

Capturing Hand-Object Interaction and Reconstruction of Manipulated Objects

Dissertation

zur

Erlangung des Doktorgrades (*Dr. rer. nat.*)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Dimitrios TZIONAS

aus

Kozani, Griechenland

Bonn 2016

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Juergen Gall
2. Gutachter: Prof. Dr. Antonis Argyros

Tag der Promotion: 23.01.2017

Erscheinungsjahr: 2017

Capturing Hand-Object Interaction and Reconstruction of Manipulated Objects



Dimitrios Tzionas

Abstract

by Dimitrios Tzionas

for the degree of

Doctor rerum naturalium

Hand motion capture with an RGB-D sensor gained recently a lot of research attention, however, even most recent approaches focus on the case of a single isolated hand. We focus instead on hands that interact with other hands or with a rigid or articulated object.

Our framework successfully captures motion in such scenarios by combining a generative model with discriminatively trained salient points, collision detection and physics simulation to achieve a low tracking error with physically plausible poses. All components are unified in a single objective function that can be optimized with standard optimization techniques. We initially assume a-priori knowledge of the object's shape and skeleton.

In case of unknown object shape there are existing 3d reconstruction methods that capitalize on distinctive geometric or texture features. These methods though fail for textureless and highly symmetric objects like household articles, mechanical parts or toys. We show that extracting 3d hand motion for in-hand scanning effectively facilitates the reconstruction of such objects and we fuse the rich additional information of hands into a 3d reconstruction pipeline.

Finally, although shape reconstruction is enough for rigid objects, there is a lack of tools that build rigged models of articulated objects that deform realistically using RGB-D data. We propose a method that creates a fully rigged model consisting of a watertight mesh, embedded skeleton and skinning weights by employing a combination of deformable mesh tracking, motion segmentation based on spectral clustering and skeletonization based on mean curvature flow.

To my family
Maria, Konstantinos, Glyka.

In the loving memory of
γιαγιά Όλγα & παππούς Γιάννης.
(Olga Matoula & Ioannis Matoulas)

ὥστε ἡ ψυχὴ ὡσπερ ἡ χεὶρ ἐστίν·
καὶ γὰρ ἡ χεὶρ ὄργανόν ἐστιν ὀργάνων,
καὶ ὁ νοῦς εἶδος εἰδῶν
καὶ ἡ αἴσθησις εἶδος αἰσθητῶν.

Ἀριστοτέλης,
Περὶ ψυχῆς

*It follows that the soul is analogous to the hand;
for as the hand is a tool of tools,
so the mind is the form of forms
and sense the form of sensible things.*

*Aristotle,
On the Soul*

Acknowledgements

It feels weird writing these lines, a big chapter has closed! At the same time having to thank so many people gives a nice feeling of both fulfillment and nostalgia.

I would like to start by thanking my advisor Juergen Gall. My appointment coincided with his first appointment as an independent group leader at MPI, and I am thankful that he trusted me as his first student under his full guidance. He introduced me to the field of motion capture and provided valuable guidance as to how to perform research in a principled way and target quality results without sacrifices. Judging from the opportunities opening up after our collaboration, I would like to deeply thank him for helping me evolve and for showing me important research directions.

Furthermore, I would like to thank the professors Antonis Argyros, Reinhard Klein and Carsten Urbach for being members of my PhD committee and reviewing this thesis. I thank the first also for giving me the opportunity to give my first research-overview talk and get valuable experience in exposing my work to a critical audience.

It was a privilege that in the beginning of my PhD our group was part of the Perceiving Systems Department of Michael Black at the Max Planck Institute for Intelligent Systems in Tübingen. I consider this the most important step in my PhD, as it was my first real international working experience and my stay there was utterly didactic; I learned how to raise the bar for my goals and skip the low hanging fruit. So big thanks to the group there: Andreas, Chaohui, Deqing (also officemate for a month), Elena, Emma, Eric, Federica, Hueihan, Jessica, Jon, Jonas, Martin, Mat, Melanie, Naejin, Naureen (also officemate for a couple of months), Peter, Silvia, Søren and Varun. Special thanks goes out to Javier for always offering advice when asked, to Michael for the advices, reference letters and for ensuring economical stability even during relocation to Bonn, as well as to Aggeliki for her support and advices not only during the PhD but even before arriving in Tübingen. Big thanks also to the Tübingen “gang” for making life easier and more enjoyable; Amalia, Eleni, Fanis, Fanis (x2), Giannis, and the two “matakia” Dimitra-Despoina and Takis, the latter especially for making me feel welcome even before day one.

As we later moved to the University of Bonn to start a new computer vision lab, we had the opportunity to start fresh for the second time during the PhD and gain valuable experience in starting a group from scratch. There our group also got bigger, so thanks to Alex, Hilde, Martin and Umar for all the help, discussions and tips. For the same reasons I would like to thank the “computer vision friends” Dominik and Jens. Thank you also to Pablo for working with me and to Marc and Luca from ETH Zürich for the nice collaboration for the IJCV paper, the patience with countless emails and the advices. Life out of office was enjoyable due to the Bonn “gang”, so big thanks to John (also for introducing me in this), Elena, Maria, Frantzi, Olympia, Antonis, Spyros, as well to Xenia for making things beautifully messy and the occasional chicken with feta.

The co-traveler in both Tübingen and Bonn was Abhilash, the partner in crime as

my officemate for the whole PhD. Big thanks for tolerating me, for the endless support, advices, explanations and for being the firewall against PhD stress at deadlines.

I wouldn't forget to thank the people from Thessaloniki that played and still play an important role. A big thank you goes to my mentor Leontios Hadjileontiadis for being the inspiration to pursue a PhD and to always aim at blue sky ideas, no matter what. This realization wouldn't have been possible without the "Epione" team Leontios, Kostas, Stamatis and Stefanos with whom we challenged our boundaries and the "Epione friend" Paris, always available around the clock to help, support and advise. Big thanks to Tonia as well for the friendship, countless discussions and consulting for important decisions and steps.

Finally I would like to thank the constant variables in life, the ones that give meaning to everything. Big thanks to Omiros and Alexis for being lifetime assets as true friends. Mentioning my extended family and friend circle wouldn't be possible, but each one already knows my gratitude. Special thanks to the two "matoulakia" Nina and Olga for being last-minute proofreading ninjas.

This thesis is dedicated to my family Kostas, Maria and Glyka for their unconditional love and support and for all the reasons that are impossible to describe in words. They are the reason for true prosperity in life and the driving force that turns dreams into reality. This thesis partially belongs to them too.

During the PhD I gained a lot of new experiences, but certain ones will be very much missed, so this thesis is also dedicated in the loving memory of my grandparents, yiayia Olga and papous Yiannis.

To all the people above,

ευχαριστώ!

Financial support was provided by the Max Planck Society for the period 05.2012 - 06.2013 and the DFG Emmy Noether program (GA 1927/1-1) during 07.2013 - 09.2016.

The photo of the cover was kindly donated by the photographer Marc Mennigmann. The photo is part of Marc's project "*HANDS: The story of music through the story of the hands that make it*"¹ and depicts the hands of the musician Steven Wilson². Thank you Marc for trusting me with your art for the front page of my PhD thesis.

¹<http://hands.mennigmann.com>

²<http://stevenwilsonhq.com>

Contents

List of Figures	vi
List of Tables	vii
Nomenclature	vi
Publications	viii
1 Introduction	1
1.1 Motivation	1
1.2 Problem formulation	2
1.3 Importance	2
1.3.1 Academic Interest	2
1.3.2 Applications	3
1.4 Contributions	4
1.4.1 Evaluation for Hand Pose Estimation	4
1.4.2 Capturing Hands in Action	4
1.4.3 3D Object Reconstruction from Hand-Object Interactions	5
1.4.4 Reconstructing Object Skeletons from RGB-D Videos	5
1.5 Thesis Structure	5
2 Hand Motion Capture	7
2.1 Introduction	7
2.2 Challenges	8
2.3 Related Work	8
2.3.1 Generative Methods	8
2.3.2 Discriminative Methods	9
2.3.3 Hybrid Methods	10
2.3.4 Hand Models	10
2.3.5 Visual Cues	12
2.3.6 Hands in Action	12
2.3.7 Contact Points for Hand-Object Interaction	13
2.3.8 Datasets	14
2.4 Summary	16
3 A Comparison of Directional Distances for Hand Pose Estimation	17
3.1 Introduction	17
3.2 Related Work	18
3.3 Hand Pose Estimation	19
3.4 Generalized Chamfer Distance	20
3.5 Benchmark	21

3.6	Experiments	22
3.6.1	Implementation Details	22
3.6.2	Results	23
3.7	Summary	24
4	Capturing Hands in Action using Discriminative Salient Points and Physics Simulation	27
4.1	Introduction	27
4.2	Pose Estimation	28
4.2.1	Hand and Object Models	29
4.2.2	Objective Function	31
4.2.3	Multicamera RGB	42
4.3	Experimental Evaluation	42
4.3.1	Monocular RGB-D - Hand-Hand Interactions	43
4.3.2	Monocular RGB-D - Hand-Object Interactions	49
4.3.3	Limitations	53
4.3.4	Multicamera RGB	54
4.4	Summary	57
5	3D Object Reconstruction from Hand-Object Interactions	63
5.1	Introduction	64
5.2	Related work	65
5.3	Hand motion capture for in-hand scanning	66
5.3.1	Preprocessing	66
5.3.2	Hand motion capture	67
5.4	Object reconstruction	67
5.4.1	Contact Points Computation	67
5.4.2	Reconstruction	68
5.5	Experiments	70
5.5.1	Synthetic data	70
5.5.2	Realistic data	71
5.6	Summary	76
6	Reconstructing Articulated Rigged Models from RGB-D Videos	81
6.1	Introduction	81
6.2	Related work	82
6.3	Mesh motion	83
6.3.1	Preprocessing	84
6.3.2	Mesh tracking	84
6.4	Kinematic model acquisition	85
6.4.1	Motion segmentation	86
6.4.2	Kinematic topology	87
6.5	Experiments	89
6.6	Summary	92

7	Conclusions	103
7.1	Overview	103
7.2	Contributions and Discussion	103
7.2.1	Evaluation for Hand Pose Estimation	104
7.2.2	Capturing Hands in Action	104
7.2.3	3D Object Reconstruction from Hand-Object Interactions	105
7.2.4	Reconstructing Object Skeletons from RGB-D Videos	106
7.3	Future Work	106
7.4	Summary	109
	Bibliography	111
	Curriculum Vitae	131

List of Figures

2.1	Hand models used by several generative or hybrid approaches	11
3.1	Initial pose, synthetic and realistic target silhouettes of a camera view .	19
3.2	Performance evaluation of DCH-Thres with different values of τ and both signed (360) and unsigned (180) distance $ \cdot _\phi$	23
3.3	Performance evaluation of DCH-DT3 with different values of λ for 16 quantization bins, as well as different quantizations of ϕ for $\lambda = 25$. . .	25
3.4	Evaluation of DCH-Quant and DCH-Quant2 with different quantizations of ϕ	26
3.5	Comparison of all distances with best settings	26
4.1	Qualitative results of our method for the case of hand-hand interaction .	28
4.2	Hand model used for tracking	29
4.3	Object models used for tracking and their DoF	30
4.4	Segmentation of the meshes based on the skinning weights	31
4.5	Mesh intersections during interactions without the collision term	34
4.6	Distance field Ψ_{f_s} generated by the face f_s	34
4.7	Correspondences between the fingertips ϕ_t of the model and the associated detections δ'_s	38
4.8	Physical plausibility during hand-object interaction, with and without the physics component \mathcal{P}	39
4.9	Low resolution representation of the hands and objects for the physics simulation component \mathcal{P}	40
4.10	Finger parts that form all possible supporting combinations in the physics simulation component \mathcal{P}	40
4.11	The function that penalizes the deviation from the defined angle limits for each revolute joint	41
4.12	Hand joints used for quantitative evaluation	43
4.13	Precision-recall plot of a fingertip detector trained on depth or RGB images for our RGB-D dataset and the Dexter dataset	45
4.14	Qualitative comparison of our tracker with the FORTH tracker	48
4.15	Failure case due to missing data and detection errors	51
4.16	Number of iterations required to converge for LO + \mathcal{SCP} and LO	54
4.17	Quantitative evaluation of our algorithm on noisy data with respect to the salient point detection rate and the number of iterations	56
4.18	Results for RGB-D sequences of one or two interacting hands	58
4.19	The impact of the physics component with some qualitative results for the setups LO + \mathcal{SCx} and LO + \mathcal{SCP}	59
4.20	Results for RGB-D sequences of hand-object interaction	60
4.21	Results for RGB sequences of hand-hand or hand-object interaction . .	61

5.1	Reconstruction of a symmetric, textureless object	64
5.2	The hand tracker used in the in-hand scanning pipeline	66
5.3	Illustration of the contact correspondences (X_{hand}, X'_{hand})	68
5.4	Reconstruction without and with hand tracking on synthetic data	71
5.5	The objects to be reconstructed, with high symmetry and lack of distinctive geometrical and textural features	72
5.6	Quantitative evaluation of the weight γ_t of the energy function	73
5.7	The bowling-pin object enriched with 2d texture using stickers	74
5.8	Qualitative evaluation of the weight γ_t of the energy function	75
5.9	Reconstruction failure when a contact detector is used instead of the contact points based on a hand tracker	76
5.10	Qualitative results for the bowling-pin object without the term E_{icp}	77
5.11	Qualitative comparison of different in-hand scanning systems, including KinFu, Skanect, our pipeline with(out) hand motion data	78
5.12	Qualitative results of our pipeline for all four objects of Figure 5.5 when a hand rotates the object in front of the camera	79
6.1	Tracked mesh with the deformable tracker and the corresponding 3d vertex trajectories	86
6.2	The steps of our pipeline to reconstruct articulated rigged models from RGB-D videos	88
6.3	Target poses for each object with escalating difficulty. The target poses are used for quantitative evaluation	89
6.4	Deformable tracking while spanning the parameter space of γ_{def} that steers the influence of the smoothness and data terms	90
6.5	Results for the best configuration $(\gamma_{def}, \lambda_{thr})$ for each object. The motion segments and the inferred 3d skeleton are shown.	91
6.6	Results for the four configurations $(\gamma_{def}, \lambda_{thr})$ from the proposed parameters. The motion segments and the inferred 3d skeleton are shown.	92
6.7	Motion segments for all sets $(\gamma_{def}, \lambda_{thr})$ for the object “spray”.	94
6.8	Motion segments for all sets $(\gamma_{def}, \lambda_{thr})$ for the object “donkey”.	95
6.9	Motion segments for all sets $(\gamma_{def}, \lambda_{thr})$ for the object “lamp”.	96
6.10	Motion segments for all sets $(\gamma_{def}, \lambda_{thr})$ for the object “pipe 1/2”.	97
6.11	Motion segments for all sets $(\gamma_{def}, \lambda_{thr})$ for the object “pipe 3/4”.	97
6.12	Inferred 3d skeleton for all sets $(\gamma_{def}, \lambda_{thr})$ for the object “spray”.	98
6.13	Inferred 3d skeleton for all sets $(\gamma_{def}, \lambda_{thr})$ for the object “donkey”.	99
6.14	Inferred 3d skeleton for all sets $(\gamma_{def}, \lambda_{thr})$ for the object “lamp”.	100
6.15	Inferred 3d skeleton for all sets $(\gamma_{def}, \lambda_{thr})$ for the object “pipe 1/2”.	101
6.16	Inferred 3d skeleton for all sets $(\gamma_{def}, \lambda_{thr})$ for the object “pipe 3/4”.	102

List of Tables

2.1	Related literature regarding the use of contact points	14
2.2	Public datasets for hand pose tracking/estimation	15
3.1	Mean error \pm std.dev. and average time for 1,5,10,15 frame differences for all distance measures	24
4.1	Mapping between mesh fingertips ϕ_t and fingertip detections δ_s	36
4.2	Evaluation of point-to-point ($p2p$) and point-to-plane ($p2plane$) distance metrics, along with iterations number of the optimization framework	44
4.3	Evaluation of the weighting parameter λ for the mapping between mesh fingertips ϕ_t and fingertip detections δ_s	45
4.4	Evaluation of collision weights γ_c for the setup $LO + \mathcal{SC}$	46
4.5	Comparison of the proposed collision term based on 3d distance fields with correspondences between vertices of colliding triangles	46
4.6	Evaluation of the components $LO, \mathcal{C}, \mathcal{S}$ of our pipeline	47
4.7	Pose estimation error for each sequence of our single-hand and hand- hand interaction dataset	47
4.8	Quantitative comparison with the FORTH tracker	48
4.9	Evaluation of our tracker for each sequence of the Dexter dataset	49
4.10	Evaluation of the components $LO, \mathcal{C}, \mathcal{S}$ of our pipeline on the Dexter dataset	50
4.11	Evaluation of the friction value for the physics simulation component \mathcal{P}	52
4.12	Evaluation of physics weights γ_{ph} for the setup $LO + \mathcal{SCP}$	52
4.13	Evaluation of the components $LO, \mathcal{C}, \mathcal{S}$ and \mathcal{P} of our pipeline	53
4.14	Pose estimation error for each sequence of our hand-object interaction dataset	53
4.15	Quantitative evaluation of the algorithm with respect to the used visual features: edges \mathcal{E} , collisions \mathcal{C} , optical flow \mathcal{O} , and salient points \mathcal{S}	55
4.16	Quantitative results for the multicamera RGB sequences	57
5.1	Quantitative evaluation of the captured object shapes with ground- truth. We compare our setup with the methods KinFu and Skanect	72
5.2	Quantitative evaluation of the pairwise registration for our pipeline based on hand pose $E_{contact}$ and the detector-based baseline E_{detect}	75
6.1	Evaluation of our approach using the target poses in Figure 6.3 while spanning the parameter space $(\gamma_{def}, \lambda_{thr})$	91
6.2	Evaluation of our method and resulting kinematic models for the public sequences “Bending a Pipe” and “Bending a Rope” of Chapter 4	93

Nomenclature

Abbreviations

The abbreviations used in this thesis are listed below in alphabetical order.

ARAP	As-Rigid-As-Possible
AUC	Area Under the Curve
AR	Augmented Reality
BVH	Bounding Volume Hierarchies
CAD	Computer Aided Design
CH	Chamfer distance without any orientation information
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSHOT	Color-SHOT, a SHOT variant incorporating texture
DCH-Thres	Directional CH - Corresp. with inconsistent orientations are rejected
DCH-Quant	Directional CH - Uses a 2d distance field and normal angle quantization
DCH-Quant2	Directional CH - A DCH-Quant variant with soft binning
DCH-DT3	Directional CH - Approximated with a 3d distance field (DT3)
DT3	3D Distance Field
DOF	Degrees of Freedom
GMM	Gaussian-Mixtures-Model
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HCI	Human Computer Interaction
HMD	Head Mounted Display
HOG	Histogram of Oriented Gradients
ICP	Iterative Closest Point
ISS3D	Intrinsic Shape Signatures keypoints
KLT	Kanade-Lucas-Tomasi feature tracker
LBS	Linear Blend Skinning model
NUI	Natural User Interface
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
RANSAC	RANdom SAmple Consensus
RGB	Red, Green, Blue
RGB-D	Red, Green, Blue - Depth
SDF	Signed Distance Function
SfM	Structure-From-Motion
SHOT	Signature of Histograms of Orientations
SIFT	Scale-Invariant Feature Transform

SLAM	Simultaneous Localization And Mapping
SoG	Sum of Gaussians
Std.Dev.	Standard Deviation
SVM	Support Vector Machine
ToF	Time-of-Flight
TSDf	Truncated Signed Distance Function
VR	Virtual Reality

Units

The units used in this thesis are listed below in alphabetical order.

cm	Centimeter
fps	Frames Per Second
kg	Kilogram
mm	Millimeter
px	Pixel

Mathematical Symbols

The mathematical symbols used in this thesis are listed below in approximate order of first appearance and grouped per chapter.

