, the hand-designed image descriptors are not optimal. We think the success of holistic methods is not in

contrast with our model, but tells us that better ways to exploit image information are needed. This could be an interesting direction for future work.

We introduced a novel approach for pose estimation in video sequences. Existing methods make use of motion models to propagate solutions from one frame to the next, or formulate the problem as inference in a graphical model with edges between corresponding body parts in adjacent frames that are associated with priors for body part motion. We observed that with the recent advances in methods for computing optical flow, also for large motion, useful information can come from considering computed dense optical flow as an observation. We defined *flowing puppets*, which are DS models that move across frames driven from dense optical flow. With flowing puppets we can propagate solutions from frame to frame across the video sequence without relying on motion prior models. We illustrated the effectiveness of this approach in particular for pose estimation of wrists, for which it is hard to define motion models in video with people that talk and interact animatedly.

DS is a view-dependent model, and we have presented applications where subjects were mostly seen from a frontal view. While to deal with arbitrary camera viewpoints it would be possible to learn a multi-view DS model, we have preferred to put our efforts in designing a 3D part-based puppet. In fact, with DS we generated training data by setting camera parameters to chosen values and generating training samples as projections of a SCAPE model, adding noise to the camera. When using DS for pose estimation from images, there is an assumption that the camera parameters used for generating the DS training data are not too different from the camera that generated the image data. This would still hold for the multi-view model, for which an assumption about the camera height should be made at the training stage. If our part-based representation was defined *before* the projection, meaning in 3D, we would have a more general model.

To deal with these issues, we introduced a part-based 3D model, named Stitched Puppet (SP), which is a mesh-based 3D model as realistic as SCAPE, that is able to represent pose-dependent deformations and people with different intrinsic shapes.

Intrinsic shape is a global attribute, and its parameterization would break the part-based assumption. We kept our tree-based structure and defined local intrinsic shape variables, showing how our optimization scheme can estimate realistic body shapes in the alignment of the 3D model to scans of real people with very different body shape, as shown in the previous chapter. We called the model a Stitched Puppet as the pairwise potentials between connected parts are simple functions defined over corresponding vertexes that encourage the parts to stitch at their interfaces. In contrast to previous non-realistic 3D part-based models, our stitching potentials do not represent priors over pose: SP is *just* a body model, and its tree-structure is consistent with the human body anatomy.

We applied the SP to the FAUST challenge, and obtained the first results on this benchmark. In applying the SP model, we used the same optimization scheme based on Max-Product belief propagation that we have used for pose estimation with the DS model.

In contrast to many techniques that involve aligning a 3D model to data, we did not use a data-to-model likelihood term that, as many authors observed, is crucial to avoid local minima. We believe that our optimization method based on the part-based model is able to explore the solution space in a more efficient way than traditional methods (gradient-based or stochastic), thus limiting the need of a data-to-model term. However, we also experienced that performing a local gradient-based optimization is important to align the model to the FAUST scans, as our 3D model, parametrized for pose-dependent and intrinsic shape deformations, is very high dimensional. Avoiding data-to-model terms is attractive for applying the model to noisy data with many outliers. In FAUST data is clean, meaning there are no outliers, while on the contrary missing data is possible. In future work it could be interesting to apply SP to noisy depth data or for fitting the 3D model to 2D images with cluttered background, where data-to-model terms are also hard to define.

In future work we want to apply the SP model to problems of pose estimation from images or depth data. These problems raise an interesting challenge for the 3D

part-based model. In fact, an advantage of using 3D models for pose estimation from multiple images is that occlusion is automatically handled by modeling the relative position of the object and the camera. This is not true for a part-based model, as for our distributed framework to hold we have to assume parts can be rendered on the image without knowing the state of the other parts. To deal with this we would have to add edges between nodes in our tree-structured graphical model, and use a loopy belief propagation method for inference.

Another direction for future work is the design of dressed models, with hair and maybe even shoes. This is very challenging, as for some types of clothing there is not an intuitive segmentation into parts, for example in the case of long skirts. Also clothes can involve further challenges in modeling their pose-dependent shape.