Chapter 3

\mathbf{V}	3d vertex
κ	Skinning weight
θ	Pose parameters
T	Rigid transformation
$\hat{\xi}$	Twist
$\theta\hat{\xi}$	Twist-encoded rigid body transformation
$se(3)$	Lie algebra - the set of 4x4 matrices for translation and rotations
$SE(3)$	Lie group SE(3) - 3D Rigid Transformations
\mathbf{q}	A 2d point in the image
c	Camera view
\mathbf{d}	Direction of a Plucker line
\mathbf{m}	Moment of a Plucker line
N	Cardinality of a set
I	Identity matrix
\mathcal{C}	Set of contour pixels
\mathcal{P}	Set of projected rim vertices
\mathbf{p}	A projected rim vertex $\in \mathcal{P}$ on the 2d image plane
$d_{Chamfer}$	Chamfer distance

$d(\cdot, \cdot)$	Distance function in 2d
$f(\cdot, \cdot)$	Penalty function
Z	Normalization factor
\mathcal{K}	Threshold on the maximum squared distance
τ	Circular distance threshold
ϕ	Angle of a 3d normal projected in 2d
$ \cdot _\phi$	Circular distance
$\ \cdot\ $	Euclidean distance ($L2$ norm)
λ	Weight

Chapter 4

κ	Skinning weight
$\vartheta\xi$	Twist
$\vartheta\hat{\xi}$	Twist action
u	Component of twist
ω	Component of twist
$\exp(\vartheta\hat{\xi})$	Exponential map operator
$T(\vartheta\hat{\xi})$	Group action
R	Rotation matrix 3×3
t	Translation vector 3×1
$p(\cdot)$	Parent bone
K	Number of revolute joints
\hat{T}	Relative transformation
E	Energy / Objective function
$E_{model \rightarrow data}$	Energy term - Fits the model to the data
$E_{data \rightarrow model}$	Energy term - Fits the data to the model
$E_{collision}$	Energy term - Collision penalizer
$E_{salient}$	Energy term - Salient point term based on fingertip detections
$E_{physics}$	Energy term - Grasp enhancing term based on physics simulation
$E_{anatomy}$	Energy term - Anatomical prior for the hand joint angles
$E_{regularization}$	Energy term - Regularizer for immunity to occlusions or sensor noise
γ_c	Steering weight for $E_{collision}$
γ_{ph}	Steering weight for $E_{physics}$
γ_a	Steering weight for $E_{anatomy}$
γ_r	Steering weight for $E_{regularization}$
D	Current preprocessed depth image (or point cloud)
\mathbf{n}	Normal in 3d
X	Point in the 3d point cloud
Ψ	Distance field (3d)
f	Face / triangle
\mathcal{C}	Set of colliding triangles
\mathbb{R}	Real numbers
\mathbf{o}_f	Circumcenter of a face f

r_f	Radius of the circumcircle of face f
\mathcal{S}	Salient point detection
c_{thr}	Detection confidence threshold
\mathcal{T}	Number of model fingertips
ϕ_t	Fingertip of the model
\mathcal{S}	Number of fingertip detections
δ_s	Fingertip detection
V	Virtual model fingertip / fingertip detection
e	Binary assignment variable
α	Binary assignment variable
β	Binary assignment variable
w_{st}	3D distance between detection δ_s and finger ϕ_t
w_s	Detection confidence
λ	Assignment cost
δ'_s	3D point cloud in detection δ_s bounding box
c_s	Confidence of detection δ_s
\mathcal{P}	Physics Simulation
C_{all}	Total number of correspondences
i_{thr}	Maximum number of iterations
Π	Projection function
x	2D point in an image
$p2p$	Point-to-point distance metric
$p2plane$	Point-to-plane distance metric
$d2m$	<i>Data-to-model</i>
$m2d$	<i>Model-to-data</i>
ε	Stopping criterion (mm)
\mathcal{E}	Edges
\mathcal{C}	Collisions
\mathcal{O}	Optical flow
\mathcal{S}	Salient points

Chapter 5

D_h	Masked RGB-D images for the hand using a GMM
D_o	Masked RGB-D images for the object using a GMM
(X_{hand}, X'_{hand})	3D Contact correspondences (source, target frame)
(x_{2d}, x'_{2d})	2D SIFT correspondences (source, target frame)
(X_{2d}, X'_{2d})	3D SIFT correspondences (source, target frame)
(X_{3d}, X'_{3d})	3D ISS3D/CSHOT correspondences (source, target frame)
(X_{icp}, X'_{icp})	3D ICP correspondences (source, target frame)
\mathcal{C}_{hand}	Set of contact correspondences
\mathcal{C}_{feat2d}	Set of correspondences from 2d SIFT features
\mathcal{C}_{feat3d}	Set of correspondences from 3d ISS3D keypoints / CSHOT features
\mathcal{C}_{icp}	Set of ICP correspondences

$E_{contact}$	Energy based on \mathcal{C}_{hand} correspondences
E_{feat2d}	Energy based on \mathcal{C}_{feat2d} correspondences
E_{feat3d}	Energy based on \mathcal{C}_{feat3d} correspondences
E_{visual}	Energy based on E_{feat2d} and E_{feat3d} energies
E_{icp}	Energy based on \mathcal{C}_{icp} correspondences
$SO(3)$	Special orthogonal group - 3D rotation group
$\varphi(\cdot)$	Function back-projecting 2D points in 3D: $\mathbb{R}^2 \rightarrow \mathbb{R}^3$
γ_t	Weight that steers the influence of E_{visual} and $E_{contact}$
T	Rigid transformation $T = (R, t)$
\mathbf{R}	Rotation matrix (unknown for optimization)
\mathbf{t}	Translation vector (unknown for optimization)
p	Parameter corresponding to a distinctive dimension of an object
p_{est}	Estimated value for a distinctive dimension of an object
p_{GT}	Ground-truth value for a distinctive dimension of an object
(X_{gt}, X'_{gt})	Ground-truth correspondences (source, target frame) for evaluation

Chapter 6

\mathcal{M}	Mesh
\mathcal{E}_{smooth}	Smoothness term in energy
γ_{def}	Steering weight in deformable tracking energy
\mathbf{X}	Closest point of the 3d point cloud for a mesh vertex of \mathcal{M}
\mathbf{V}	Vertices of the mesh \mathcal{M}
\mathbf{L}	Cotangent Laplacian matrix
$\mathcal{N}_1(\mathbf{V})$	Set of one-ring neighbor vertices of vertex \mathbf{V}
α_{ij}	One of the two angles opposite of the edge (i, j)
β_{ij}	One of the two angles opposite of the edge (i, j)
A	Voronoi cell
\mathcal{T}	Vertex 3d trajectory
Φ	Affinity matrix
λ	Weight
dt	Frame interval for computation of relative distances
$d(\mathcal{T}_i, \mathcal{T}_j)$	Distance between two trajectories
$d^v(\mathcal{T}_i, \mathcal{T}_j)$	Distance between two trajectories - Only position into account
$d^n(\mathcal{T}_i, \mathcal{T}_j)$	Distance between two trajectories - Only normal angles into account
\mathbf{N}	Vertex normals
\mathcal{L}	Normalized Laplacian graph
λ_{thresh}	Threshold on the eigenvalues
\mathcal{O}	Reconstructed object from an initial pose

Publications

The work presented in this thesis has been evaluated and approved by the computer vision community through the peer-reviewed papers listed below. The conference papers had to undergo a double-blind review process, while the journal paper had to undergo a single-blind review process of high standards. The new datasets for each paper are public in the websites noted, along with helpful software and supplementary material. The source code of the proposed ICCV 2015 method is also public.

- D. Tzionas and J. Gall.
A comparison of directional distances for hand pose estimation.
In German Conference on Pattern Recognition (GCPR), 2013.
http://files.is.tue.mpg.de/dtzionas/GCPR_2013
[Tzionas and Gall, 2013]
- D. Tzionas, A. Srikantha, P. Aponte and J. Gall.
Capturing hand motion with an RGB-D sensor, fusing a generative model with salient points.
In German Conference on Pattern Recognition (GCPR), 2014.
http://files.is.tue.mpg.de/dtzionas/GCPR_2014
[Tzionas et al., 2014]
- D. Tzionas and J. Gall.
3D object reconstruction from hand-object interactions.
In International Conference on Computer Vision (ICCV), 2015.
<http://files.is.tue.mpg.de/dtzionas/In-Hand-Scanning>
[Tzionas and Gall, 2015]
- D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys and J. Gall.
Capturing Hands in Action using Discriminative Salient Points and Physics Simulation.
In International Journal of Computer Vision (IJCV), 118(2):172-193, 2016.
<http://files.is.tue.mpg.de/dtzionas/Hand-Object-Capture>
[Tzionas et al., 2016]
- D. Tzionas and J. Gall.
Reconstructing Articulated Rigged Models from RGB-D Videos.
In European Conference on Computer Vision Workshops 2016 (ECCVW'16)
<http://files.is.tue.mpg.de/dtzionas/Skeleton-Reconstruction>
[Tzionas and Gall, 2016]

The IJCV paper is joint work with L. Ballan and M. Pollefeys and includes small parts of their past work [Ballan et al., 2012].

Introduction

“We live between two realms: our physical environment and cyberspace. Despite our dual citizenship, the absence of seamless couplings between these parallel existences leaves a great divide between the worlds of bits and atoms. At the present, we are torn between these parallel but disjoint spaces.”

*Hiroshi Ishii and Brygg Ullmer
MIT Media Laboratory
[Ishii and Ullmer, 1997]*

Contents

1.1	Motivation	1
1.2	Problem formulation	2
1.3	Importance	2
1.3.1	Academic Interest	2
1.3.2	Applications	3
1.4	Contributions	4
1.4.1	Evaluation for Hand Pose Estimation	4
1.4.2	Capturing Hands in Action	4
1.4.3	3D Object Reconstruction from Hand-Object Interactions	5
1.4.4	Reconstructing Object Skeletons from RGB-D Videos	5
1.5	Thesis Structure	5

1.1 Motivation

The advent of commodity RGB-D sensors has transformed the landscape of computer vision in the last decade. By combining an RGB and a depth camera they provide not only light intensity and color information, but also the 3d distance for each observed point. As a result they greatly facilitate the 3d perception of computers by effortlessly measuring the 3d structure of the observed scene. Integration of RGB-D sensors in smartphones and tablets promises to make spatially-aware computing truly ubiquitous.

In this direction researchers pursue the long term goal of holistic 3d scene understanding. This involves modeling scenes, objects and people, detecting them in

images or videos, estimating and tracking their pose. It further involves inferring semantic information about the scene structure, the actions of people and the use of objects. Along with the increasingly popular augmented or virtual reality (AR/VR) head mounted displays (HMD), holistic 3d scene understanding forms the gate between the two worlds in which we are citizens, the physical and digital worlds.

This gate motivates us to radically rethink the interface between people, technology and society, to develop tools that amplify both machine and human perception, intelligence and creativity. It enables the design of transparent human-computer interaction paradigms responsive to our attention, actions and changes in the real world, the design of interfaces hiding unobtrusively in plain sight. It facilitates the seamless and harmonic “coupling of bits and atoms”.

1.2 Problem formulation

An important aspect of this long term vision is interaction of humans with the environment. Humans spend a big part of their daily life interacting with surrounding objects to perform tasks and enhance their productivity. For such manipulation tasks, hands are the main interface between the human brain and the physical world. They are used to touch, grasp, lift and manipulate objects. As we rapidly move to the era of virtual and augmented reality, as well as ambient intelligence, humans should be able to interact with both physical and virtual objects.

For this the computer needs spatial awareness through a virtual representation of a scene. This representation should be dynamic in a twofold way. On the one side, it should capture transparently the pose of known objects and the interacting hands. On the other side, it should be enriched by reconstructing unknown objects through observation and manipulation. An update of the virtual representation through time enforces a consistent reflection of the changes in the physical world, providing a one-to-one mapping between the real and the virtual world.

1.3 Importance

The importance of the problems defined in Section 1.2 is shown by the intense research interest in this direction not only of academic labs [Erol et al., 2007a, Supančič III et al., 2015, Salvi et al., 2007], but also of important industry corporations like Microsoft [Taylor et al., 2016, Izadi et al., 2011], Leap Motion [LEAP MOTION], Oculus [OCULUS, NIMBLE VR], and others.

1.3.1 Academic Interest

The academic interest for these problems spans the space of both the underlying technical and perceptual aspects, raising important scientific questions worth exploring.

Motion capture of hands interacting with other hands or objects is a high dimensional problem that makes the search for the correct pose difficult, time-consuming and error-prone. In that respect there are several open problems including, among others:

shape and kinematic model representation, tracking the pose in video sequences, pose estimation from a single frame for initialization or automatic recovery in case of tracking failure, recognition of distinctive parts that robustify pose estimation, penalization of collisions between tracked meshes, suitable optimization techniques or method fusion.

When a Head Mounted Display (HMD) is used for augmented or virtual reality (AR/VR) applications additional questions are raised. Since the user's hands are replaced or augmented with virtual ones, the tracking speed and accuracy, as well as the shape of the hands in action greatly influence user's perception of "ownership" of the virtual hands, leaving large space for scientific exploration. Similar multi-user applications pose new questions about the concept of collaborative environments.

Finally, object reconstruction has several open problems regarding both the reconstruction of the shape and the kinematic model. The ideal shape representation for objects of all sizes is still an open problem, as well as suitable frame alignment methods, the use of semantic information along with lower level features, or for non-rigid-objects the inference of motion segments, kinematic joints and appropriate motion prior models.

1.3.2 Applications

Aside from pure academic interest, the problems of Section 1.2 are important because they also give rise to numerous practical applications.

Capturing the motion of human hands facilitates a plethora of Human Computer Interaction (HCI) techniques. The rich information of hand movement enables Natural User Interfaces (NUI) that can replace the Graphical User Interface (GUI), the main HCI paradigm for the last three decades along with the mouse and keyboard. A non-restrictive list of example applications includes gesture-based interfaces for home, mobile or wearable appliances, sign language recognition, touchless NUI interfaces for critical environments like a hospital operating room, rehabilitation applications, animation for movies, or monitoring applications for driving, sports or music.

By adding articulated object tracking, hand-object motion capture broadens the spectrum of possible novel applications. Along with a Head Mounted Display (HMD) it facilitates a wide range of augmented or virtual reality (AR/VR) applications like fully immersive 3d Computer Aided Design (CAD), collaborative virtual spaces and gaming. Along with mobile robots it aids data-driven methods for robot-object interaction through demonstration, while it also redefines the concept of telepresence and teleoperation

The idea of reconstructing 3d models of physical objects, including both their shape and kinematics, facilitates the creation of virtual blueprints of objects and their replication through 3d printing, the transfer of physical objects in virtual worlds like gaming environments, as well as educational or story-telling applications in the form of creating virtual or augmented reality scenes while placing, moving, transforming or animating objects in 3d with the user's hands.

1.4 Contributions

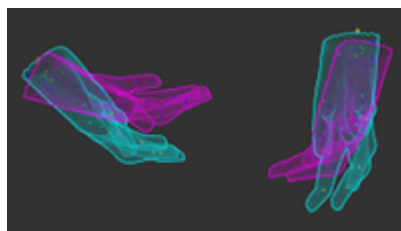
Despite the important applications described in Section 1.3.2, the majority of even the most recent works [Supančič III et al., 2015] focus on the case of a single isolated hand. However humans interface with the physical world by touching, grasping, lifting and manipulating objects. Often both hands are used collaboratively to enhance manipulation efficiency.

To this end, in this thesis we capture the 3d motion of one or two hands interacting with each other or with a known object, either rigid or articulated. In case of an unknown object we reconstruct its unknown shape with an in-hand scanning setup that incorporates the tracked 3d hand motion. For articulated objects the unknown skeleton is reconstructed by observing its deformations during manipulation.

This thesis contributes in the state-of-the-art by studying several aspects of the above problems as described in the following. All the new datasets are public in the websites noted in Chapter “Publications” (page vii), along with helpful software, supplementary material and source code for certain projects.

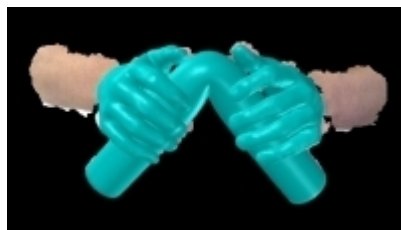
1.4.1 Evaluation for Hand Pose Estimation

Initially we present a protocol to evaluate features and methods during the implementation of a tracking pipeline without employing the latter for this, as it might not be yet complete. To this end, we create testing frame pairs of increasing difficulty and measure the pose estimation error separately for each of them. Following this protocol, we evaluate various directional distances in the context of silhouette-based 3d hand tracking, expressed as special cases of a generalized Chamfer distance form. This part is based on work published in [Tzionas and Gall, 2013].



1.4.2 Capturing Hands in Action

We then present a method to capture the 3d motion of hands in sequences where they interact with other hands or objects and present a framework that successfully captures motion in such interaction scenarios for both rigid and articulated objects. Our framework combines a generative model with discriminatively trained salient points to achieve a low tracking error and with collision detection and contact points based on physics simulation to achieve physically plausible estimates even in case of occlusions and missing visual data. Since all components are unified in a single objective function which is almost everywhere differentiable, it can be optimized with standard optimization techniques. Our approach works for monocular RGB-D sequences but can also be ap-



plied on setups with multiple synchronized RGB cameras. This part is based on work published in [Tzionas et al., 2014, 2016].

1.4.3 3D Object Reconstruction from Hand-Object Interactions

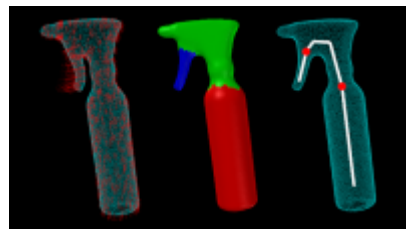
The above method is based on a-priori knowledge of the shape and the kinematic model of both the hands and the object. Such prior knowledge is a reasonable assumption for the hand that is an integral part of the human body. However a human or a robot navigating in the real world might have to interact with objects that are not modeled yet. In



that respect, we focus on the case of reconstructing the unknown shape of a rigid object during hand-object manipulations. Shape reconstruction approaches rely either on a camera rotating around the object or on in-hand scanning, for which an operator rotates the object in front of the camera with his hand(s). Although existing approaches reconstruct successfully the shape of many object classes, they need stable and distinctive geometric or texture features. As a result existing in-hand scanning systems fail for highly symmetric or textureless objects. However they traditionally ignore the motion information of the hands that rotate the object. We show that capturing and incorporating the rich information of 3d hand motion in an in-hand scanning pipeline facilitates the reconstruction of even featureless and highly symmetric objects. This part is based on work published in [Tzionas and Gall, 2015].

1.4.4 Reconstructing Object Skeletons from RGB-D Videos

Although shape reconstruction is enough for rigid objects, for articulated objects we might have to reconstruct their unknown kinematic model, i.e. an underlying skeleton that describes the object's structure and possible motion. However there is a lack of tools for RGB-D data that build rigged models of articulated objects consisting of a watertight mesh, embedded skeleton, and skinning weights. In this direction we present such a



method that combines deformable tracking of the known mesh, motion segmentation based on spectral clustering and skeletonization based on mean curvature flow. As a result our approach creates a fully rigged model of an articulated object from depth data of a single sensor. This part is based on work published in [Tzionas and Gall, 2016].

1.5 Thesis Structure

This thesis is divided in five main parts. The first summarizes the related literature for hand motion capture, that is the inspiration for this thesis. The rest of the parts

correspond to the contributions described in Section 1.4 for the problems defined in Section 1.2.

- In **Chapter 2** we present an overview of the *literature* for 3d *hand motion capture* methods. The presented methods motivate the hand tracking approach presented in this thesis, as well as applications that incorporate hand tracking.
- In **Chapter 3** we present an *evaluation protocol for hand pose estimation* based on testing frame pairs of increasing difficulty. Following this protocol we evaluate various directional distances for silhouette-based 3d hand tracking.
- In **Chapter 4** we present a method to *capture the 3d motion of hands in action*. The approach is successful for scenarios varying from hands in isolation to hands that interact with other hands or a known object, either rigid or articulated.
- In **Chapter 5** we present an *in-hand scanning approach* that incorporates the rich information of 3d hand motion. The approach reconstructs successfully the unknown shape of even highly symmetric and textureless rigid objects.
- In **Chapter 6** we present an approach that *reconstructs the unknown skeleton* of an articulated object from observations of its deformations. The output is a fully rigged model ready for tracking or animation.

Hand Motion Capture

“[A] user wearing a [see-through] HMD might hold up her real hand and see a virtual hand. This virtual hand should be displayed exactly where she would see her real hand, if she were not wearing an HMD. [...] [R]egistration errors [] result in visual-visual conflicts between the images of the virtual and real hands. [...] Even tiny offsets in the images of the real and virtual hands are easy to detect.”

*A Survey of Augmented Reality
[Azuma, 1997]*

Contents

2.1	Introduction	7
2.2	Challenges	8
2.3	Related Work	8
2.3.1	Generative Methods	8
2.3.2	Discriminative Methods	9
2.3.3	Hybrid Methods	10
2.3.4	Hand Models	10
2.3.5	Visual Cues	12
2.3.6	Hands in Action	12
2.3.7	Contact Points for Hand-Object Interaction	13
2.3.8	Datasets	14
2.4	Summary	16

2.1 Introduction

Capturing the 3d motion of human hands is an important research topic in computer vision since decades [Heap and Hogg, 1996, Erol et al., 2007b] due to its importance for numerous applications including, but not limited to, computer graphics, animation, human computer interaction, rehabilitation and robotics. The research interest has increased in the last few years [Erol et al., 2007a, Ye et al., 2013, Supančić III et al., 2015] with the recent technology advancements of consumer RGB-D sensors.

2.2 Challenges

Despite the increased research interest though the problem of capturing the motion of hands is still considered to be unsolved. Although it is a special instance of the well studied full human body tracking [Gavrila, 1999, Poppe, 2007, Moeslund and Granum, 2001, Moeslund et al., 2006, Helten et al., 2013], it can not be easily solved by applying known techniques for human pose estimation like [Shotton et al., 2011] to human hands. The reason is that, despite the similarities, capturing the motion of hands poses additional challenges.

Hands are very dexterous and their pose space has a much higher dimensionality, while segmentation of hands and fingers is difficult due to great similarity. A tracked hand in action might result in unrealistic (self-)collisions and (self-)intersections, while occlusions cause increased ambiguities. Such occlusions might be caused not only due to the (potentially single) camera view, but also due to the hand pose itself. Moreover, hands often exhibit fast motion that causes motion blur and violates temporal continuity, while they often take up only a small part in images or videos, resulting in increased measurement noise and ambiguities due to lack of constraints. Obtaining ground-truth for hand joints is also a much more tedious and time consuming process because of the number of joints involved, the high accuracy needed and the high ambiguities even for human annotators.

Furthermore, the ideal camera setup to capture hand motion is a single camera for marker-less observations. Additional instrumentation like wearable cameras [Kim et al., 2012], markers [Zhao et al., 2012], data-gloves [Dipietro et al., 2008] or color-gloves [Wang and Popović, 2009] simplify the problem, however this intrusive way causes discomfort, alters the hand shape and distorts its pose space by hindering its motion. Commodity RGB-D cameras fulfill the above criteria, however, they still have notable measurement noise and average frame rate.

The problem gets much more complicated for hands interacting with other hands or objects. The pose space is then of higher dimensionality, while ambiguities drastically increase due to occlusions from the object, or due to collisions and intersections that are much more frequent and intense.

2.3 Related Work

The challenges described in Section 2.2 explain why the problem of tracking the 3d motion of hands is still an open problem. In that respect, in this section we summarize the related work in the literature both for hand tracking and pose estimation in general, as well as for the more focused topic of hand-object interaction.