In conclusion we have introduced new models of the human body that are part-based but can realistically represent body shape variations with pose and, in case of the 3D model, with intrinsic shape. We have successfully applied the models to interesting and challenging problems in computer vision and computer graphics.

This work suggests that carefully defined realistic models can be important for computer vision, and should encourage more work at the intersection of vision and graphics.

Bibliography

- [1] G. J. Agin and T. O. Binford. Computer description of curved objects. *IEEE Trans. Comput.*, 25(4):439–449, Apr. 1976.
- [2] K. Alahari, G. Seguin, J. Sivic, and I. Laptev. Pose estimation and segmentation of people in 3d movies. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, pages 2112–2119, Dec. 2013.
- [3] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3):587–594, July 2003.
- [4] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1–8, June 2008.
- [5] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1014–1021, June 2009.
- [6] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 623–630, June 2010.
- [7] M. Andriluka, S. Roth, and B. Schiele. Discriminative appearance models for pictorial structures. *International Journal of Computer Vision*, 99(3):259–280, Sept. 2012.
- [8] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. SCAPE: Shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, July 2005.
- [9] D. Anguelov, P. Srinivasan, H. C. Pang, D. Koller, S. Thrun, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Advances in Neural Information Processing Systems, NIPS*, pages 33–40, Dec. 2004.
- [10] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, Mar. 2011.
- [11] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

- [12] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 3794–3801, June 2014.
- [13] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3D human pose annotations. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV, pages 1365–1372, Sept. 2009.
- [14] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences of the United States of America*, 103(5):1168–1172, Jan. 2006.
- [15] A. O. Bălan and M. J. Black. The naked truth: Estimating body shape under clothing. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision–ECCV 2008*, volume 5303 of *Lecture Notes in Computer Science*, pages 15–29. Springer Berlin Heidelberg, 2008.
- [16] A. O. Bălan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker. Detailed human shape and pose from images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 1–8, June 2007.
- [17] P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman. Upper body detection and tracking in extended signing sequences. *International Journal of Computer Vision*, 95(2):180–197, July 2011.
- [18] X. Burgos-Artizzu, D. Hall, P. Perona, and P. Dollár. Merging pose estimates across space and time. In *Proceedings of the British Machine Vision Conference*, BMVC, pages 58.1–58.11, Sept. 2013.
- [19] H. Chen, Z. J. Xu, Z. Q. Liu, and S. C. Zhu. Composite templates for cloth modeling and sketching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 943–950, June 2006.
- [20] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems*, NIPS, pages 1736–1744, Dec. 2014.
- [21] Y. Chen, Z. Liu, and Z. Zhang. Tensor-based human body modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 105–112, June 2013.
- [22] A. Cherian, J. Mairal, K. Alahari, and C. Schmid. Mixing body-part sequences for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 2361–2368, June 2014.

- [23] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 886–893, June 2005.
- [24] M. Dantone, J. Gall, C. Leistner, and L. van Gool. Human pose estimation using body parts dependent joint regressors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 3041–3048, June 2013.
- [25] M. Eichner, V. Ferrari, and S. Zurich. Better appearance models for pictorial structures. In *Proceedings of the British Machine Vision Conference*, BMVC’09, pages 1–11, Sept. 2009.
- [26] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari. 2D articulated human pose estimation and retrieval in (almost) unconstrained still images. *International Journal of Computer Vision*, 99(2):190–214, 2012.
- [27] S. M. A. Eslami and C. Williams. A generative model for parts-based object segmentation. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, NIPS, pages 100–107, Dec. 2012.
- [28] M. D. Fairchild. *Color Appearance Models*. Addison-Wesley, Reading, MA, USA, 1998.
- [29] P. Felzenszwalb and D. P. Huttenlocher. Efficient matching of pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 66–73, June 2000.
- [30] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, Jan. 2005.
- [31] V. Ferrari, M. Marin-Jimenez, , and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 1–8, June 2008.
- [32] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. on Computers*, 22(1):67–92, Jan 1973.
- [33] K. Fragkiadaki, H. Hu, and J. Shi. Pose from flow and flow from pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 2059–2066, June 2013.
- [34] O. Freifeld, A. Weiss, S. Zuffi, and M. J. Black. Contour people: A parametrized model of 2D articulated human shape. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 639–646, June 2010.
- [35] V. Gandhi and R. Ronfard. Detecting and Naming Actors in Movies using Generative Appearance Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, June 2013.