2.3.1 Generative Methods

One of the first hand tracking approaches was [Rehg and Kanade, 1994] that introduced the use of local optimization in the field. Several filtering approaches have been presented [MacCormick and Isard, 2000, Stenger et al., 2001, Wu et al., 2001,

Bray et al., 2007], while also belief-propagation proved to be suitable for articulated objects [Hamer et al., 2010, 2009, Sudderth et al., 2004]. Oikonomidis et al. [2011a] employ Particle Swarm Optimization (PSO) as a form of stochastic search, while later they present a novel evolutionary algorithm that capitalizes on quasi-random sampling [Oikonomidis et al., 2014]. Kim et al. [2012] and Wang and Popović [2009] use inverse-kinematics, while Heap and Hogg [1996] and Wu et al. [2001] reduce the search space using linear subspaces. Athitsos and Sclaroff [2003] resort to probabilistic line matching, while Thayananthan et al. [2003] combine Bayesian filtering with Chamfer matching. Recently, Schmidt et al. [2014] extended the popular Signed Distance Function (SDF) representation to articulated objects, while Qian et al. [2014] combine a gradient based ICP approach with PSO, showing the complementary nature of the two approaches. Sridhar et al. [2013] explore the use of a Sum of Gaussians (SoG) model for hand tracking on RGB images, which is later replaced by a Sum of Anisotropic Gaussians [Sridhar et al., 2014]. In a different fashion, [Melax et al., 2013] present an approach capitalizing on a physics solver.

All these approaches have in common that they are *generative* models. They use an explicit model to generate pose hypotheses, which are evaluated against the observed data. The evaluation is based on an objective function which implicitly measures the likelihood by computing the discrepancy between the pose estimate (hypothesis) and the observed data in terms of an error metric. To keep the problem tractable, each iteration is initialized by the pose estimate of the previous step, relying thus heavily on temporal continuity and being prone to accumulative error. The objective function is evaluated in the high-dimensional, continuous parameter space. Recent approaches relax the assumption of a fixed predefined shape model, either by allowing online non-rigid shape deformation [Taylor et al., 2014] or by personalizing a hand shape model learned from a large population [Khamis et al., 2015], enabling in this way better data fitting and user-specific adaptation.

2.3.2 Discriminative Methods

Discriminative methods learn a direct mapping from the observed image features to the discrete [Athitsos and Sclaroff, 2003, Romero et al., 2009, 2010, Rogez et al., 2014, 2015a,b] or continuous [de Campos and Murray, 2006, Rosales et al., 2001, Tang et al., 2015] target parameter space. Some approaches also segment the body parts first and estimate the pose in a second step [Tompson et al., 2014, Keskin et al., 2012]. Most methods operate on a single frame, being thus immune to pose-drifting due to error accumulation. Generalization in terms of capturing illumination, articulation and view-point variation can be realized only through adequate representative training data. Acquisition and annotation of realistic training data is though a cumbersome and costly procedure. For this reason most approaches rely on synthetic rendered data [Keskin et al., 2012, Romero et al., 2010] that has inherent ground-truth, though recently [Oberweger et al., 2016] presented a semi-automatic way to automatically choose a minimal set of frames for annotation which is then propagated for all frames with global optimization. Special care is needed to avoid over-fitting to the training set,

while the discrepancy between realistic and synthetic data is an important limiting factor. Recent approaches [Tang et al., 2013] tried to address the latter using transductive regression forests to transfer knowledge from fully labeled synthetic data to partially labeled realistic data. Finally, the accuracy of discriminative methods heavily depends on the invariance, repeatability and discriminative properties of the features employed and is lower in comparison to generative methods.

2.3.3 Hybrid Methods

In a *hybrid* approach a discriminative method can effectively complement a generative method, either in terms of initialization or recovery, driving the optimization framework away from local minima in the search space and aiding convergence to the global minimum.

[Ballan et al., 2012] present a tracking system that combines a generative model with a discriminatively trained fingernail detector in RGB images. Sridhar et al. [2013] combine in a real time system a Sum of Gaussians (SoG) generative model with a discriminatively trained fingertip detector in depth images using a linear SVM classifier. The fusion of fingertip detections gained popularity among monocular RGB-D trackers [Tagliasacchi et al., 2015, Taylor et al., 2016, Tzionas et al., 2016].

Alternatively, the model can also be combined with a part classifier based on random forests [Sridhar et al., 2015, 2016]. Recently, Sharp et al. [2015] combined a PSO optimizer with a robust, two-stage regression re-initializer that predicts a distribution over hand poses from a single RGB-D frame.

2.3.4 Hand Models

As mentioned in Section 2.3.1 generative approaches use an explicit hand model to generate pose hypotheses. Different generative methods use different hand models.

A popular approach is to approximate the hand with shape primitives as in Figure 2.1a for easier and faster evaluation of distances [Oikonomidis et al., 2011a,b, 2012, 2014, Tagliasacchi et al., 2015, Qian et al., 2014]. Each shape-primitive of such a model is then voxelized by [Schmidt et al., 2014] and a Signed Distance Function (SDF) is computed for the local coordinate frame as shown in Figure 2.1b. An alternative is the Sum-of-Gaussians model shown in Figure 2.1c [Sridhar et al., 2013, 2014, 2015]. However these approaches are only a rough approximation of the hand. For increased accuracy a triangular mesh as in Figure 2.1d can be used to better fit the image data [Ballan et al., 2012, Tzionas et al., 2016], an approach that is also adopted in this thesis. The piece-wise planar surface of the mesh though might cause problems for derivative-driven optimization approaches. For this reason some works like [Taylor et al., 2016] generate a smooth Loop subdivision surface [Loop, 1987] for a control mesh as in Figure 2.1e that facilitates elegant computation of accurate derivatives.

The need for tricky implementations for accurate derivatives is removed by the approach of [Oberweger et al., 2015]. Instead of using a single Convolutional Neural Network (CNN) to predict hand joints from the input image as [Tompson et al., 2014],

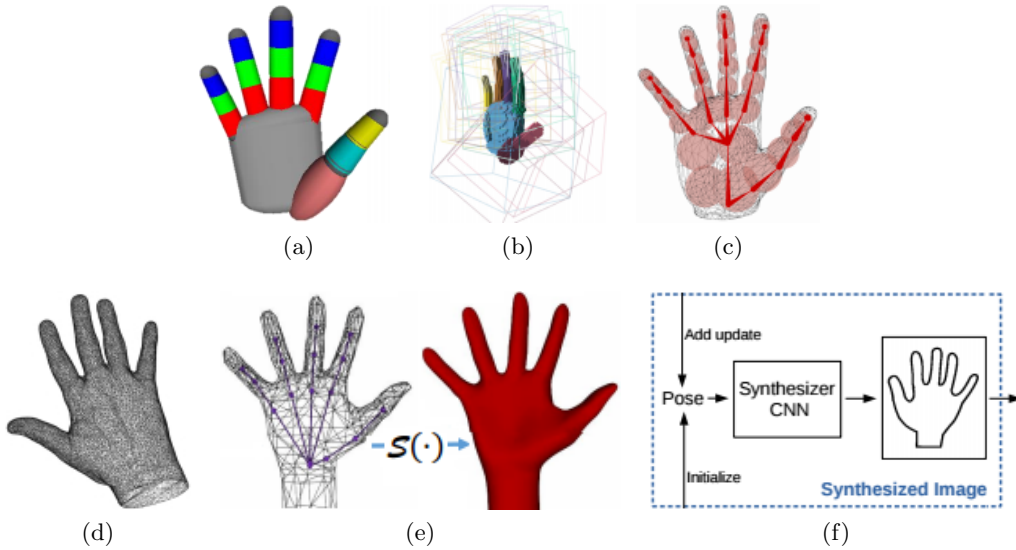


Figure 2.1: Hand models used by several generative or hybrid approaches. (a) *Shape Primitives* approximation of [Oikonomidis et al., 2011a], (b) *Articulated TSDF* for a voxelized shape-primitive hand model of [Schmidt et al., 2014], (c) *Sum-of-Gaussians* model of [Sridhar et al., 2013], (d) *Triangular Mesh* used by [Ballan et al., 2012] and in Chapter 4, (e) *Loop Subdivision Surface* of a triangular control mesh used in [Khamis et al., 2015], (f) *Learned Model* of [Oberweger et al., 2015] using a CNN to synthesize images of a given hand pose.

they train a second CNN to close the loop from output to input and refine the predicted pose. This second CNN shown in Figure 2.1f is an image synthesizer inspired by [Dosovitskiy et al., 2015] that renders a hand image given the hand pose that is predicted by the first CNN from the input depth image. In that respect the pose is refined until the discrepancy between the input and the rendered image is minimized. Interestingly, this essentially results in an “analysis-by-synthesis” generative model with strong discriminative models as building components. Furthermore, gradients are obtained easily with the standard back-propagation rules for CNN networks.

Personalized hand models

An important aspect for tracking is also the use of a personalized model for each user, as used by [Ballan et al., 2012] and in this thesis, by reconstructing a detailed 3d mesh model of a hand with a multiview-stereo method. However this approach is costly and time-consuming, while it is not scalable for many users.

In that respect [Taylor et al., 2014] personalize a template hand mesh by adapting its shape to fit the depth data of a user’s hand. To this end the approach performs simultaneous Levenberg-Marquardt optimization over both correspondences and model parameters across a smooth Loop subdivision surface with As-Rigid-As-Possible (ARAP) regularizers [Sorkine and Alexa, 2007], while initial correspondences are automatically inferred by a regression forest directly from the input image. However,

fitting is done only for a single user, while long calibration sequences are needed in an offline step.

On the contrary, [Khamis et al., 2015] parameterize an entire population of individuals by learning a compact and efficient model of the surface deformation of human hands, while using shorter calibration sequences. Instead of attempting to separately build complete scans per subject and perform Principal Component Analysis (PCA) [Anguelov et al., 2005] the approach fits jointly all noisy depth images available. The output of the method is still a standard subdivision surface as in [Taylor et al., 2014], while the learned model simultaneously accounts for variation in subject-specific shape as well as shape-agnostic poses.

Surprisingly, the positive effect of a detailed personalized hand model on tracking accuracy was shown only later by [Tan et al., 2016] that employ the model of [Khamis et al., 2015] in a quick and easy calibration step and the approach of [Sharp et al., 2015] for tracking. However the highly desirable online and fully automatic model personalization without a calibration step is still an open problem.

In a similar fashion, [Taylor et al., 2016] present impressive tracking performance by employing the aforementioned hand model of [Tan et al., 2016, Khamis et al., 2015] along with fingertip detection based on [Qian et al., 2014] and a gradient based tracker based on smooth Loop subdivision surfaces [Loop, 1987]. The approach performs joint optimization over both correspondences and model pose parameters, while local minimas are avoided with multiple starting poses as a form of global search.

2.3.5 Visual Cues

Generative and discriminative methods have used various low level image cues for hand tracking that are often combined, namely silhouettes [Nirei et al., 1996], edges [Heap and Hogg, 1996], shading [de La Gorce et al., 2011], color [Oikonomidis et al., 2011a,b, 2012], optical flow [Nirei et al., 1996, Lu et al., 2003, de La Gorce et al., 2011, Ballan et al., 2012], or a combination of them [Lu et al., 2003]. Depth [Delamarre and Faugeras, 2001, Bray et al., 2007, Hamer et al., 2009] has recently gained popularity with the ubiquity of RGB-D sensors [Oikonomidis et al., 2011a, Supančič III et al., 2015, Taylor et al., 2016]. Furthermore, contact points between the hand and a manipulated object have emerged as a topic-specific form of cues as described later in Section 2.3.7.

2.3.6 Hands in Action

Due to the challenges described in Section 2.2, the research from the first efforts in the field [Heap and Hogg, 1996] even until very recent approaches [Tompson et al., 2014, Sharp et al., 2015, Supančič III et al., 2015] has mainly focused on a single isolated hand. While isolated hands are indeed useful for a few applications like gesture control, humans use hands mainly for interacting with the environment and manipulating objects.

In this thesis we focus therefore on *hands in action*, i.e. hands that interact with

other hands or objects. This problem has been addressed so far only by a few works, either with a multicamera RGB camera system for less occlusions and ambiguities [Oikonomidis et al., 2011b, Ballan et al., 2012, Wang et al., 2013] or with a monocular RGB-D camera [Hamer et al., 2009, 2010, Romero et al., 2010, Oikonomidis et al., 2012, Kyriazis and Argyros, 2013, 2014, Rogez et al., 2014, 2015a,b, Panteleris et al., 2015, Tzionas and Gall, 2015, Tzionas et al., 2016, Sridhar et al., 2016].

Several systems focus only on the hand pose, ignoring the state of the object. Some of them perform grasping pose classification for a hand either from a front-viewing camera [Romero et al., 2010] or from an egocentric view [Rogez et al., 2014, 2015a,b]. In a different fashion [Hamer et al., 2009] track the hand by considering objects only as occluders, while [Hamer et al., 2010] derive a pose prior from the manipulated objects to support hand tracking. This approach, however, assumes that training data is available to learn the prior.

On the contrary, several efforts try to model the low level interactions between hands and objects. In that respect it is important to detect and penalize the collisions between the tracked hand and object. A common approach is to approximate the hand by spheres. This approach is adopted by [Oikonomidis et al., 2011b, 2012, Panteleris et al., 2015] within a real-time Particle Swarm Optimization (PSO) framework. In the same framework [Kyriazis and Argyros, 2013, 2014] enrich the set of particles by using a physical simulation for hypothesizing the state of one or several rigid objects, similarly to [Wang et al., 2013]. On the other hand [Ballan et al., 2012, Tzionas et al., 2016] avoid an additional explicit model with spheres by using the mesh itself as a collision model. Although all aforementioned approaches assume a known object, [Panteleris et al., 2015] track a hand interacting with an unknown object whose 3d shape is reconstructed on the fly, so that using the partial object shape improves the overall tracking accuracy through collision detection.

Another important aspect is modeling natural contact phenomena between the hand and the object like forces. [Pham et al., 2015] present an approach to infer forces only through camera observations, while [Rogez et al., 2015b] apart from forces try to predict contact points between the hand and the object directly from visual input. Such contact points have recently been used for tracking, computed either by proximity of tracked mesh [Pham et al., 2015, Sridhar et al., 2016] or along with physics simulation [Tzionas et al., 2016] or collision detection [Wang et al., 2013].

2.3.7 Contact Points for Hand-Object Interaction

As described in Section 2.3.6 humans use their hands mainly to interact with the surrounding environment and manipulate the objects in it. An important factor during hand-object interaction is the contact points between the skin of the hand and the surface of the object. Through them humans can sense important properties of the object like its texture, condition (e.g. wet or not) and temperature, or even create a mental map of the object's shape just by haptic exploration and without vision. More importantly though, forces are applied by the hand onto an object during manipulation through the contact points. In that respect, contact points have been recently used

	Method	Hand MoCap	Known Object	Contact Points	
Tracking	Oikonomidis et al. [2011b] Kyriazis and Argyros [2013] Ballan et al. [2012]	✓	✓	✗	
	Pham et al. [2015]	✓	✓	✓	MoCap → Contact points for tracking & force prediction
	Chapter 4 and Tzionas et al. [2016]	✓	✓	✓	MoCap → Contact points for tracking
	Sridhar et al. [2016]	✓	✓	✓	MoCap → Contact points for tracking
In-hand Scanning	Rusinkiewicz et al. [2002] Weise et al. [2008] Weise et al. [2011] Yuheng Ren et al. [2013]	✗	✗	✗	Useful hand information is ignored/rejected
	Panteleris et al. [2015]	✓	✗	✗	MoCap → for Hand/Object segmentation
	Chapter 5 and Tzionas and Gall [2015]	✓	✗	✓	MoCap → Contact points for object reconstruction

Table 2.1: Related literature regarding the use of contact points. The first group of methods focuses on tracking hand-object interaction (Chapter 4), while the second group on in-hand scanning for 3d object reconstruction (Chapter 5).

for tracking and object reconstruction.

Systems that capture the 3d motion of hand-object interaction find the pose of a template model for both the hand and the object. Using the captured pose it is easy to compute contact points between the hand and the object based on a simple proximity test using their vertices. In this respect [Pham et al., 2015, Sridhar et al., 2016] resort to this simple approach and use contact points to stabilize tracking and have more physically plausible poses. In a similar fashion [Tzionas and Gall, 2015] compute contact points between the tracked mesh of a hand and the point cloud of an unknown object and use this information in an in-hand scanning pipeline that reconstructs the object. The last approach is described in detail in Chapter 5.

However contact points computation based on simple distance thresholds might cause instabilities and jitter in transitional periods, i.e. when reaching the object to start manipulation, or when leaving the object after manipulation. For this reason, [Tzionas et al., 2016] first check for object stability using physics simulation by computing the displacement of the object after simulation. In case the object is resting on the scene and is stable, contact points are not used. On the contrary, in case the object is not stable, it is supposed to be under manipulation by the hand and constraints in form of contact points are added to enforce more realistic grasping poses. The approach is described in more detail in Chapter 4.

The related methods for both tracking (Chapter 4) and reconstruction with in-hand scanning (Chapter 5) are summarized in Table 2.1.

2.3.8 Datasets

Even until the beginning of this decade there were no public datasets with ground-truth for 3d hand pose estimation. As identified in the review [Erol et al., 2007a], one

FORTH	[Oikonomidis et al., 2011a]	5 subjects, no hand joints annotations
ETHZ	[Ballan et al., 2012]	Multicamera RGB - One sequence annotated
MPI-GCPR13	[Tzionas and Gall, 2013]	Multicamera RGB
Dexter	[Sridhar et al., 2013]	Multicamera RGB + Monocular RGB-D
ICL	[Tang et al., 2014]	10 subjects
MSRA	[Qian et al., 2014]	6 subjects
MPI-GCPR14	[Tzionas et al., 2014]	Hand-Hand interaction
UCI-EGO	[Rogez et al., 2014]	2 subjects - Hand-Object - Egocentric
NYU	[Tompson et al., 2014]	Automatic ground-truth with the FORTH tracker
FingerPaint Dataset	[Sharp et al., 2015]	5 subjects - Pixel segment. proxy ground-truth
HandNet	[Wetzler et al., 2015]	10 subjects - Fingertip ground-truth (magnetic track.)
GUN-71	[Rogez et al., 2015b]	8 subjects - Hand-Object - Grasp ID ground-truth
A-STAR	[Xu et al., 2016]	15 subjects - Data-glove ground-truth
Hands in Action	[Tzionas et al., 2016]	Hand-Hand / Hand-Object rigid and articulated
Dexter2	[Sridhar et al., 2016]	Hand-Object, only rigid, ground-truth for object-cube
<hr/>		
In-hand-ICCV	[Tzionas and Gall, 2015]	In-Hand scanning dataset
EgoHands	[Bambach et al., 2015]	Hand pixel-wise detection dataset - Egocentric view
Deep-Hand	[Koller et al., 2016]	Hand gesture dataset
NVidia	[Molchanov et al., 2016]	Hand gesture dataset

Table 2.2: Public datasets introduced in this decade, in the order of publication date. Our contributions are highlighted with bold. Unless otherwise noted, the datasets are for hand pose tracking/estimation with RGB-D data, including manual hand joint annotations and containing a single hand of one subject with a front camera view. The last group of datasets is loosely related to hand pose, regarding hand region or gesture detection, but is included for the sake of completeness.

reason for this is the difficulty of acquiring ground truth data. As a result, apart from qualitative results, quantitative evaluation was mostly performed on synthetic data, e.g. in [Rosales et al., 2001, Athitsos and Sclaroff, 2003, Zhou and Huang, 2005, Ballan et al., 2012, Oikonomidis et al., 2012].

The increased research focus of the recent years though brought along several new annotated datasets that facilitate benchmarking and the advancement of the field. Table 2.2 summarizes the datasets of the last years in the order of publication date, while our contributions are highlighted with bold.

The problem of annotated datasets nowadays does not regard only benchmarking, since learning based methods like [Xu and Cheng, 2013, Tang et al., 2013, 2014, Rogez et al., 2014] and especially deep learning ones like [Tompson et al., 2014, Oberweger et al., 2015, Rogez et al., 2015b, Wetzler et al., 2015] need a lot of training and testing data for their deployment. In this direction [Oberweger et al., 2016] presented recently a promising semi-automatic method to minimize the annotation effort by automatically choosing reference frames, for which it automatically infers 3d hand joints from 2d manual user annotations and propagates the results for all the frames with global optimization.

2.4 Summary

This chapter presented an overview of the existing literature regarding hand motion capture. This is a core element of the problems that this thesis focuses on, as presented in Section 1.2.

In this direction in Chapter 3 we present an evaluation protocol for hand pose estimation using frame pairs, while in Chapter 4 we capture the 3d motion of hands in action using videos. Hand tracking works then as a motivation for Chapter 5 to reconstruct the unknown shape of an object by including the hand motion in the reconstruction pipeline.

A Comparison of Directional Distances for Hand Pose Estimation

Benchmarking methods for 3d hand tracking is still an open problem due to the difficulty of acquiring ground truth data. We introduce a new dataset and benchmarking protocol that is insensitive to the accumulative error of other protocols. To this end, we create testing frame pairs of increasing difficulty and measure the pose estimation error separately for each of them. This approach gives new insights and allows to accurately study the performance of each feature or method without employing a full tracking pipeline. Following this protocol, we evaluate various directional distances in the context of silhouette-based 3d hand tracking, expressed as special cases of a generalized Chamfer distance form. An appropriate parameter setup is proposed for each of them, and a comparative study reveals the best performing method in this context.