- [36] G. Gkioxari, P. Arbelaez, L. Bourdev, and J. Malik. Articulated pose estimation using discriminative armlet classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 3342–3349, June 2013.
- [37] P. Guan, O. Freifeld, and M. Black. A 2d human body model dressed in eigen clothing. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision–ECCV 2010*, volume 6311 of *Lecture Notes in Computer Science*, pages 285–298. Springer Berlin Heidelberg, 2010.
- [38] P. Guan, D. Reiss, L. and Hirshberg, A. Weiss, and M. J. Black. Drape: Dressing any person. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH)*, 31(4), July 2012.
- [39] P. Guan, A. Weiss, A. Balan, and M. J. Black. Estimating human shape and pose from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV, pages 1381–1388, Oct. 2009.
- [40] D. Hahnel, S. Thrun, and W. Burgard. An extension of the icp algorithm for modeling non-rigid objects with mobile robots. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI, pages 915–920, Aug. 2003.
- [41] K. Hara and R. Chellappa. Computationally efficient regression on a dependency graph for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 3390–3397, June 2013.
- [42] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A statistical model of human pose and body shape. *Computer Graphics Forum*, 28(2):337–346, Mar. 2009.
- [43] A. Hernandez-Vela, S. Escalera, and S. Sclaroff. Contextual rescoring for human pose estimation. In *Proceedings of the British Machine Vision Conference*, BMVC, Sept. 2014.
- [44] G. E. Hinton. Using relaxation to find a puppet. In *Proc. of the A.I.S.B. Summer Conference*, pages 148–157, July 1976.
- [45] D. Hirshberg, M. Loper, E. Rachlin, and M. Black. Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision–ECCV 2012*, volume 7577 of *Lecture Notes in Computer Science*, pages 242–255. Springer Berlin Heidelberg, 2012.
- [46] A. Ihler and D. McAllester. Particle belief propagation. In *AISTATS*, pages 256–263, Mar. 2009.
- [47] M. Isard. Pampas: Real-valued graphical models for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 613–620, June 2003.
- [48] Š. Ivekovič, E. Trucco, and Y. R. Petillot. Human body pose estimation with particle swarm optimisation. *Evol. Comput.*, 16(4):509–528, Dec. 2008.

- [49] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, pages 3192–3199, Dec. 2013.
- [50] V. John, E. Trucco, and S. Ivekovic. Markerless human articulated tracking using hierarchical particle swarm optimisation. *Image and Vision Computing*, 28(11):1530 – 1547, 2010.
- [51] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference, BMVC*, pages 12.1–11, Dec. 2010.
- [52] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated motion. In *IEEE Int. Conf. Automatic Face and Gesture Recognition*, pages 38–44, Oct. 1996.
- [53] M. Kiefel and P. Gehler. Human pose estimation with fields of parts. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision–ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, pages 331–346. Springer International Publishing, 2014.
- [54] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. In *ACM Transactions on Graphics (Proc. ACM SIGGRAPH)*, SIGGRAPH '11, pages 79:1–79:12, New York, NY, USA, 2011. ACM.
- [55] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Efficient discriminative learning of parts-based models. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, pages 552–559, Oct. 2009.
- [56] L. Ladicky, P. H. S. Torr, and A. Zisserman. Human pose estimation using a joint pixel-wise and part-wise formulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3578–3585, June 2013.
- [57] X. Lan and D. Huttenlocher. Beyond trees: Common factor models for 2D human pose recovery. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, pages 470–477, Oct. 2005.
- [58] S. Li, Z.-Q. Liu, and A. Chan. Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. *International Journal of Computer Vision*, pages 1–18, Sept. 2014.
- [59] Y. Lipman and T. Funkhouser. Mobius voting for surface correspondence. In *ACM Transactions on Graphics (Proc. ACM SIGGRAPH)*, SIGGRAPH '09, pages 72:1–72:12, New York, NY, USA, 2009. ACM.
- [60] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz. Exploring features in a bayesian framework for material recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 239–246, June 2010.