Contents

3.1	Introduction	17
3.2	Related Work	18
3.3	Hand Pose Estimation	19
3.4	Generalized Chamfer Distance	20
3.5	Benchmark	21
3.6	Experiments	22
3.6.1	Implementation Details	22
3.6.2	Results	23
3.7	Summary	24

3.1 Introduction

Benchmarking methods for 3d hand tracking has been identified in the review [Erol et al., 2007a] as an open problem due to the difficulty of acquiring ground truth data. As in one of the earliest works on marker-less 3d hand tracking [Nirei et al., 1996], quantitative evaluations are still mostly performed on synthetic data, e.g. [Rosales

et al., 2001, Athitsos and Sclaroff, 2003, Zhou and Huang, 2005, Ballan et al., 2012, Oikonomidis et al., 2012]. The vast majority of the related literature, however, is limited to visual, qualitative performance evaluation, where the estimated model is overlaid on the images.

While there are several datasets and evaluation protocols for benchmarking human pose estimation methods publicly available, where markers [Sigal et al., 2010, Van der Aa et al., 2011], inertial sensors [Baak et al., 2010], or a semi-automatic annotation approach [Tenorth et al., 2009] have been used to acquire ground truth data, there are no datasets available for benchmarking articulated hand pose estimation. We propose thus a benchmark dataset consisting of 4 sequences of two interacting hands captured by 8 cameras, where the ground truth position of the 3d joints has been manually annotated.

Tracking approaches are usually evaluated by providing the pose for the first frame and measuring the accumulative pose estimation error for all consecutive frames of the sequence, e.g. [Sigal et al., 2010]. While this protocol is optimal for comparing full tracking systems, it makes it difficult to analyze the impact of individual components of a system. For instance, a method that estimates the joint positions with a high accuracy, but fails in a few cases and is unable to recover from errors, will have a high tracking error if an error occurs very early in a test sequence. However, the tracking error will be very low if the error occurs at the end of the sequence. The accumulation of tracking errors makes it difficult to analyze in-depth situations where an approach works or fails. We therefore propose a benchmark that analyzes the error not over a full sequence, but over a set of pairs consisting of a starting pose and a test frame. Based on the start pose and the test frame, the pairs have different grades of difficulty.

In this chapter, we use the proposed benchmark to analyze various silhouette-based distance measures for hand pose estimation. Distance measures that are based on a closest point distance, like the Chamfer distance, are commonly used due to its efficiency [Nirei et al., 1996] and often extended by including directional information [Gavrila, 1998, Thayananathan et al., 2003]. Recently, a fast method that computes a directional Chamfer distance using a 3d distance tensor has been proposed [Liu et al., 2010] for shape matching. In this chapter, we introduce a general form of the Chamfer distance for hand pose estimation and quantitatively compare several special cases.

3.2 Related Work

Since the earliest days of vision-based hand pose estimation [Rehg and Kanade, 1994, Erol et al., 2007a], low-level features like silhouettes [Nirei et al., 1996], edges [Heap and Hogg, 1996], depth [Delamarre and Faugeras, 2001], optical flow [Nirei et al., 1996], shading [de La Gorce et al., 2011] or a combination of them [Lu et al., 2003] have been used for hand pose estimation. Although Chamfer distances combined with an edge orientation term have been used in [Thayananathan et al., 2003, Athitsos and Sclaroff, 2003, Sudderth et al., 2004, Stenger et al., 2006], the different distances have not been thoroughly evaluated for hand pose estimation. While a kd-tree is

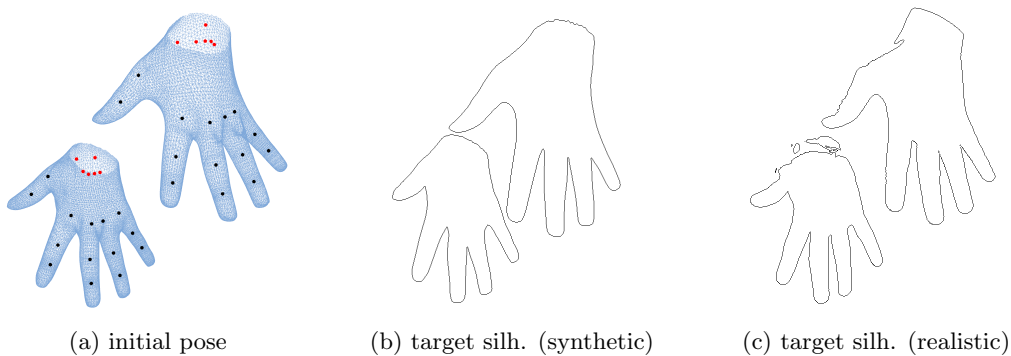


Figure 3.1: *Initial pose* (a) and synthetic (b) and realistic (b) *target silhouettes* of one camera view. The benchmark measures the pose estimation error of the joints of both hands. In the synthetic experiments all joints (*all dots in (a)*) are taken into account, while in the realistic only a subset (*black dots in (a)*) is evaluated.

used in [Sudderth et al., 2004] to compute a directional Chamfer distance, Liu et al. [Liu et al., 2010] recently proposed a distance transform approach to efficiently use a directional Chamfer distance for shape matching. Different methods of shape matching for pose estimation have been compared in the context of rigid objects [Han et al., 2008] or articulated objects [Pons-Moll et al., 2011]. While previous work mainly considered to estimate the pose of a hand in isolation, recent works consider more complicated scenarios where two hands interact with each other [Oikonomidis et al., 2012, Ballan et al., 2012] or with objects [Hamer et al., 2009, Romero et al., 2010, Hamer et al., 2010, Oikonomidis et al., 2011b, Ballan et al., 2012].

3.3 Hand Pose Estimation

For evaluation, we use a publicly available hand model [Ballan et al., 2012], consisting of a set of vertices, an underlying kinematic skeleton with 35 degrees of freedom (DoF) per hand, and skinning weights. The vertices and the joints of the skeleton are shown in Figure 3.1. Each 3d vertex \mathbf{V} is associated to a bone j by the skinning weights $\kappa_{\mathbf{V},j}$, where $\sum_j \kappa_{\mathbf{V},j} = 1$. The articulated deformations of a skeleton are encoded by the vector θ that represents the rigid bone transformations $T_j(\theta)$, i.e. rotation and translation, by twists $\hat{\xi} \in se(3)$ [Murray et al., 1994, Bregler et al., 2004]. Each twist-encoded rigid body transformation $\theta_j \hat{\xi}_j$ for a bone j can be converted into a homogeneous transformation matrix by the exponential map operator, i.e. $T_j(\theta) = \exp(\theta_j \hat{\xi}_j) \in SE(3)$. The mesh deformations based on the pose parameters θ are obtained by the linear blend skinning operator [Lewis et al., 2000] using homogeneous coordinates:

$$\mathbf{V}(\theta) = \sum_j \kappa_{\mathbf{V},j} T_j(\theta) \mathbf{V} . \quad (3.1)$$

In order to estimate the hand pose for a given frame, correspondences between the

mesh and the image of each camera c are established. Each correspondence $(\mathbf{V}_i, \mathbf{q}_i, c_i)$ associates a vertex \mathbf{V}_i to a 2d point \mathbf{q}_i in camera view c_i . Assuming that the cameras are calibrated, the point \mathbf{q}_i can be converted into a projection ray that is represented by the direction \mathbf{d}_i and moment \mathbf{m}_i of the line [Stolfi, 1991, Rosenhahn et al., 2007]. The hand pose can then be determined by the pose parameters that minimize the shortest distance between the 3d vertices \mathbf{V}_i and 3d projection rays $(\mathbf{d}_i, \mathbf{m}_i)$:

$$\arg \min_{\theta} \frac{1}{2N} \sum_{i=1}^N \|\mathbf{V}_i(\theta) \times \mathbf{d}_i - \mathbf{m}_i\|^2. \quad (3.2)$$

This non-linear least-squares problem can be iteratively solved [Rosenhahn et al., 2007]:

- Extract correspondences for all cameras $(\mathbf{V}_i, \mathbf{q}_i, c_i)$,
- Solve (3.2) using the linearization $T_j(\theta) = \exp(\theta_j \hat{\xi}_j) \approx I + \theta_j \hat{\xi}_j$,
- Update vertex positions by (3.1).

In this chapter, we reformulate (3.2) as a Chamfer distance minimization problem.

3.4 Generalized Chamfer Distance

As discussed in Section 3.2, the Chamfer distance is commonly used for shape matching and has been also used for pose estimation by shape matching. In our context, the Chamfer distance between pixels of a contour \mathcal{C} for a given camera view and the set of projected rim vertices $\mathcal{P}(\theta)$, which depend on the pose parameters θ and project onto the contour of the projected surface, is

$$d_{Chamfer}(\theta, \mathcal{C}) = \frac{1}{|\mathcal{P}(\theta)|} \sum_{\mathbf{p} \in \mathcal{P}(\theta)} \min_{\mathbf{q} \in \mathcal{C}} \|\mathbf{p} - \mathbf{q}\|. \quad (3.3)$$

This expression can be efficiently computed using a 2d distance transform [Felzenszwalb and Huttenlocher, 2004].

The Chamfer distance (3.3) can be generalized by

$$d_{Chamfer}^{Z,f,d}(\theta, \mathcal{C}) = \frac{1}{Z} \sum_{\mathbf{p} \in \mathcal{P}(\theta)} f\left(\mathbf{p}, \arg \min_{\mathbf{q} \in \mathcal{C}} d(\mathbf{p}, \mathbf{q})\right), \quad (3.4)$$

where $d(\mathbf{p}, \mathbf{q})$ is a 2d distance function to compute the distance between two points, $f(\mathbf{p}, \mathbf{q})$ is a penalty function for two closest points, and Z is a normalization factor. If we use

$$d(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|, \quad f(\mathbf{p}, \mathbf{q}) = d(\mathbf{p}, \mathbf{q}), \quad Z = |\mathcal{P}(\theta)|, \quad (3.5)$$

$d_{Chamfer}^{Z,f,d}(\theta, \mathcal{C})$ is the standard Chamfer distance (3.3). In order to increase the robustness to outliers, $f(\mathbf{p}, \mathbf{q}) = \min(d(\mathbf{p}, \mathbf{q})^2, \mathcal{K})$ is used in [Stenger et al., 2006], where \mathcal{K} is a threshold on the maximum squared distance.

Orientation can be integrated by penalizing correspondences with inconsistent orientations:

$$d(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|, \quad f(\mathbf{p}, \mathbf{q}) = \begin{cases} d(\mathbf{p}, \mathbf{q}) & \text{if } |\phi(\mathbf{p}) - \phi(\mathbf{q})|_\phi < \tau \\ \mathcal{K} & \text{otherwise} \end{cases}, \quad Z = |\mathcal{P}(\theta)|, \quad (3.6)$$

or by computing the closest distance to points of similar orientation based on a circular distance threshold τ [Gavrila, 1998]:

$$d(\mathbf{p}, \mathbf{q}) = \begin{cases} \|\mathbf{p} - \mathbf{q}\| & \text{if } |\phi(\mathbf{p}) - \phi(\mathbf{q})|_\phi < \tau \\ \infty & \text{otherwise} \end{cases}, \quad f(\mathbf{p}, \mathbf{q}) = d(\mathbf{p}, \mathbf{q}), \quad Z = |\mathcal{P}(\theta)|, \quad (3.7)$$

where $|\phi(\mathbf{p}) - \phi(\mathbf{q})|_\phi$ is the circular distance between two angles, which can be signed, i.e. in the range of $[0, \pi]$, or unsigned, i.e. in the range of $[0, \frac{\pi}{2}]$.

The directional Chamfer distance [Liu et al., 2010] can be written as

$$d(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\| + \lambda |\phi(\mathbf{p}) - \phi(\mathbf{q})|_\phi, \quad f(\mathbf{p}, \mathbf{q}) = d(\mathbf{p}, \mathbf{q}), \quad Z = |\mathcal{P}(\theta)|. \quad (3.8)$$

To compute $d_{Chamfer}^{Z,f,d}(\theta, \mathcal{C})$ with (3.8) efficiently, ϕ can be quantized to compute a 3d distance transform [Liu et al., 2010]. As in [Liu et al., 2010], we compute $\phi(\mathbf{q})$ by converting \mathcal{C} into a line representation [Ramer, 1972]. $\phi(\mathbf{p})$ is obtained by projecting the normals of the corresponding vertices in $\mathcal{P}(\theta)$.

In order to use the generalized Chamfer distance $d_{Chamfer}^{Z,f,d}(\theta, \mathcal{C})$ for pose estimation from multiple views (3.2), only f and Z need to be adapted. Let $\mathcal{C}(c)$ denote the contour of camera view c and $\mathcal{P}(\theta, c)$ the set of projected vertices for pose parameters θ and camera c . (3.2) can be rewritten as

$$\arg \min_{\theta} \frac{1}{2 \sum_c |\mathcal{P}(\theta, c)|} \sum_c d_{Chamfer}^{Z,f,d}(\theta, \mathcal{C}(c)) \quad (3.9)$$

$$\text{with } f(\mathbf{p}, \mathbf{q}) = \|\mathbf{V}(\theta) \times \mathbf{d} - \mathbf{m}\|^2, \quad Z = 1, \quad (3.10)$$

where $\mathbf{V}(\theta)$ is the 3d vertex corresponding to $\mathbf{p} \in \mathcal{P}(\theta)$ and (\mathbf{d}, \mathbf{m}) is the 3d projection ray corresponding to \mathbf{q} . $d(\mathbf{p}, \mathbf{q})$ can be any of the functions (3.5)-(3.8).

In case of (3.6), instead of adding a fixed penalty term \mathcal{K} , correspondences with inconsistent orientation can be simply removed and $\mathcal{P}(\theta, c)$ becomes the set of correspondences with $|\phi(\mathbf{p}) - \phi(\mathbf{q})|_\phi < \tau$.

3.5 Benchmark

We propose a benchmarking protocol that analyzes the error not over full sequences, but over a sampled *set of testing pairs*. Each pair consists of a *starting pose* and a *test frame*, ignoring the intermediate frames to simulate various difficulties. This approach gives new insights and provides means to analyze in-depth the contributions of various features or methods to the overall tracking pipeline under varying difficulty and to thoroughly study failure cases.

In this respect, 4 publicly available sequences¹ are used, containing realistic scenarios of two strongly interacting hands [Ballan et al., 2012]. 10% of the total frames are randomly selected, forming the set of *test frames* of the final pairs. This is the basis to create 4 different sets of image pairs, having 1,5,10,15 frames difference respectively between the *starting pose* and the *test frame*, presenting thus increasing difficulty for tracking systems. These 4 sets and the overall combination constitute a challenging dataset, representing realistic scenarios the occur due to low frame rates, fast motion or estimation errors in the previous frame.

The created testing sets are used in two experimental setups: a purely *synthetic* and a *realistic*. In both cases, the *starting pose* is given by the publicly available motion data outputted by the tracker of [Ballan et al., 2012]. In the *synthetic* experimental setup the *test frame* is synthesized by the hand model and the aforementioned motion data, while the required ground truth exists inherently in them. In the *realistic* setup the *test frame* is given by the camera images, for which no ground-truth data are available, thus the frames have been manually annotated². As error measure, we use the average of the Euclidean distances between the estimated and the ground-truth 3d positions of the joints. For the realistic setup we use only the joints of the model that could be annotated, which are depicted with black color in Figure 3.1. For the synthetic setup all joints of the model (black and red) are taken into account.

3.6 Experiments

3.6.1 Implementation Details

The aforementioned benchmark is used to evaluate four special cases of the generalized Chamfer distance (Section 3.4) for hand pose estimation.

CH denotes the Chamfer distance without any orientation information (3.5).

DCH-Thres rejects correspondences if the orientations are inconsistent, depending on the circular distance threshold τ (3.6).

DCH-Quant computes a 2d distance field for all quantizations of ϕ and assigns a vertex to one bin based on the orientation of its normal (3.7). Instead of hard binning, soft binning can also be performed, denoted by **DCH-Quant2**. In this case, the two closest bins are used, yielding two correspondences per vertex.

DCH-DT3 denotes the approximation of the directional Chamfer distance (3.8) proposed by Liu et al. [Liu et al., 2010]. The approach computes a 3d distance field $DT3$ and depends on two parameters. While λ steers the impact of the orientation term in (3.8), ϕ is quantized by a fixed number of bins.

As mentioned in Section 3.4, the *target silhouette* is approximated with linear line segments for all the directional distances DCH , using [Ramer, 1972]. We also investi-

¹Model, videos, and motion data are provided at <http://cvg.ethz.ch/research/ih-mocap>. Sequences: *Finger tips touching and praying*, *Fingers crossing and twisting*, *Fingers folding*, *Fingers walking*. Video: 1080 × 1920 px, 50 fps, 8 camera-views.

²The ground-truth annotated dataset, along with a viewer-application, is available at http://files.is.tue.mpg.de/dtzionas/GCPR_2013.html.

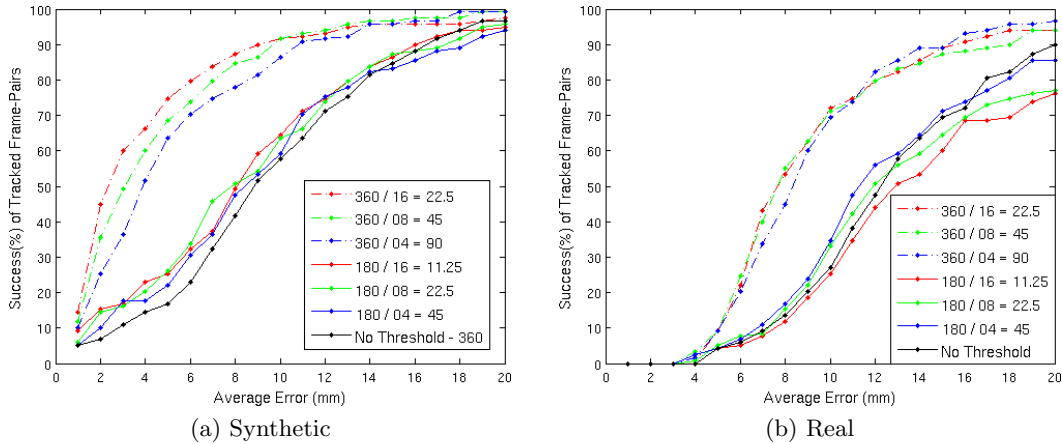


Figure 3.2: Performance evaluation of **DCH-Thres** with different values of τ and both signed (360) and unsigned (180) distance $|\cdot|_\phi$. The plots show the percentage of frame pairs (y -axis) below a given average error (x -axis). The *signed* distance (360) significantly outperforms the *unsigned* distance (180), and the best performing circular distance threshold value is $\tau = 22.5$.

gate two versions of the circular distance $|\cdot|_\phi$, namely the unsigned version, denoted by *180*, and the signed version, denoted by *360*.

3.6.2 Results

We have evaluated all Chamfer distances both on the synthetic and the realistic dataset in order to compare the distances for 3d hand pose estimation, but also in order to investigate the performance predicting abilities of synthetic test data. As measure, we use the average joint error per test frame and compute the percentage of frames with an error below a given threshold. We first evaluated the differences between the signed and unsigned circular distance for *DCH-Thres* and varied the threshold parameter τ . The results are plotted in Figure 3.2. The plot shows that the signed distance outperforms the unsigned distance. Since we observed the same result for *DCH-DT3*, we only report results for the signed distance (360) in the remaining experiments.

For *DCH-DT3*, we evaluated the impact of the two parameters λ and the number of quantization bins for ϕ . The results are plotted in Figure 3.3. Figures 3.3a and 3.3b show the importance of directional information for hand pose estimation, and reveal that there is a large range of λ that works well. With a finer quantization of ϕ , the original directional Chamfer distance (3.8) is better approximated. Figures 3.3c and 3.3d show that 16 bins are sufficient for this task.

We finally evaluated the number of bins for *DCH-Quant* and *DCH-Quant2*. Figure 3.4 shows that *DCH-Quant2* performs better than *DCH-Quant*. In this case, a large number of bins results in a very orientation sensitive measure, and the performance decreases with a finer quantization, in contrast to *DCH-DT3*.

Figure 3.5 summarizes the results for each distance with the best parameter setting.

Table 3.1: Mean error \pm std.dev.(mm) and average time (sec) for 1,5,10,15 frame differences. Time measurements regard single-threaded code on a 6-core 3GHz Xeon PC.

		1	5	10	15	All	Time
Synthetic	<i>CH</i>	1.0 \pm 1.0	2.5 \pm 2.5	4.3 \pm 4.6	6.4 \pm 6.1	3.5 \pm 4.5	103
	<i>DCH-DT3</i>	2.0 \pm 1.3	2.3 \pm 1.3	3.8 \pm 2.9	6.2 \pm 5.8	3.6 \pm 3.8	115
	<i>DCH-Quant</i>	4.0 \pm 1.6	4.2 \pm 1.7	5.4 \pm 2.5	7.0 \pm 4.0	5.1 \pm 2.9	161
	<i>DCH-Thres</i>	1.1 \pm 0.8	1.3 \pm 1.1	2.5 \pm 2.4	4.1 \pm 4.5	2.2 \pm 2.9	077
Realistic	<i>Initial</i>	6.4 \pm 2.0	10.5 \pm 5.6	16.5 \pm 11.5	22.6 \pm 16.9	14.0 \pm 12.3	-
	<i>Ballan et al. [Ballan et al., 2012]</i>	5.9 \pm 1.9	-	-	-	-	-
	<i>CH</i>	7.1 \pm 1.9	7.8 \pm 2.4	9.3 \pm 4.3	10.9 \pm 5.9	8.8 \pm 4.2	-
	<i>DCH-DT3</i>	6.3 \pm 1.5	6.7 \pm 2.0	8.7 \pm 5.1	11.1 \pm 7.9	8.3 \pm 5.4	-
	<i>DCH-Quant</i>	6.8 \pm 1.6	7.2 \pm 2.1	9.0 \pm 4.4	10.7 \pm 7.3	8.4 \pm 4.7	-
	<i>DCH-Thres</i>	6.1 \pm 1.3	6.4 \pm 1.8	7.6 \pm 3.3	9.4 \pm 5.3	7.4 \pm 3.6	-

As expected, the results show that directional information improves the estimation accuracy. However, it is not *DCH-DT3* that performs best for hand pose estimation, but *DCH-Thres*, which is also more efficient to compute. While for *DCH-DT3* the full hand model converges smoothly to the final pose, the thresholding yields a better fit to the silhouette after convergence (*see supplementary video*³). Comparing the performances between synthetic and real data, we conclude that synthetic data is a good performance indicator, but might be misleading sometimes. For instance, *CH* performs well on the synthetic data but worst on the real data. This is also reflected by the mean error for the various frame differences provided in Table 3.1, that introduce an increasing difficulty in the benchmark. Denoted with the term *initial* is the average 3d distance of the joints before running the pose estimation algorithm. The result of a full tracking system [Ballan et al., 2012] is provided for comparison, which expectedly performs better due to the number of features combined. Finally, runtime is provided for the synthetic experiments to indicate the time efficiency of each method.