- [61] D. Marr and H. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. In *Proc. of the Royal Society of London, series B*, volume 200, 1140, pages 269–294, Feb. 1978.
- [62] C. Miller, O. Arikian, and D. Fussell. Frankenrigs: Building character rigs from multiple sources. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, I3D '10*, pages 31–38, New York, NY, USA, 2010. ACM.
- [63] A. Mittal, A. Zisserman, and P. H. S. Torr. Hand detection using multiple proposals. In *Proceedings of the British Machine Vision Conference, BMVC*, pages 1–11, Sept. 2011.
- [64] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2 of *CVPR*, pages 326–333, June 2004.
- [65] T. Neumann, K. Varanasi, S. Wenger, M. Wacker, M. Magnor, and C. Theobalt. Sparse localized deformation components. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH ASIA)*, 32(6):179:1–179:10, Nov. 2013.
- [66] W. Ouyang, X. Chu, and X. Wang. Multi-source deep learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2337–2344, June 2014.
- [67] J. Pacheco, S. Zuffi, M. J. Black, and E. Sudderth. Preserving modes and messages via diverse particle selection. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, volume 32(1), pages 1152–1160, Beijing, China, June 2014.
- [68] D. Parikh and L. Zitnick. Finding the weakest link in person detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, June 2011.
- [69] D. Park and D. Ramanan. N-best maximal decoders for part models. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, pages 2627–2634, Nov. 2011.
- [70] F. Perbet, S. Johnson, M.-T. Pham, and B. Stenger. Human body shape estimation using a multi-resolution manifold forest. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 668–675, June 2014.
- [71] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1–8. IEEE, IEEE, June 2013.
- [72] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, pages 3487–3494, Dec. 2013.

- [73] J. C. Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In M. Press, editor, *Advances in Large Margin Classifiers*, pages 61–74, 1999.
- [74] G. Pons-Moll, A. Baak, T. Helten, M. Müller, H.-P. Seidel, and B. Rosenhahn. Multisensor-fusion for 3d full-body human motion capture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 663–670, June 2010.
- [75] G. Pons-Moll and B. Rosenhahn. *Model-Based Pose Estimation*. Springer, June 2011.
- [76] J. Puwein, L. Ballan, R. Ziegler, and M. Pollefeys. Foreground consistent human pose estimation using branch and bound. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision–ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, pages 315–330. Springer International Publishing, 2014.
- [77] V. Ramakrishna, D. Munoz, M. Hebert, J. Andrew Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision–ECCV 2014*, volume 8690 of *Lecture Notes in Computer Science*, pages 33–47. Springer International Publishing, 2014.
- [78] D. Ramanan. Learning to parse images of articulated bodies. In *Advances in Neural Information Processing Systems*, pages 1129–1136, Dec. 2006.
- [79] D. Ramanan, D. A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 271–278, June 2005.
- [80] D. Ramanan, D. A. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):65–81, Jan. 2007.
- [81] J. Rodgers, D. Anguelov, H.-C. Pang, and D. Koller. Object pose detection in range scan data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR’06, pages 2445–2452, June 2006.
- [82] B. Rothrock, S. Park, and S.-C. Zhu. Integrating grammar and segmentation for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 3214–3221, June 2013.
- [83] B. Sapp, C. Jordan, and B. Taskar. Adaptive pose priors for pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 422–429, June 2010.
- [84] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 3674–3681, June 2013.

- [85] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision–ECCV 2010*, volume 6312 of *Lecture Notes in Computer Science*, pages 406–420. Springer Berlin Heidelberg, 2010.
- [86] B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1281–1288, June 2011.
- [87] G. Sheasby, J. Warrell, Y. Zhang, N. Crook, and P. Torr. Simultaneous human segmentation, depth and pose estimation via dual decomposition. In *Proceedings of the British Machine Vision Conference, Students Workshop, BMVC*, 2012.
- [88] H. Sidenbladh and M. J. Black. Learning the statistics of people in images and video. *International Journal of Computer Vision*, 54(1–3):183–209, Aug.–Oct. 2003.
- [89] L. Sigal and M. J. Black. Predicting 3D people from 2D pictures. In *LNCS vol. 4069, Proc. IV Conf. on Articulated Motion and Deformable Objects (AMDO)*, pages 185–195, July 2006.
- [90] L. Sigal, M. Isard, H. Haussecker, and M. J. Black. Loose-limbed people: Estimating 3D human pose and motion using non-parametric belief propagation. *International Journal of Computer Vision*, 98:15–48, 2011.
- [91] L. Sigal, M. I. Isard, B. H. Sigelman, and M. J. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *Advances in Neural Information Processing Systems, NIPS*, pages 1539–1546, Dec. 2003.
- [92] C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, 22(6):371–391, June 2003.
- [93] C. Stoll, N. Hasler, J. Gall, H.-P. Seidel, and C. Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, pages 951–958, Nov. 2011.
- [94] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1 of *CVPR*, pages 605–612, June 2003.
- [95] J. R. Tena, F. D. l. Torre, and I. Matthews. Interactive region-based linear 3d face models. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH)*, Aug. 2011.
- [96] T.-P. Tian and S. Sclaroff. Fast multi-aspect 2d human detection. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision–ECCV 2010*, volume 6313 of *Lecture Notes in Computer Science*, pages 453–466. Springer Berlin Heidelberg, 2010.
- [97] Y. Tian, C. Zitnick, and S. Narasimhan. Exploring the spatial hierarchy of mixture models for human pose estimation. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid,

- editors, *Computer Vision–ECCV 2012*, volume 7576 of *Lecture Notes in Computer Science*, pages 256–269. Springer Berlin Heidelberg, 2012.
- [98] R. Tokola, W. Choi, and S. Savarese. Breaking the chain: liberation from the temporal markov assumption for tracking human poses. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, pages 2424–2431, Dec. 2013.
- [99] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems, NIPS*, pages 1799–1807, Dec. 2014.
- [100] V. Vineet, G. Sheasby, J. Warrell, and P. H. Torr. Posefield: An efficient mean-field based method for joint estimation of human pose, segmentation, and depth. *Proceedings of International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 8081:180–194, Aug. 2013.
- [101] F. Wang and Y. Li. Beyond physical connections: Tree models in human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 996–608, June 2013.
- [102] H. Wang and D. Koller. Multi-level inference by relaxed dual decomposition for human pose segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2433–2440, June 2011.
- [103] Y. Wang, D. Tran, and Z. Liao. Learning hierarchical poselets for human parsing. In *CVPR*, pages 1705–1712, June 2011.
- [104] Y. Weiss and W. T. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Trans. Inf. Theor.*, 47(2):736–744, Sept. 2006.
- [105] S. Wuhler, C. Shu, and P. Xi. Landmark-free posture invariant human shape correspondence. *Vis. Comput.*, 27(9):843–852, Sept. 2011.
- [106] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(9):1744–1757, Sept. 2012.
- [107] Y. Yang and D. Ramanan. Articulated pose estimation using flexible mixtures of parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1385–1392, June 2011.
- [108] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2878–2890, Dec. 2013.

- [109] H. Zhang, A. Sheffer, D. Cohen-Or, Q. Zhou, O. van Kaick, and A. Tagliasacchi. Deformation-driven shape correspondence. In *Proceedings of the Symposium on Geometry Processing, SGP*, pages 1431–1439, Aire-la-Ville, Switzerland, 2008. Eurographics Association.
- [110] L. Zhu, Y. Chen, Y. Lu, C. Lin, and A. Yuille. Max margin and/or graph learning for parsing the human body. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1–8, June 2008.
- [111] S. Zuffi, O. Freifeld, and M. J. Black. From pictorial structures to deformable structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3546–3553, June 2012.
- [112] S. Zuffi, J. Romero, C. Schmid, and M. J. Black. Estimating human pose with flowing puppets. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, pages 3312–3319, Dec. 2013.