3.7 Summary

In this chapter we propose a new benchmark dataset and protocol for hand pose estimation using frame pairs. As an example, we discuss a generalized Chamfer distance and evaluate four special cases. The experiments reveal that directional information is important and a signed circular distance performs better than an unsigned distance in the case of silhouettes. Interestingly, a distance using a circular threshold outperforms a smooth directional Chamfer distance both in terms of accuracy and runtime. We finally conclude that synthetic data can be a good indicator for the performance, but might be misleading when comparing different methods.

The benchmark protocol presented in this chapter can be used to evaluate the performance of methods or features for the design of a hand tracking system, without the need to employ a full tracking pipeline at this stage. In Chapter 4 we present such a complete tracking pipeline that captures the motion of hands in action, either in isolation or interacting with other hands or objects.

³<http://youtu.be/Cbu3eEc11qk>

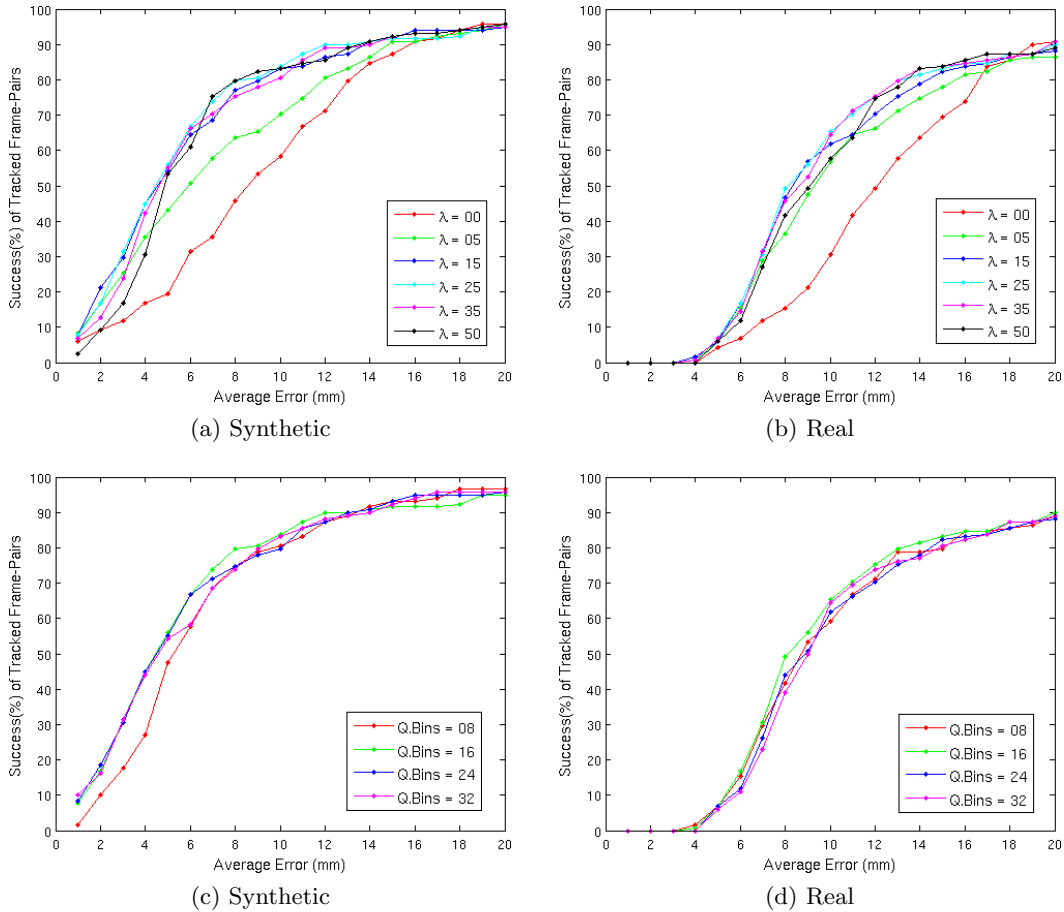


Figure 3.3: **(a-b)** Performance evaluation of **DCH-DT3** with different values of λ , using 16 quantization bins. While the orientation term significantly improves the performance, the performance gets saturated for values in the range $[15,35]$. **(c-d)** Performance evaluation of **DCH-DT3** with different quantizations of ϕ , using $\lambda = 25$. The synthetic data shows that more than 8 bins are required, though the differences are rather small on the real dataset. This is in accordance with Figure 3.2 since a threshold of 22.5 corresponds to 16 quantization bins.

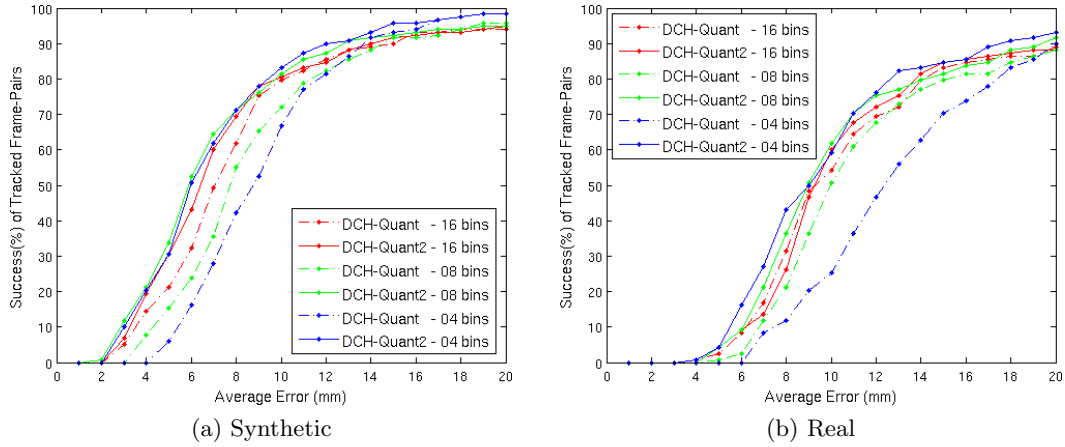


Figure 3.4: Performance evaluation of **DCH-Quant** and **DCH-Quant2** with different quantizations of ϕ . Soft-binning outperforms hard assignments and in this case fewer bins perform better than many bins.

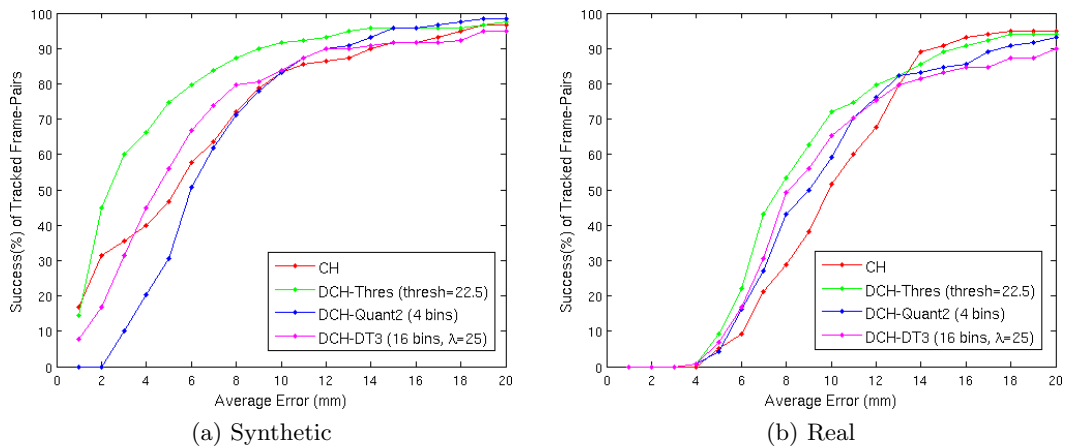


Figure 3.5: Comparison of all distances with best settings. Although *DCH-DT3* provides a smoother distance measure, *DCH-Thres* performs best on both datasets.

Capturing Hands in Action using Discriminative Salient Points and Physics Simulation

In Chapter 3 we presented a benchmark protocol to evaluate the performance of methods or features for hand pose estimation. In this way we can choose the best performing methods and features for the design of a hand tracking pipeline, when the complete tracking pipeline is not yet available.

In this chapter we present instead a complete tracking pipeline that captures the 3d motion of hands in action. Our system performs successful tracking of hands either in isolation or interacting with other hands or objects.

Contents

4.1	Introduction	27
4.2	Pose Estimation	28
4.2.1	Hand and Object Models	29
4.2.2	Objective Function	31
4.2.3	Multicamera RGB	42
4.3	Experimental Evaluation	42
4.3.1	Monocular RGB-D - Hand-Hand Interactions	43
4.3.2	Monocular RGB-D - Hand-Object Interactions	49
4.3.3	Limitations	53
4.3.4	Multicamera RGB	54
4.4	Summary	57

4.1 Introduction

Hand motion capture is a popular research field, recently gaining more attention due to the ubiquity of RGB-D sensors. However, even most recent approaches focus on the case of a single isolated hand. In this chapter, we focus on hands that interact with other hands or objects and present a framework that successfully captures motion in such interaction scenarios for both rigid and articulated objects.



Figure 4.1: Qualitative results of our approach for the case of hand-hand interaction. Each pair shows the aligned RGB and depth input images after depth thresholding along with the pose estimate

Our framework combines a generative model, based on data terms that align the model with the observed data, with discriminatively trained salient points to achieve a low tracking error. By further combining collision detection and physics simulation it achieves better realism and physically plausible estimates even in case of occlusions and missing visual data. Since all components are unified in a single objective function which is almost everywhere differentiable, it can be optimized with standard optimization techniques.

In our experiments we use thus local optimization. Our objective function is then enriched with discriminatively learned salient points to avoid pose estimation errors due to local minima. Salient points, like finger tips, have been used in the earlier work of [Rehg and Kanade, 1994]. Differently from their scenario, however, these salient points cannot be tracked continuously due to the huge amount of occlusions and the similarity in appearance of these features. Therefore we cannot rely on having a fixed association between the salient points and the respective fingers. To cope with this, we propose a novel approach that solves the salient point association jointly with the hand pose estimation problem.

The present chapter unifies the pose estimation for multiple synchronized RGB cameras [Ballan et al., 2012] and a monocular RGB-D camera. In the experiments, we qualitatively and quantitatively evaluate our approach on 29 RGB or RGB-D sequences with a large variety of interactions and up to 150 degrees of freedom. Furthermore, we present for the first time successful tracking results of two hands strongly interacting with non-rigid objects.

4.2 Pose Estimation

Our approach for capturing the motion of hands and manipulated objects can be applied to RGB-D and multi-view RGB sequences. In both cases hands and objects are modeled in the same way as described in Section 4.2.1. The main difference between RGB-D and RGB sequences is the used data term, which depends on depth or edges and optical flow, respectively. We therefore introduce first the objective function for a monocular RGB-D sequence in Section 4.2.2 and describe the differences for RGB sequences in Section 4.2.3.

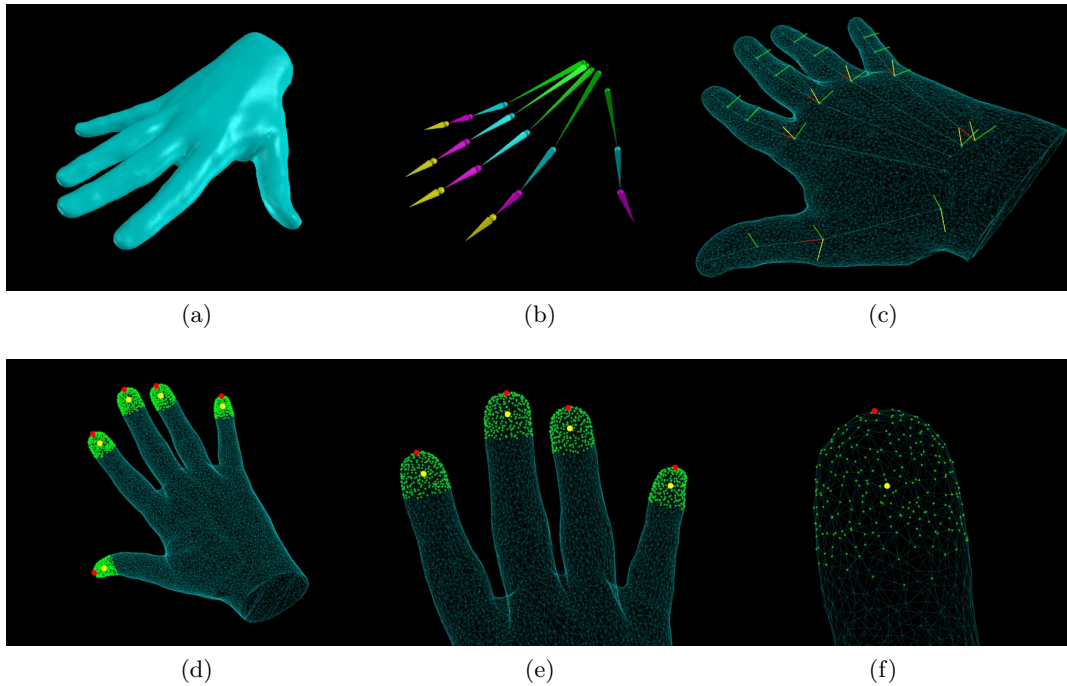


Figure 4.2: Hand model used for tracking. (a) Mesh (b) Kinematic Skeleton (c) Degrees of Freedom (DoF) (d-f) Mesh fingertips (green) used for the salient point detector. The vertices of the fingertips are found based on the manually annotated red vertices. The centroid of the fingertips, as defined in Section 4.2.2.5, is depicted with yellow color

4.2.1 Hand and Object Models

We resort to the popular linear blend skinning (LBS) model [Lewis et al., 2000], consisting of a triangular mesh with an underlying kinematic skeleton, as depicted in Figure 4.2a-c, and a set of skinning weights. In our experiments, a triangular mesh of a pair of hands was obtained by a 3d scanning solution, while meshes for several objects (ball, cube, pipe, rope) were created manually with a 3d graphics software. Some objects are shown in Figure 4.3. A skeletal structure defining the kinematic chain was manually defined and fitted into the meshes. The skinning weight $\kappa_{\mathbf{V},j}$ defines the influence of bone j on 3d vertex \mathbf{V} , where $\sum_j \kappa_{\mathbf{V},j} = 1$. Figure 4.4 visualizes the mesh using the largest skinning weight for each vertex as bone association. The deformation of each mesh is driven by its underlying skeleton with pose parameter vector θ through the skinning weights and is expressed by the LBS operator:

$$\mathbf{V}(\theta) = \sum_j \kappa_{\mathbf{V},j} T_j(\theta) T_j(0)^{-1} \mathbf{V}(0) \quad (4.1)$$

where $T_j(0)$ and $\mathbf{V}(0)$ are the bone transformations and vertex positions at the known rigging pose. The skinning weights are computed using [Baran and Popović, 2007].

The global rigid motion is represented by a 6 DoF twist $\vartheta\xi = \vartheta(u_1, u_2, u_3, \omega_1, \omega_2, \omega_3)$

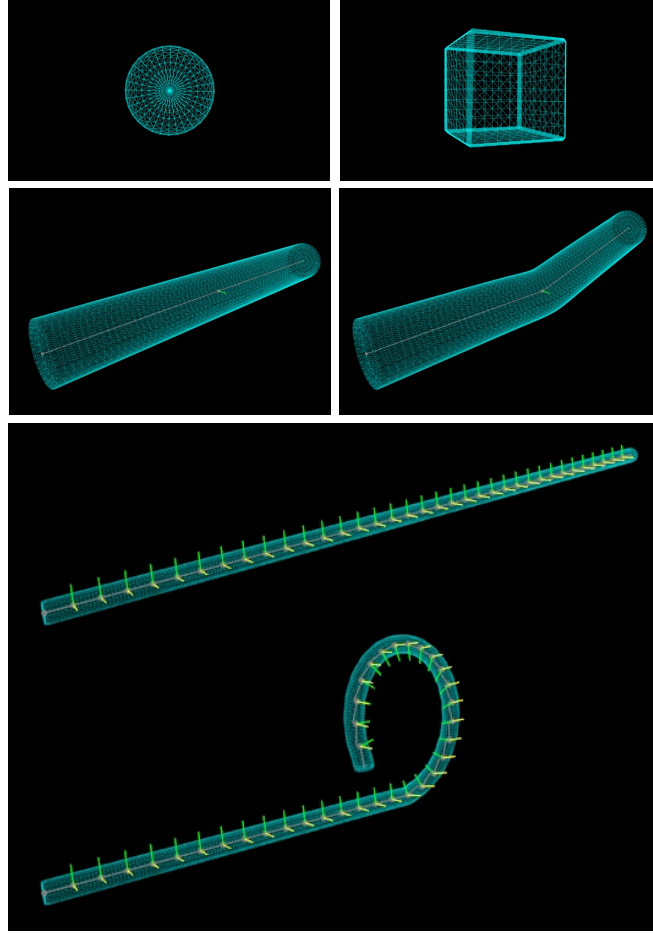


Figure 4.3: Object models used for tracking and their DoF: (top-left) a rigid ball with 6 DoF; (top-right) a rigid cube with 6 DoF; (middle) a pipe with 1 revolute joint, i.e. 7 DoF; (bottom) a rope with 70 revolute joints, i.e. 76 DoF

with $\|\omega\| = 1$ [Bregler et al., 2004, Murray et al., 1994, Pons-Moll and Rosenhahn, 2011]. The twist action $\vartheta \hat{\xi} \in se(3)$ has the form of a 4×4 matrix

$$\vartheta \hat{\xi} = \vartheta \begin{pmatrix} \hat{\omega} & u \\ 0_{1 \times 3} & 0 \end{pmatrix} = \vartheta \begin{pmatrix} 0 & -\omega_3 & \omega_2 & u_1 \\ \omega_3 & 0 & -\omega_1 & u_2 \\ -\omega_2 & \omega_1 & 0 & u_3 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.2)$$

and the exponential map operator $\exp(\vartheta \hat{\xi})$ defines the group action:

$$T(\vartheta \hat{\xi}) = \begin{pmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{pmatrix} = \exp(\vartheta \hat{\xi}) \in SE(3). \quad (4.3)$$

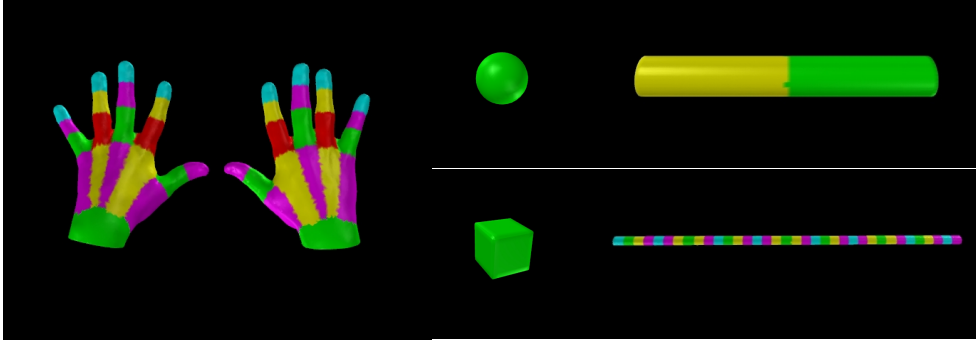


Figure 4.4: Segmentation of the meshes based on the skinning weights. The ball and the cube are rigid objects while the pipe and rope are modeled as articulated objects. Each hand has 20 skinning bones, the pipe has 2, while the rope has 36

While $\theta = \vartheta\xi$ for a rigid object, articulated objects have additional parameters. We model the joints by revolute joints. A joint with one DoF is modeled by a single revolute joint, i.e. the transformation of the corresponding bone j is given by $\exp(\vartheta_{p(j)}\xi_{p(j)})\exp(\vartheta_j\xi_j)$ where $p(j)$ denotes the parent bone. If a bone does not have a parent bone, it is the global rigid transformation. The transformation of an object with one revolute joint is thus described by $\theta = (\vartheta\xi, \vartheta_1)$. Joints with two or three DoF are modeled by a combination of K_j revolute joints, i.e. $\prod_{k=1}^{K_j}\exp(\vartheta_{j,k}\xi_{j,k})$. For simplicity, we denote the relative transformation of a bone j by $\hat{T}_j(\theta) = \prod_{k=1}^{K_j}\exp(\vartheta_{j,k}\xi_{j,k})$. The global transformation of a bone j is then recursively defined by

$$T_j(\theta) = T_{p(j)}(\theta)\hat{T}_j(\theta). \quad (4.4)$$

In our experiments, a single hand consists of 31 revolute joints, i.e. 37 DoF, as shown in Figure 4.2c. The rigid objects have 6 DoF. The deformations of the non-rigid shapes shown in Figure 4.3 are approximated by a skeleton. The pipe has 1 revolute joint, i.e. 7 DoF, while the rope has 70 revolute joints, i.e. 76 DoF. Thus, for sequences with two interacting hands we have to estimate all 74 DoF and together with the rope 150 DoF.

4.2.2 Objective Function

Our objective function for pose estimation consists of seven terms:

$$\begin{aligned} E(\theta, D) = & E_{model \rightarrow data}(\theta, D) + E_{data \rightarrow model}(\theta, D) + \\ & \gamma_c E_{collision}(\theta) + E_{salient}(\theta, D) + \\ & \gamma_{ph} E_{physics}(\theta) + \gamma_a E_{anatomy}(\theta) + \\ & \gamma_r E_{regularization}(\theta) \end{aligned} \quad (4.5)$$

where θ are the pose parameters of the template meshes and D is the current preprocessed depth image. The preprocessing is explained in Section 4.2.2.1. The first two terms minimize the alignment error of the transformed mesh and the depth data. The alignment error is measured by $E_{model \rightarrow data}$, which measures how well the model fits the observed depth data, and $E_{data \rightarrow model}$, which measures how well the depth data is explained by the model. $E_{salient}$ measures the consistency of the generative model with detected salient points in the image. The main purpose of the term in our framework is to recover from tracking errors of the generative model. $E_{collision}$ penalizes intersections of fingers and $E_{physics}$ enhances the realism of grasping poses during interaction with objects. Both of the terms $E_{collision}$ and $E_{physics}$ ensure physically plausible poses and are complementary. The term $E_{anatomy}$ enforces anatomically inspired joint limits, while the last term is a simple regularization term that prefers the solution of the previous frame if there are insufficient observations to determine the pose.

In the following, we give details for the terms of the objective function (4.5) as well as the optimization of it.

4.2.2.1 Preprocessing

For pose estimation, we first remove irrelevant parts of the RGB-D image by thresholding the depth values, in order to avoid unnecessary processing like normal computation for points far away. Segmentation of the hand from the arm is not necessary and is therefore not performed. Subsequently we apply skin color segmentation on the RGB image [Jones and Rehg, 2002]. As a result we get masked RGB-D images, denoted as D in (4.5). The skin color segmentation separates hands and non-skin colored objects, facilitating hand and object tracking accordingly.

4.2.2.2 Fitting the model to the data - LO_{m2d}

The first term in Equation (4.5) aims at fitting the mesh parameterized by pose parameters θ to the preprocessed data D . To this end, the depth values are converted into a 3d point cloud based on the calibration data of the sensor. The point cloud is then smoothed by a bilateral filter [Paris and Durand, 2009] and normals are computed [Holzer et al., 2012]. For each visible vertex of the model $\mathbf{V}_i(\theta)$, with normal $\mathbf{n}_i(\theta)$, we search for the closest point X_i in the point cloud. This gives a 3d-3d correspondence for each vertex. We discard the correspondence if the angle between the normals of the vertex and the closest point is larger than 45° or the distance between the points is larger than 10 mm. We can then write the term $E_{model \rightarrow data}$ as a least squared error of *point-to-point* distances:

$$E_{model \rightarrow data}(\theta, D) = \sum_i \|\mathbf{V}_i(\theta) - X_i\|^2 \quad (4.6)$$

An alternative to the *point-to-point* distance is the *point-to-plane* distance, which is commonly used for 3d reconstruction [Chen and Medioni, 1991, Rusinkiewicz and

Levoy, 2001, Rusinkiewicz et al., 2002]. In this case:

$$E_{model \rightarrow data}(\theta, D) = \sum_i \|\mathbf{n}_i(\theta)^T(\mathbf{V}_i(\theta) - X_i)\|^2. \quad (4.7)$$

The two distance metrics are evaluated in Section 4.3.1.1.

4.2.2.3 Fitting the data to the model - LO_{d2m}

Only fitting the model to the data is not sufficient as we will show in our experiments. In particular, poses with self-occlusions can have a very low error since the measure only evaluates how well the visible part of the model fits the point cloud. The second term $E_{data \rightarrow model}(\theta, D)$ matches the data to the model to make sure that the solution is not degenerate and explains the data as well as possible. However, matching the data to the model is more expensive since after each iteration the pose changes, which would require to update the data structure for matching, e.g. distance fields or kd-trees, after each iteration. We therefore reduce the matching to depth discontinuities [Gall et al., 2011a]. To this end, we extract depth discontinuities from the depth map and the projected depth profile of the model using an edge detector [Canny, 1986]. Correspondences are again established by searching for the closest points, but now in the depth image using a 2d distance transform [Felzenszwalb and Huttenlocher, 2004]. Similar to $E_{model \rightarrow data}(\theta, D)$, we discard correspondences with a large distance. The depth values at the depth discontinuities in D , however, are less reliable not only due to the depth ambiguities between foreground and background, but also due to the noise of consumer sensors. The depth of the point in D is therefore computed as average in a local 3×3 pixels neighborhood and the outlier distance threshold is increased to 30 mm. The approximation is sufficient for discarding outliers, but insufficient for minimization. For each matched point in D we therefore compute the projection ray uniquely expressed as a Plücker line [Pons-Moll and Rosenhahn, 2011, Rosenhahn et al., 2007, Stolfi, 1991] with direction \mathbf{d}_i and moment \mathbf{m}_i and minimize the least square error between the projection ray and the vertex $\mathbf{V}_i(\theta)$ for each correspondence:

$$E_{data \rightarrow model}(\theta, D) = \sum_i \|\mathbf{V}_i(\theta) \times \mathbf{d}_i - \mathbf{m}_i\|^2 \quad (4.8)$$

We compared the matching based on depth discontinuities with a direct matching of the point cloud to the model using a kd-tree. The direct matching increases the runtime by 40% or more without reducing the error.

4.2.2.4 Collision detection - \mathcal{C}

Collision detection is based on the observation that two objects cannot share the same space and is of high importance in case of self-penetration, inter-finger penetration or general intensive interaction, as in the case depicted in Figure 4.5.

Collisions between meshes are detected by efficiently finding the set of colliding triangles \mathcal{C} using bounding volume hierarchies (BVH) [Teschner et al., 2004]. In order

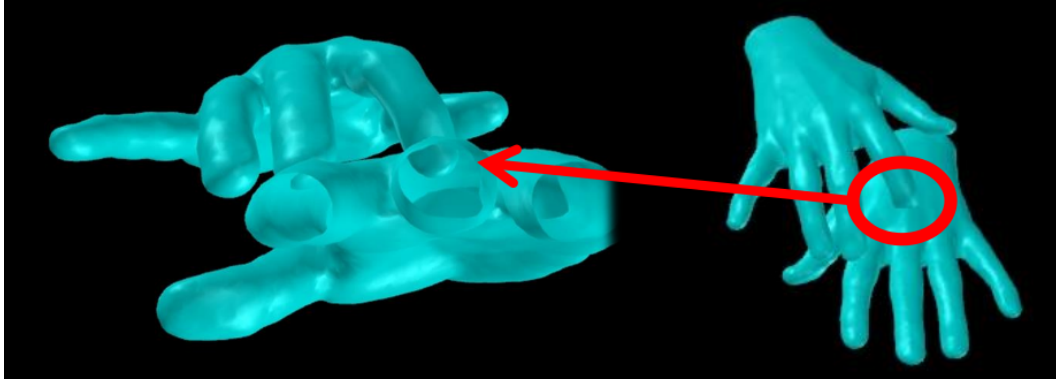


Figure 4.5: “Walking” sequence. Without the collision term unrealistic mesh intersections are observed during interactions

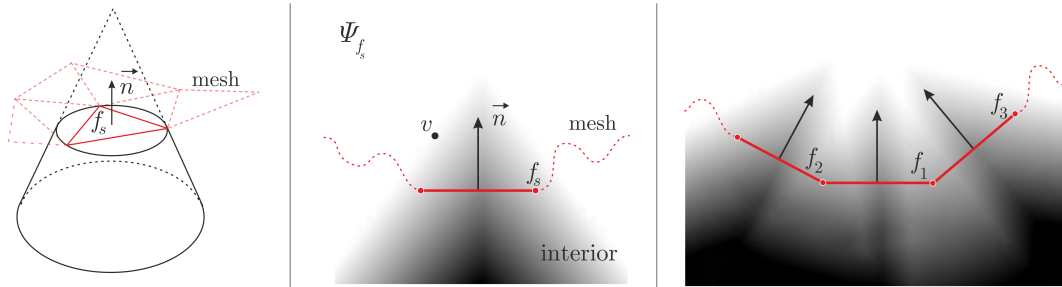


Figure 4.6: (Left) Domain of the distance field Ψ_{f_s} generated by the face f_s . (Middle) Longitudinal section of the distance field Ψ_{f_s} : darker areas correspond to higher penalties. (Right) Distance fields add up in case of multiple collisions

to penalize collisions and penetrations, we avoid using a signed 3d distance field for the whole mesh due to its high computational complexity and the fact that it has to be recomputed at every iteration of the optimization framework. Instead, we resort to a more efficient approach with local 3d distance fields defined by the set of colliding triangles \mathcal{C} that have the form of a cone as depicted in Figure 4.6. In case of multiple collisions the defined conic distance fields are summed up as shown in the same figure. Having found a collision between two triangles f_t and f_s , the amount of penetration can be computed by the position inside the conic distance fields. The value of the distance field represents the intention of the repulsion that is needed to penalize the intrusion.

Let us consider the case where the vertices of f_t are the *intruders* and the triangle f_s is the *receiver* of the penetration. The opposite case is then similar. The cone for computing the 3d distance field $\Psi_{f_s} : \mathbb{R}^3 \rightarrow \mathbb{R}_+$ is defined by the circumcenter of the triangle f_s . Letting $\mathbf{n}_{f_s} \in \mathbb{R}^3$ denote the normal of the triangle, $\mathbf{o}_{f_s} \in \mathbb{R}^3$ the

circumcenter, and $r_{f_s} \in \mathbb{R}_{>0}$ the radius of the circumcircle, we have

$$\Psi_{f_s}(\mathbf{V}_t) = \begin{cases} |(1 - \Phi(\mathbf{V}_t))\Upsilon(\mathbf{n}_{f_s} \cdot (\mathbf{V}_t - \mathbf{o}_{f_s}))|^2 & \Phi(\mathbf{V}_t) < 1 \\ 0 & \Phi(\mathbf{V}_t) \geq 1 \end{cases} \quad (4.9)$$

$$\Phi(\mathbf{V}_t) = \frac{\|(\mathbf{V}_t - \mathbf{o}_{f_s}) - (\mathbf{n}_{f_s} \cdot (\mathbf{V}_t - \mathbf{o}_{f_s}))\mathbf{n}_{f_s}\|}{-\frac{r_{f_s}}{\sigma}(\mathbf{n}_{f_s} \cdot (\mathbf{V}_t - \mathbf{o}_{f_s})) + r_{f_s}} \quad (4.10)$$

$$\Upsilon(x) = \begin{cases} -x + 1 - \sigma & x \leq -\sigma \\ -\frac{1-2\sigma}{4\sigma^2}x^2 - \frac{1}{2\sigma}x + \frac{1}{4}(3 - 2\sigma) & x \in (-\sigma, +\sigma) \\ 0 & x \geq +\sigma. \end{cases} \quad (4.11)$$

The term Φ projects the vertex \mathbf{V} onto the axis of the right circular cone defined by the triangle normal \mathbf{n} going through the circumcenter \mathbf{o} and measures the distance to it as illustrated in Figure 4.6. The distance is scaled by the radius of the cone at this point. If $\Phi(\mathbf{V}) < 1$ the vertex is inside the cone and if $\Phi(\mathbf{V}) = 0$ the vertex is on the axis. The term Υ measures how far the projected point is from the circumcenter and defines the intensity of the repulsion. If $\Upsilon < 0$, the projected point is behind the triangle. Within the range $(-\sigma, +\sigma)$, the penalizer term is quadratic with values between zero and one. If the penetration is larger than $|\sigma|$ the penalizer term becomes linear. The parameter σ also defines the field of view of the cone and is fixed to 0.5.

For each vertex penetrating a triangle, a repulsion term in the form of a 3d-3d correspondence that pushes the vertex back is computed. The direction of the repulsion is given by the inverse normal direction of the vertex and its intensity by Ψ . Using *point-to-point* distances, the repulsion correspondences are computed for the set of colliding triangles \mathcal{C} :

$$E_{collision}(\theta) = \sum_{(f_s(\theta), f_t(\theta)) \in \mathcal{C}} \left\{ \sum_{\mathbf{V}_s \in f_s} \|\Psi_{f_t}(\mathbf{V}_s)\mathbf{n}_s\|^2 + \sum_{\mathbf{V}_t \in f_t} \|\Psi_{f_s}(\mathbf{V}_t)\mathbf{n}_t\|^2 \right\} \quad (4.12)$$

Though not explicitly denoted, f_s and f_t depend on θ and therefore also Ψ , \mathbf{V} and \mathbf{n} . For *point-to-plane* distances, the equation gets simplified since $\mathbf{n}^T \mathbf{n} = 1$:

$$E_{collision}(\theta) = \sum_{(f_s(\theta), f_t(\theta)) \in \mathcal{C}} \left\{ \sum_{\mathbf{V}_s \in f_s} \|\Psi_{f_t}(\mathbf{V}_s)\|^2 + \sum_{\mathbf{V}_t \in f_t} \|\Psi_{f_s}(\mathbf{V}_t)\|^2 \right\} \quad (4.13)$$

This term takes part in the objective function (4.5) regulated by weight γ_c . An evaluation of different γ_c values is presented in Section 4.3.1.3.

Table 4.1: The graph contains \mathcal{T} mesh fingertips ϕ_t and \mathcal{S} fingertip detections δ_s . The cost of assigning a detection δ_s to a fingertip ϕ_t is given by w_{st} as shown in table (a). The cost of declaring a detection as false positive is λw_s , where w_s is the detection confidence. The cost of not assigning any detection to fingertip ϕ_t is given by λ . The binary solution of table (b) is constrained to sum up to one for each row and column

(a)		Fingertips ϕ_t				V
		ϕ_1	ϕ_2	\dots	$\phi_{\mathcal{T}}$	α
δ_s	δ_1	w_{11}	w_{12}	\dots	$w_{1\mathcal{T}}$	λw_1
δ_2	δ_2	w_{21}	w_{22}	\dots	$w_{2\mathcal{T}}$	λw_2
δ_3	δ_3	w_{31}	w_{32}	\dots	$w_{3\mathcal{T}}$	λw_3
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$\delta_{\mathcal{S}}$	$\delta_{\mathcal{S}}$	$w_{\mathcal{S}1}$	$w_{\mathcal{S}2}$	\dots	$w_{\mathcal{S}\mathcal{T}}$	$\lambda w_{\mathcal{S}}$
V	β	λ	λ	\dots	λ	∞

(b)		Fingertips ϕ_t				V
		ϕ_1	ϕ_2	\dots	$\phi_{\mathcal{T}}$	α
δ_s	δ_1	e_{11}	e_{12}	\dots	$e_{1\mathcal{T}}$	α_1
δ_2	δ_2	e_{21}	e_{22}	\dots	$e_{2\mathcal{T}}$	α_2
δ_3	δ_3	e_{31}	e_{32}	\dots	$e_{3\mathcal{T}}$	α_3
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$\delta_{\mathcal{S}}$	$\delta_{\mathcal{S}}$	$e_{\mathcal{S}1}$	$e_{\mathcal{S}2}$	\dots	$e_{\mathcal{S}\mathcal{T}}$	$\alpha_{\mathcal{S}}$
V	β	β_1	β_2	\dots	$\beta_{\mathcal{T}}$	0

4.2.2.5 Salient point detection - \mathcal{S}

Our approach is so far based on a generative model, which provides accurate solutions in principle, but recovers only slowly from ambiguities and tracking errors. However, this can be compensated by integrating a discriminatively trained salient point detector into a generative model.

To this end, we train a fingertip detector on raw depth data. We manually annotate¹ the fingertips of 56 sequences consisting of approximately 2000 frames, with 32 of the sequences forming the training and 24 forming the testing set. We use a Hough forest [Gall et al., 2011b] with 10 trees, each trained with 100000 positive and 100000 negative patches. The negative patches are uniformly sampled from the background. The trees have a maximal depth of 25 and a minimum leaf size of 20 samples. Each patch is sized 16×16 and consists of 11 feature channels: 2 channels obtained by a 5×5 min- and max-filtered depth channel and 9 gradient features obtained by 9 HOG bins using a 5×5 cell and soft binning. As for the pool of split functions at a test node, we randomly generate a set of 20000 binary tests. Testing is performed at multiple scales and non-maximum suppression is used to retain the most confident detections that do not overlap by more than 50%.

Since we resort to salient points only for additional robustness, it is usually sufficient to have only sparse fingertip detections. We therefore collect detections with a high confidence, choosing a threshold of $c_{thr} = 3.0$ for our experiments. The association between the \mathcal{T} fingertips ϕ_t of the model depicted in Figure 4.2 (d-f) and the \mathcal{S}

¹All annotated sequences are available at <http://files.is.tue.mpg.de/dtzionas/hand-object-capture.html>

detections δ_s is solved by integer programming [Belongie et al., 2002]:

$$\begin{aligned}
& \arg \min_{e_{st}, \alpha_s, \beta_t} && \sum_{s,t} e_{st} w_{st} + \lambda \sum_s \alpha_s w_s + \lambda \sum_t \beta_t \\
\text{subject to} &&& \sum_s e_{st} + \beta_t = 1 \quad \forall t \in \{1, \dots, \mathcal{T}\} \\
&&& \sum_t e_{st} + \alpha_s = 1 \quad \forall s \in \{1, \dots, \mathcal{S}\} \\
&&& e_{st}, \alpha_s, \beta_t \in \{0, 1\}
\end{aligned} \tag{4.14}$$

As illustrated in Table 4.1, $e_{st} = 1$ defines an assignment of a detection δ_s to a fingertip ϕ_t . The assignment cost is defined by w_{st} . If $\alpha_s = 1$, the detection δ_s is declared as a false positive with cost λw_s and if $\beta_t = 1$, the fingertip ϕ_t is not assigned to any detection with cost λ .

The weights w_{st} are given by the 3d distance between the detection δ_s and the finger of the model ϕ_t . For each finger ϕ_t , a set of vertices are marked in the model. The distance is then computed between the 3d centroid of the visible vertices of ϕ_t (Figure 4.2d-f) and the centroid of the detected region δ_s . The latter is computed based on the 3d point cloud δ'_s corresponding to the detection bounding box. For the weights w_s , we investigate two approaches. The first approach uses $w_s = 1$. The second approach takes the confidences c_s of the detections into account by setting $w_s = \frac{c_s}{c_{thr}}$. The weighting parameter λ is evaluated in Section 4.3.1.2.

If a detection δ_s has been associated to a fingertip ϕ_t , we have to define correspondences between the set of visible vertices of ϕ_t and the detection point cloud δ'_s . If the fingertip ϕ_t is already very close to δ'_s , i.e. $w_{st} < 10mm$, we do not compute any correspondences since the localization accuracy of the detector is not higher. In this case just the close proximity of the fingertip ϕ_t to the data suffices for a good alignment. Otherwise, we compute the closest points between the vertices \mathbf{V}_i and the points X_i of the detection δ'_s as illustrated in Figure 4.7a:

$$E_{salient}(\theta, D) = \sum_{s,t} e_{st} \left(\sum_{(X_i, \mathbf{V}_i) \in \delta'_s \times \phi_t} \|\mathbf{V}_i(\theta) - X_i\|^2 \right) \tag{4.15}$$

As in (4.7), a *point-to-plane* distance metric can replace the *point-to-point* metric. When less than 50% of the vertices of ϕ_t project inside the detection bounding box, we even avoid the additional step of computing correspondences between the vertices and the detection point cloud. Instead we associate all vertices with the centroid of the detection point cloud as shown in Figure 4.7b.

4.2.2.6 Physics Simulation - \mathcal{P}

A phenomenon that frequently occurs in the context of hand-object interaction are physically unrealistic poses due to occlusions or missing data. Such an example is illustrated in Figure 4.8, where a cube is grasped and moved by two fingers. Since

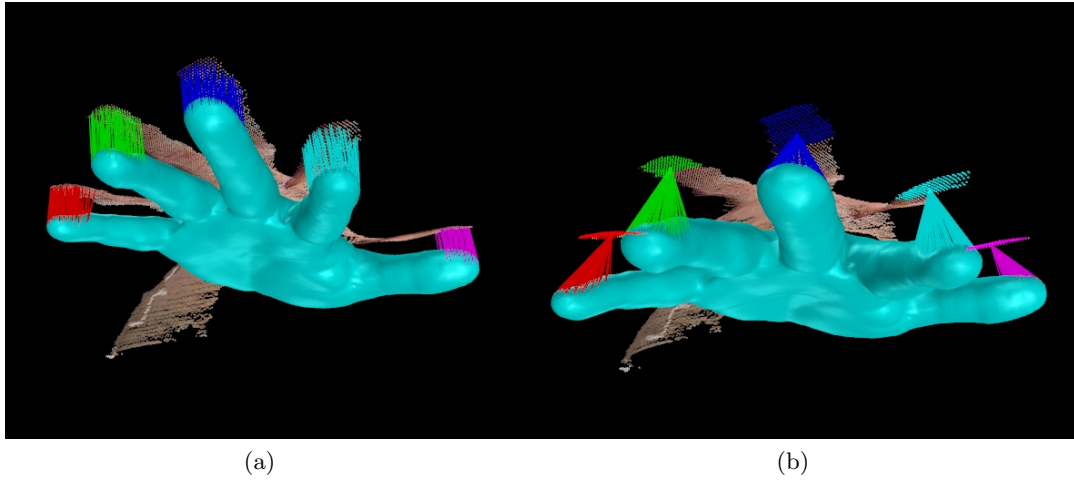


Figure 4.7: Correspondences between the fingertips ϕ_t of the model and (a) the closest points of the associated detections δ'_s (b) the centroids of the associated detections δ'_s

one of the fingers that is in contact with the cube is occluded, the estimated pose is physical unrealistic. Due to gravity, the cube would fall down.

In order to compensate for this during hand-object interaction scenarios, we resort to physics simulation [Coumans, 2013] for additional realism and physical plausibility. To this end, we model the static scene as well and based on gravity and the parameters friction, restitution, and mass for each object we can run a physics simulation. To speed up the simulation, we represent each body or object part defined by the skinning weights as shown in Figure 4.4 as convex hulls. This is visualized in Figure 4.9.

Given current pose estimates of the hands and the manipulated object, we first evaluate if the current solution is physically plausible. To this end, we run the simulation for 35 iterations with a time-step of 0.1 seconds. If the centroid of the object moves by less than 3mm we consider the solution as stable. Otherwise, we have to search for the hand pose which results in a more stable estimate. Since it is intractable to evaluate all possible hand poses, we search only for configurations which require a minor change of the hand pose. This is a reasonable assumption for our tracking scenario. To this end, we first compute the distances between all parts of the fingers, as depicted in Figure 4.10, and the object [Aggarwal et al., 1987, Gärtner and Schönherr, 2000]. Each finger part with distance less than 10mm is then considered as candidate for being in contact with the object and each combination of at least two and maximum four candidate parts is taken into account.

The contribution of each combination to the stability of the object is examined through the physics simulation after rigidly moving the corresponding finger parts towards the closest surface point of the object. Figure 4.9 illustrates the case for a combination of two finger parts. The simulation is repeated for all combinations and we select the combination with the lowest movement of the object, i.e. the smallest displacement of its centroid from the initial position. Based on the selected combina-

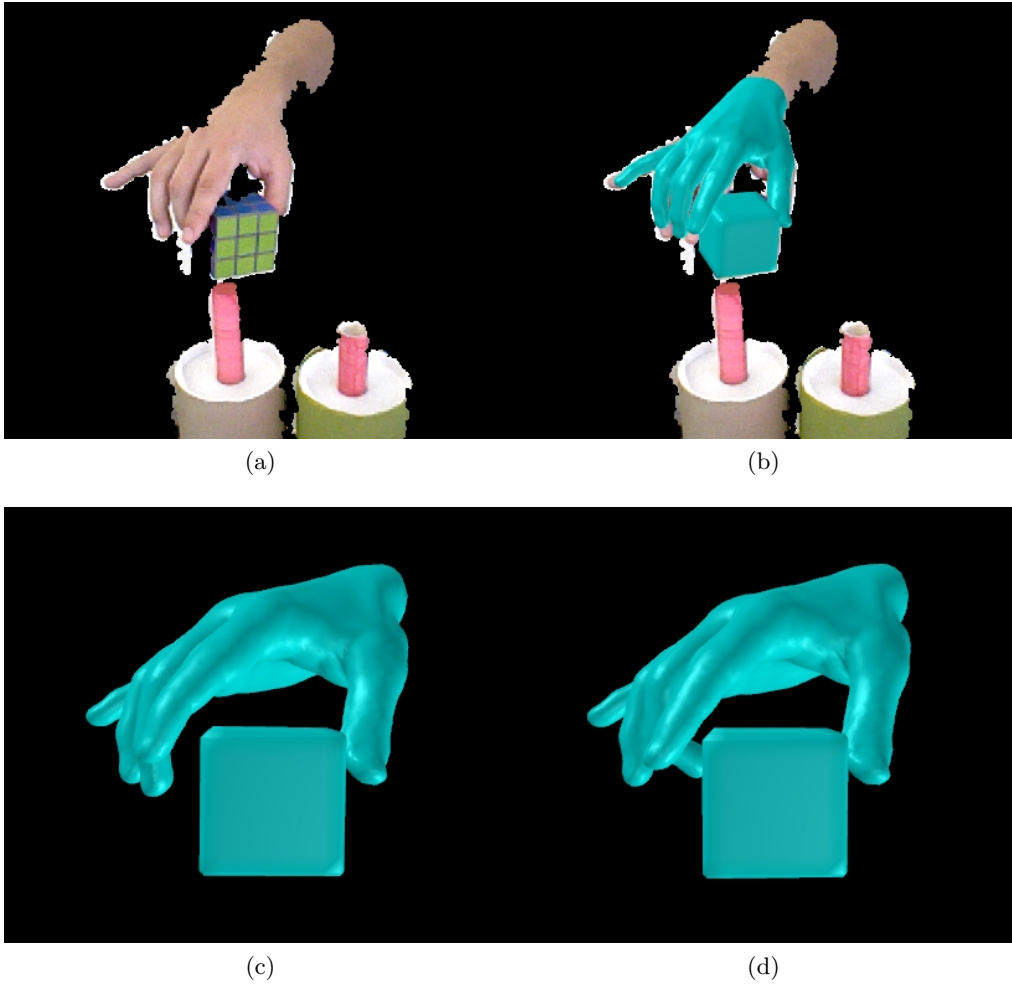


Figure 4.8: Physical plausibility during hand-object interaction. (a) Input RGB-D image. (b-c) Obtained results without the physics component. (d) Obtained results with the physics component, ensuring a more realistic pose during interaction

tion, we define an energy that forces the corresponding finger parts to get in contact with the object by minimizing the closest distance between the parts i and the object:

$$E_{physics}(\theta) = \sum_i \|\mathbf{V}_i(\theta) - X_i\|^2 \quad (4.16)$$

The vertices \mathbf{V}_i and X_i correspond to the closest point of a finger part and the object, respectively. As in (4.7), a *point-to-plane* distance metric can replace the *point-to-point* metric.

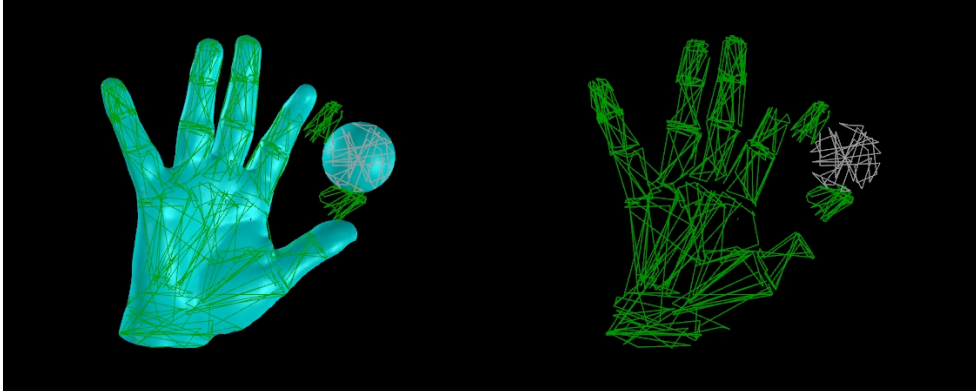


Figure 4.9: Low resolution representation of the hands and objects for the physics simulation. In order to predict the finger parts (green) that give the physically most stable results if they were in contact with the object (white), all combinations of finger parts close to the object are evaluated. The image shows how two finger parts are moved to the object for examining the contribution to the stability of the object. The stability is measured by a physics simulation where all green parts are static

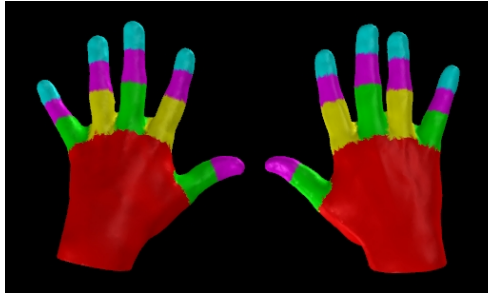


Figure 4.10: Finger parts that form all possible supporting combinations in the physics simulation component. Parts with red color do not take part in this process

4.2.2.7 Anatomical limits

Anatomically inspired joint-angle limits [Albrecht et al., 2003] are enforced as soft constraints by the term:

$$E_{anatomy}(\theta) = \sum_k (\exp(p(l_k - \theta_k)) + \exp(p(\theta_k - u_k))) \quad (4.17)$$

where $p = 10$. The index k goes over all revolute joints and $[u_k, l_k]$ is the allowed range for each of them. The term is illustrated for a single revolute joint in Figure 4.11. We use $\gamma_a = 0.0015 C_{all}$, where C_{all} is the total number of correspondences.

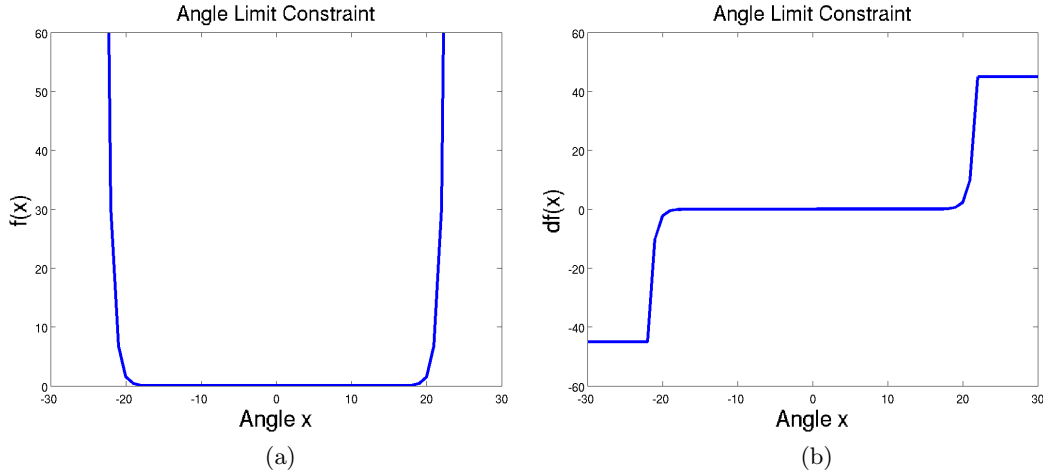


Figure 4.11: Angle limits are independently defined for each revolute joint. The plot visualizes the function (a), and its truncated derivative (b), that penalizes the deviation from an allowed range of ± 20.0 degree

Algorithm 1: Pose estimation for RGB-D data with *point-to-point* distances

$\tilde{\theta}$ = pose estimate of the previous frame

$i = 0, \theta_0 = \tilde{\theta}$

Repeat until convergence or max i_{thr} iterations

- Render meshes at pose θ
 - Find corresp. LO_{m2d} Section 4.2.2.2 - Eq. (4.6)
 - Find corresp. LO_{d2m} Section 4.2.2.3 - Eq. (4.8)
 - Find corresp. \mathcal{C} Section 4.2.2.4 - Eq. (4.12)
 - Find corresp. \mathcal{S} Section 4.2.2.5 - Eq. (4.15)
 - Find corresp. \mathcal{P} Section 4.2.2.6 - Eq. (4.16)
 - $\theta_{i+1} = \arg \min_{\theta} E(\theta, D)$
 - $i = i + 1$
-

4.2.2.8 Regularization

In case of occlusions or due to missing depth data, the objective function (4.5) based solely on the previous terms can be ill-posed. We therefore add a term that penalizes deviations from the previous estimated joint angles $\tilde{\theta}$:

$$E_{regularization}(\theta) = \sum_k (\theta_k - \tilde{\theta}_k)^2. \quad (4.18)$$

We use $\gamma_r = 0.02 C_{all}$.

4.2.2.9 Optimization

For pose estimation, we alternate between computing the correspondences LO_{m2d} (Section 4.2.2.2), LO_{d2m} (Section 4.2.2.3), \mathcal{C} (Section 4.2.2.4), \mathcal{S} (Section 4.2.2.5), and \mathcal{P}

(Section 4.2.2.6) according to the current pose estimate and optimizing the objective function (4.5) based on them as summarized in Algorithm 1. This process is repeated until convergence or until a maximum number of iterations i_{thr} is reached. It should be noted that the objective function $E(\theta, D)$ is only differentiable for a given set of correspondences. We optimize $E(\theta, D)$, which is a non-linear least squares problem, with the Gauss-Newton method as in [Brox et al., 2010].

4.2.3 Multicamera RGB

The previously described approach can also be applied to multiple synchronized RGB videos. To this end, the objective function (4.5) needs to be changed only slightly due to the differences of depth and RGB data. While the error is directly minimized in 3d for RGB-D data, we minimize the error for RGB images in 2d since all our observations are 2d. Instead of using a 3d *point-to-point* (4.6) or *point-to-plane* (4.7) measure, the error is therefore given by

$$\sum_c \sum_i \|\Pi_c(\mathbf{V}_i(\theta)) - x_{i,c}\|^2 \quad (4.19)$$

where $\Pi_c : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ are the known projection functions, mapping 3d points into the image plane of each static camera c , and $(\mathbf{V}_i, x_{i,c})$ is a correspondence between a 3d vertex and a 2d point. Furthermore, the salient point detector, introduced in Section 4.2.2.5, is not applied to the depth data but to all camera views. Since multiple high resolution views allow to detect more distinctive image features, we do not detect finger tips but finger nails in this case.

The only major change is required for the data terms $E_{model \rightarrow data}(\theta, D)$ and $E_{data \rightarrow model}(\theta, D)$ in (4.5). The term $E_{data \rightarrow model}(\theta, D)$ is replaced by an edge term that matches edge pixels in all camera views to the edges of the projected model in the current pose θ . As in the RGB-D case, the orientation of the edges is taken into account for matching and mismatches are removed by thresholding. Working with 2d distances though has the disadvantage of not being able to apply intuitive 3d distance thresholds, as presented in Section 4.2.2.3. In order to have an alternative rejection mechanism of noisy correspondences, we compute for each bone the standard deviation of the 2d error that is suggested by all of its correspondences. Subsequently, correspondences that suggest an error bigger than twice this standard deviation are rejected as outliers. The second term $E_{model \rightarrow data}(\theta, D)$ is replaced by a term based on optical flow as in [Ballan and Cortelazzo, 2008]. The term introduces temporal consistency and harness the higher resolution and frame rates of the RGB data in comparison to the RGB-D data.

4.3 Experimental Evaluation

Benchmarking in the context of 3d hand tracking remains an open problem [Erol et al., 2007b] despite recent contributions [Sridhar et al., 2013, Tang et al., 2014, 2013,

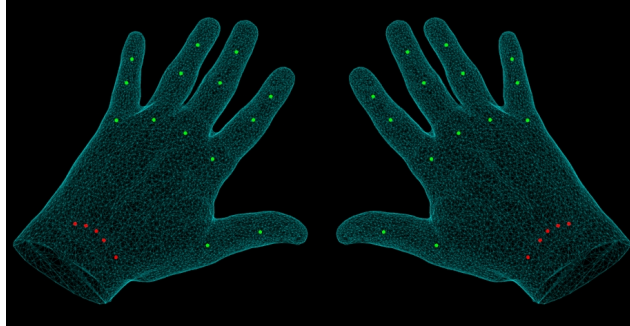


Figure 4.12: Hand joints used for quantitative evaluation. Only the green joints of our hand model are used for measuring the pose estimation error

Qian et al., 2014, Tompson et al., 2014]. The vast majority of them focuses on the problem of single hand tracking, especially in the context of real-time human computer interaction, neglecting challenges occurring during the interaction between two hands or between hands and objects. For this reason we captured 29 sequences in the context of hand-hand and hand-object interaction. The sequences were captured either with a single RGB-D camera or with 8 synchronized RGB cameras.

We first evaluate our approach on RGB-D sequences with hand-hand interactions in Section 4.3.1. Sequences with hand-object interactions are used in Section 4.3.2 for evaluation and finally our approach is evaluated on sequences captured with several RGB cameras in Section 4.3.4.

4.3.1 Monocular RGB-D - Hand-Hand Interactions

Related RGB-D methods [Oikonomidis et al., 2011a] usually report quantitative results only on synthetic sequences, which inherently include ground-truth, while for realistic conditions they resort to qualitative results.

Although qualitative results are informative, quantitative evaluation based on ground-truth is of high importance. We therefore manually annotated 14 sequences, 11 of which are used to evaluate the components of our pipeline and 3 for comparison with the state-of-the-art method [Oikonomidis et al., 2011a]. These sequences contain motions of a single hand and two interacting hands with 37 and 74 DoF, respectively. They vary from 100 to 270 frames and contain several actions, like “Walking”, “Crossing”, “Crossing and Twisting”, “Tips Touching”, “Dancing”, “Tips Blending”, “Hugging”, “Grasping”, “Flying”, as well as performing the “Rock” and “Bunny” gestures. As indicator for the accuracy of the annotations, we measured the standard deviation of 4 annotators, which is 1.46 pixels. All sequences were captured in 640x480 resolution at 30 fps with a Primesense Carmine 1.09 camera.

The error metric for our experiments is the 2d distance (pixels) between the projection of the 3d joints and the corresponding 2d annotations. The joints taken into account in the metric are depicted in Figure 4.12. Unless explicitly stated, we report the average over all frames of all relevant sequences.

Table 4.2: Evaluation of *point-to-point* (*p2p*) and *point-to-plane* (*p2plane*) distance metrics, along with iterations number of the optimization framework, using a 2d distance error metric (px). The highlighted setting is used for all other experiments

<i>Iterations</i>	5	10	15	20	30
<i>p2p</i>	7.33	5.25	5.05	4.98	4.91
<i>p2plane</i>	5.33	5.12	5.08	5.07	5.05

Our system is based on the objective function (4.5), consisting of several terms as described in Section 4.2.2. Two of them minimize the error between the posed mesh and the depth data by fitting the *model to the data* and the *data to the model*. A *salient point* detector further constrains the pose using fingertip detections in the depth image, while a *collision detection* method contributes to realistic pose estimates that are physically plausible. The function is complemented by the *physics simulation* component, that contributes towards more realistic interaction of hands with objects. However, this component is only relevant for hand-object interactions and thus it will be studied in detail in Section 4.3.2. In the following, we evaluate each component and the parameters of the objective function (4.5).

4.3.1.1 Distance Metrics

Table 4.2 presents an evaluation of the two distance metrics presented in Section 4.2.2.2, namely *point-to-point* (4.6) and *point-to-plane* (4.7), along with the number of iterations of the minimization framework. The *point-to-plane* metric leads to an adequate pose estimation error with only 10 iterations, providing a significant speed gain compared to *point-to-point*. If the number of iterations does not matter, the *point-to-point* metric is preferable since it results in a lower error and does not suffer from wrongly estimated normals.

For the first frame, we perform 50 iterations in order to ensure an accurate refinement of the manually initialized pose. For the chosen setup, we measure the runtime for the sequence “*Bunny*” that contains one hand and for the sequence “*Crossing and Twisting*” that contains two hands. For the first sequence, the runtime is 2.82 seconds per frame, of which 0.12 seconds are attributed to the salient point component \mathcal{S} and 0.65 to the collision component \mathcal{C} . For the second sequence, the runtime is 4.96 seconds per frame, of which 0.05 seconds are attributed to the component \mathcal{S} and 0.36 to the component \mathcal{C} .

4.3.1.2 Salient Point Detection - \mathcal{S}

The salient point detection component depends on the parameters w_s and λ , as described in Section 4.2.2.5. Table 4.3 summarizes our evaluation of the parameter λ spanning a range of possible values for both cases $w_s = 1$ and $w_s = \frac{c_s}{c_{thr}}$. The differences between the two versions of w_s is minor although the optimal range of λ varies for the two versions. The latter is expected since $\frac{c_s}{c_{thr}} \geq 1$ and smaller values of λ

Table 4.3: Evaluation of the weighting parameter λ in (4.14), using a 2d distance error metric (px). Weight $\lambda = 0$ corresponds to the objective function without salient points, noted as “ $LO + C$ ” in Table 4.6. Both versions of w_s described in Section 4.2.2.5 are evaluated. The highlighted setting is used for all other experiments

λ	0	0.3	0.6	0.9	1.2	1.5	1.8
$w_s = 1$	5.17	5.17	5.15	5.14	5.12	5.12	5.23
$w_s = \frac{c_s}{c_{thr}}$	5.14	5.12	5.12	5.12	5.12	5.22	5.61

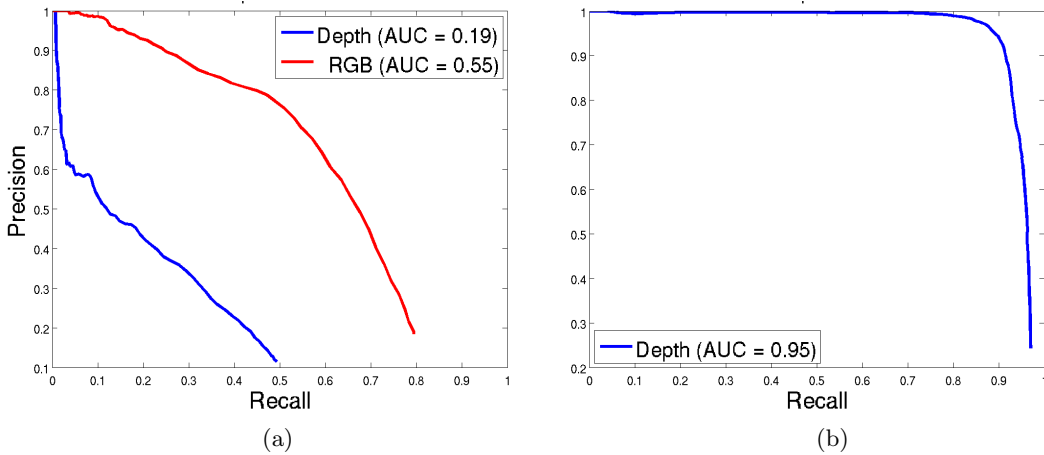


Figure 4.13: Precision-recall plot for (a) our RGB-D dataset and (b) the Dexter dataset. We show the performance of a fingertip detector trained only on depth (blue) and only rgb (red) images. The area under the curve (AUC) for our dataset (a) is 0.19 and 0.55 respectively. The AUC for the Dexter dataset (b) is 0.95

compensate for the mean difference to $w_s = 1$ in (4.14). If $\lambda = 0$ all detections are classified as false positives and the salient points are not used in the objective function (4.5).

To evaluate the performance of the detector, we follow the PASCAL-VOC protocol [Everingham et al., 2010]. Figure 4.13a shows the precision-recall plot for our RGB-D dataset including all hand-hand and hand-object sequences. The plot shows that the detector does not perform well on this dataset and suffers from the noisy raw depth data. This also explains why the salient term improves the pose estimation only slightly. We therefore trained and evaluated the detector also on the RGB data. In this case, the detection accuracy is much higher. We also evaluated the detector on the Dexter dataset [Sridhar et al., 2013]. On this dataset, the detector is very accurate. Our experiments on Dexter in Section 4.3.1.6 and a multi-camera RGB dataset in Section 4.3.4 will show that the salient points reduce the error more if the detector performs better.

Table 4.4: Evaluation of collision weights γ_c , using a 2d distance error metric (px). Weight 0 corresponds to the objective function without collision term, noted as “ $LO + \mathcal{S}$ ” in Table 4.6. Sequences are grouped in 3 categories: “*severe*” for intense, “*some*” for light and “*no apparent*” for imperceptible collision. “ $\geq \textit{some}$ ” is the union of “*severe*” and “*some*”. The highlighted value is the default value we use for all other experiments

γ_c	0	1	2	3	5	7.5	10	12.5
<i>All</i>	5.34	5.44	5.57	5.16	5.12	5.12	5.12	5.14
<i>Severe</i>	5.90	6.07	6.27	5.62	5.56	5.57	5.55	5.61
$\geq \textit{Some}$	5.44	5.57	5.72	5.23	5.18	5.19	5.18	5.22
<i>Some</i>	3.99	3.98	3.98	3.98	3.98	3.99	3.99	3.98

Table 4.5: Comparison of the proposed collision term based on 3d distance fields with correspondences between vertices of colliding triangles

	Corresponding vertices	Distance fields
All	6.66	5.12
Severe	7.96	5.55
$\geq \textit{Some}$	7.04	5.18
<i>Some</i>	4.12	3.99

4.3.1.3 Collision Detection - \mathcal{C}

The impact of the collision detection component is regulated in the objective function (4.5) by the weight γ_c . For the evaluation, we split the sequences in three sets depending on the amount of observed collision: *severe*, *some*, and *no apparent* collision. The set with *severe* collisions comprises “*Walking*”, “*Crossing*”, “*Crossing and Twisting*”, “*Dancing*”, “*Hugging*”, *some* collisions are present in “*Tips Touching*”, “*Rock*”, “*Bunny*”, and no collisions are apparent in “*Grasping*”, “*Tips Blending*”, “*Flying*”. Table 4.4 summarizes our evaluation experiments for the values of γ_c . The results show that over all sequences, the collision term reduces the error and that a weight $\gamma_c \geq 3$ gives similar results. For small weights $0 < \gamma_c < 3$, the error is even slightly increased compared to $\gamma_c = 0$. In this case, the impact is too small to avoid collisions and the term only adds noise to the pose estimation. As expected, the impact of the collision term is only observed for the sequences with *severe* collision.

The proposed collision term is based on a fast approximation of the distance field of an object. It is continuous and less sensitive to a change of the mesh resolution than a repulsion term based on 3d-3d correspondences between vertices of colliding triangles. To show this, we replaced the collision term by correspondences that move vertices of colliding triangles towards the counterpart. The results in Table 4.5 show that such a simple repulsion term performs poorly.

Table 4.6: Evaluation of the components of our pipeline. “*LO*” stands for local optimization and includes fitting both *data-to-model* (*d2m*) and *model-to-data* (*m2d*), unless otherwise specified. Collision detection is noted as “*C*”, while salient point detector is noted as “*S*”. The number of sequences where the optimization framework collapses is noted in the last row, while the mean error is reported only for the rest

<i>Components</i>	<i>LO_{m2d}</i>	<i>LO_{d2m}</i>	<i>LO</i>	<i>LO + C</i>	<i>LO + S</i>	<i>LO + CS</i>
<i>Mean Error (px)</i>	27.17	–	5.53	5.17	5.34	5.12
<i>Improvement (%)</i>			–	6.46	3.44	7.44
<i>Failed Sequences</i>	1/11	11/11	0/11	0/11	0/11	0/11

Table 4.7: Pose estimation error for each sequence using a 2d distance error metric (px)

	Walking	Crossing	Crossing Twisting	Tips Touching	Dancing	Tips Blending	Hugging	Grasping	Flying	Rock	Bunny
Mean Error	5.99	4.53	4.76	3.65	6.49	4.87	5.22	4.37	5.11	4.44	4.50
Std.Dev.	3.65	2.99	3.51	2.21	3.70	2.97	3.42	2.06	2.77	2.63	2.61
Max Error	24.19	18.03	22.80	13.60	20.25	18.36	20.03	11.05	15.03	14.76	10.63

4.3.1.4 Component Evaluation

Table 4.6 presents the evaluation of each component and the combination thereof. Simplified versions of the pipeline, fitting either just the *model to the data* (*LO_{m2d}*) or the *data to the model* (*LO_{d2m}*) can lead to a collapse of the pose estimation, due to unconstrained optimization. Our experiments quantitatively show the notable contribution of both the collision detection and the salient point detector. The best overall system performance is achieved with all four studied components of the objective function (4.5). The fifth term $E_{physics}$ is only relevant for hand-object interactions and will be evaluated in Section 4.3.2. Table 4.7 shows the error for each sequence. Figure 4.18, which is at the end of the chapter, depicts qualitative results for 8 out of the 11 sequences. It shows that the hand motion is accurately captured even in cases of close interaction and severe occlusions. The data and videos are available.²

4.3.1.5 Comparison to State-of-the-Art

Recently, Oikonomidis et al. [2011a,b, 2012] used particle swarm optimization (PSO) for a real-time hand tracker. For comparison we use the software released for tracking a single hand [Oikonomidis et al., 2011a], with the parameter setups used also in the other works. Each setup is evaluated three times in order to compensate for the manual initialization and the inherent randomness of PSO. Qualitative results depict the best result of all three runs, while quantitative results report the average error. Table 4.8 shows that our system outperforms [Oikonomidis et al., 2011a] in terms of tracking

²All annotated sequences are available at <http://files.is.tue.mpg.de/dtzionas/hand-object-capture.html>.



Figure 4.14: Qualitative comparison with [Oikonomidis et al., 2011a]. Each image pair corresponds to the pose estimate of the FORTH tracker (up) and our tracker (down)

Table 4.8: Comparison with [Oikonomidis et al., 2011a]. We evaluate the FORTH tracker with 4 parameter settings, 3 of which were used in the referenced literature of the last column

		<i>Mean (px)</i>	<i>Std.Dev. (px)</i>	<i>Max (px)</i>	<i>Generations</i>	<i>Particles</i>	<i>Reference</i>
<i>FORTH</i>	<i>set 1</i>	8.58	5.74	61.81	25	64	[Oikonomidis et al., 2011a]
	<i>set 2</i>	8.32	5.42	57.97	40	64	[Oikonomidis et al., 2011b]
	<i>set 3</i>	8.09	5.00	38.90	40	128	
	<i>set 4</i>	8.16	5.18	39.85	45	64	[Oikonomidis et al., 2012]
<i>Proposed</i>		3.76	2.22	19.92			

accuracy. Figure 4.14 shows a visual comparison. However, it should be noted that the GPU implementation of [Oikonomidis et al., 2011a] runs in real time using 25 generations and 64 particles, in contrast to our single-threaded CPU implementation.

4.3.1.6 Dexter dataset

We further evaluate our approach on the recently introduced Dexter dataset [Sridhar et al., 2013]. As suggested in [Sridhar et al., 2013], we use the first part of the sequences for evaluation and the second part for training. More specifically, the evaluation set contains the frames 018 – 158 of the sequence “adbadd”, 061 – 185 of “fingercount”, 020 – 173 of “fingerwave”, 025 – 224 of “flexex1”, 024 – 148 of “pinch”, 024 – 123 of “random”, and 016 – 166 of “tigergrasp”. We use only the depth of the Time-of-Flight camera.

The performance of our tracker is summarized in Tables 4.9 and 4.10. Since the dataset does not provide a hand model, we simply scaled our hand model in (x, y, z) direction by $(0.95, 0.95, 1)$. Since the annotations in the dataset do not correspond

Table 4.9: Pose estimation error of our tracker for each sequence of the Dexter dataset.

LO + SC	Mean Error	Std.Dev.	Max Error	
<i>Adbadd</i>	17.34	15.35	69.73	[mm]
<i>Fingercount</i>	11.94	7.18	47.77	
<i>Fingerwave</i>	10.88	5.47	49.62	
<i>Flexex1</i>	11.87	12.86	91.70	
<i>Pinch</i>	24.19	28.34	131.97	
<i>Random</i>	96.93	122.34	559.37	
<i>Tigergrasp</i>	11.77	5.36	30.18	
<i>Adbadd</i>	7.79	8.38	42.54	[px]
<i>Fingercount</i>	6.03	5.39	38.28	
<i>Fingerwave</i>	4.45	2.80	15.26	
<i>Flexex1</i>	5.24	8.37	61.40	
<i>Pinch</i>	12.56	16.48	73.16	
<i>Random</i>	59.93	77.77	307.00	
<i>Tigergrasp</i>	6.84	4.22	21.21	

to anatomical landmarks but are close to the finger tips, we compare the annotations with the endpoints of our skeleton. Table 4.9 shows the error of our tracker for each of the sequences, reporting the mean, the maximum, and the standard deviation of the error over all the tested frames. Despite of the differences of our hand model and the data, the average error is for most sequences only around $1cm$. Our approach, however, fails for the sequence “random” due to the very fast motion in the sequence.

Table 4.10 presents the evaluation of each component of our pipeline and the combination of them. On this dataset, both the collision term as well as the salient point detector reduce the error. Compared to Table 4.6, the error is more reduced. In particular, the salient point detector reduces the error more since the detector performs well on this dataset as shown in Figure 4.13b. Compared to “LO”, the average error of “LO + SC” is by more than 3.5mm lower. The average error reported by [Sridhar et al., 2013] on the slow part of the Dexter dataset is 13.1 mm.

4.3.2 Monocular RGB-D - Hand-Object Interactions

For the evaluation of the complete energy function (4.5) for hand-object interactions, we captured 7 new sequences² of hands interacting with several objects, either rigid (*ball*, *cube*) or articulated (*pipe*, *rope*). The DoF of the objects varies a lot. The rigid objects have 6 DoF, the *pipe* 7 DoF, and the *rope* 76 DoF. The sequences vary from 180 to 400 frames and contain several actions, like: “Moving a Ball” with one (43 DoF) or two hands (80 DoF), “Moving a Cube” with one hand (43 DoF), “Bending a Pipe” with two hands (81 DoF), and “Bending a Rope” with two hands (150 DoF). In addition, the sequences “Moving a Ball” with one hand and “Moving a Cube” were captured twice, one with occlusion of a manipulating finger and one without. Manual

Table 4.10: Evaluation of the components of the objective function (4.5) on the Dexter dataset. “*LO*” stands for local optimization and includes fitting both *data-to-model* and *model-to-data*. Collision detection is noted as “*C*” and salient point detector as “*S*”. The “random” sequence is excluded because our approach fails due to very fast motion

Components	Mean Error	Std.Dev.	
LO + <i>SC</i>	14.26	14.91	[mm]
LO + <i>S</i>	15.51	16.67	
LO + <i>C</i>	16.97	16.60	
LO	17.86	18.80	
LO + <i>SC</i>	6.90	8.88	[px]
LO + <i>S</i>	7.64	9.87	
LO + <i>C</i>	8.98	10.29	
LO	9.33	10.73	

ground-truth annotation was performed by a single subject.

For the salient point (*S*) and the collision detection component (*C*), we use the parameter setup presented in Section 4.3.1. The influence of the physics simulation component (*P*) and its parameters are evaluated in the following section. The error metric used is the 2d distance (pixel units) between the projection of the 3d joints and the 2d annotations as in Section 4.3.1 and visualized in Figure 4.12. Unless otherwise stated, we report the average over all frames of all seven sequences.

4.3.2.1 Physics Simulation - *P*

For the physics simulation, we model the entire scene, which includes the hands as well as manipulated and static objects, with a low resolution representation as described in Section 4.2.2.6 and visualized in Figure 4.9. Each component of the scene is characterized by three properties: friction, restitution, and mass. Since in each simulation step we consider each component except of the manipulated object as static, only the mass of the object is relevant, which we set to 1 kg. We set the restitution of the static scene and hands to 0 and of the object to 0.5. For the static scene, we use a friction value of 3. The friction for both the hand and the object are assumed to be equal. Since the main purpose of the physics simulation is to evaluate if the current pose estimates are physical stable, the exact values for friction, restitution, and mass are not crucial. To demonstrate this, we evaluate the impact of the friction value for hands and manipulated objects. For this experiment, we set the weight γ_{ph} equal to 10.0, being the same as the weight γ_c of the complementary collision detection component. The results presented in Table 4.11 show that the actual value of friction has no significant impact on the pose estimation error as long as it is in a reasonable range.

The impact of the physics simulation component $E_{physics}$ in the objective function (4.5) is regulated by the weight γ_{ph} . The term penalizes implausible manipulation or grasping poses. For the evaluation, we split the sequences in three sets depending on the amount of occlusions of the manipulating fingers: “*severe*” for intense (“Moving a

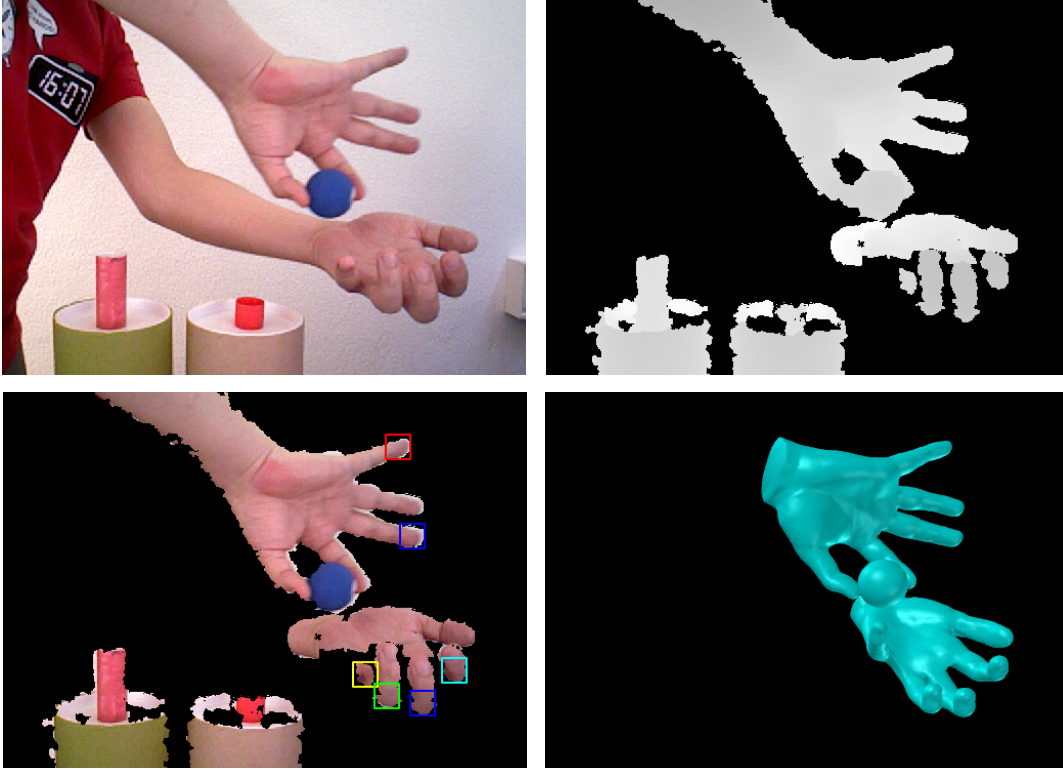


Figure 4.15: Failure case due to missing data and detection errors. The images show RGB image (top-left), input depth image (top-right), fingertip detections (bottom-left), and estimated pose (bottom-right). The detector operates on the raw depth image, while the RGB image is used just for visualization

Cube” with one hand and occlusion), “*some*” for light (“Moving a Ball” with one hand and occlusion, “Moving a Cube” with one hand) and “*no apparent*” for imperceptible occlusions (“Moving a Ball” with one and two hands, “Bending a Pipe”, “Bending a Rope”). Table 4.12 summarizes the pose estimation error for various values of γ_{ph} for the three subsets. Although the pose estimation error is only slightly reduced by $E_{physics}$, the results are physically more plausible. This is shown in Figure 4.19 at the end of the chapter, which provides a qualitative comparison between the setups “LO + \mathcal{SC} ” and “LO + \mathcal{SCP} ”. The images show the notable contribution of component \mathcal{P} towards more realistic, physically plausible poses, especially in cases of missing or ambiguous visual data, as in sequences with an occluded manipulating finger. To quantify this, we run the simulation for 35 iterations with a time-step of 0.1 seconds after the pose estimation and measured the displacement of the centroid of the object for each frame. While the average displacement is 9.26mm for the setup “LO + \mathcal{SC} ”, the displacement is reduced to 9.05mm by the setup “LO + \mathcal{SCP} ”. The tracking runtime for the aforementioned sequences for the setup “LO + \mathcal{SC} ” ranges from 4 to 8 seconds per frame. The addition of \mathcal{P} in the setup “LO + \mathcal{SCP} ” increases the runtime

Table 4.11: Evaluation of the friction value of both the hands and the object. We report the error over all the frames of all seven sequences with hand-object interactions using a 2d error metric (px). Value 3.0 is the same as the friction value of the static scene. The highlighted value is the default value we use for all other experiments

Friction	0.6	0.9	1.2	1.5	3.0	
Mean	6.19	6.18	6.19	6.17	6.17	[px]
Std.Dev.	3.82	3.81	3.81	3.81	3.81	

Table 4.12: Evaluation of physics weights γ_{ph} for “ $LO + SCP$ ”, using a 2d distance error metric (px). Weight 0 corresponds to the objective function without physics term, noted as “ $LO + SC$ ” in Table 4.13. Sequences are grouped in 3 categories: “*severe*” for intense, “*some*” for light and “*no apparent*” for imperceptible occlusion of manipulating fingers. “ $\geq some$ ” is the union of “*severe*” and “*some*”. The highlighted value is the default value we use for all other experiments

γ_{ph}	0	1	2	3	5	7.5	10	12.5
<i>All</i>	6.21	6.20	6.21	6.19	6.19	6.18	6.19	6.17
<i>Severe</i>	5.68	5.66	5.65	5.63	5.63	5.63	5.62	5.61
$\geq Some$	6.02	6.00	6.00	5.98	5.98	5.97	5.96	5.94

for most sequences for about 1 second. However, this increase might reach up to more than 1 minute depending on the complexity of the object and tightness of interaction, as in the case of “*Bending a Pipe*” with two hands (150 DoF), with the main bottleneck being the computation of the closest finger vertices to the manipulated object. Figure 4.20 depicts qualitative results for the full setup “ $LO + SCP$ ” of the objective function (4.5) for all seven sequences. The results show successful tracking of interacting hands with both rigid and articulated objects, whose articulation is described from 1 to as many as 71 DoF.

4.3.2.2 Component Evaluation

Table 4.13 presents the evaluation of each component and their combinations for the seven sequences with hand-object interaction. Since the physical simulations \mathcal{P} assumes that there are no severe intersections, it is meaningful only as a complement to the collision component \mathcal{C} . One can observe that the differences between the components are relatively small since the hand poses in the hand-object sequences are in general simpler than the poses in the sequences with tight hand-hand interactions as considered in Section 4.3.1. The collision term \mathcal{C} slightly increases the error, but without the term the hand poses are often physically implausible and intersect with the object. When comparing $LO + SC$ and $LO + SCP$, we see that the error is slightly reduced by the physics simulation component \mathcal{P} . The pose estimation errors for each sequence using $LO + SCP$ are summarized in Table 4.14.

Instead of using a fixed number of iterations per frame, a stopping criterion can

Table 4.13: Evaluation of the components of the objective function (4.5). “LO” stands for local optimization and includes fitting both *data-to-model* and *model-to-data*. Collision detection is noted as “C”, salient point detector as “S” and physics simulation as “P”. We report the error for fixed 10 iterations and for the stopping criterion $\varepsilon < 0.2mm$

Components	Fixed 10 Iterations		Stopping Thresh. 0.2 mm		
	Mean	Std.Dev.	Mean	Std.Dev.	
LO + SCP	6.19	3.81	6.25	3.86	
LO + SC	6.21	3.82	6.31	3.89	
LO + S	6.05	3.76	6.09	3.77	
LO + CP	6.19	3.83	6.31	3.90	
LO + C	6.24	3.84	6.38	3.94	
LO	6.07	3.77	6.15	3.83	
		px		px	

Table 4.14: Pose estimation error for each sequence

	<i>Moving Ball</i> 1 hand	<i>Moving Ball</i> 2 hands	<i>Bending</i> <i>Pipe</i>	<i>Bending</i> <i>Rope</i>	<i>Moving Ball</i> 1 hand, occlusion	<i>Moving Cube</i> 1 hand	<i>Moving Cube</i> 1 hand, occlusion	[px]
<i>Mean Error</i>	6.10	7.15	6.09	5.65	8.03	4.68	5.55	
<i>Std.Dev.</i>	3.90	4.82	3.07	3.04	5.47	2.61	3.28	

be used. We use the average change of the joint positions after each iteration. As threshold, we use $0.2mm$ and a maximum of 50 iterations. Table 4.13 shows that for the stopping criterion the impact of the terms is slightly more prominent, but it also shows that the error is slightly higher for all approaches. To analyze this more in detail, we report the distribution of required iterations until the stopping criterion is reached in Figure 4.16. Although LO + SCP requires a few more iterations until convergence compared to LO, it converges in 10 or less iterations in 92% of the frames, which supports our previous results. There are, however, very few frames where the approach has not converged after 50 iterations. In most of these cases, the local optimum of the energy is far away from the true pose and the error is increased with more iterations. These outliers are also the reason for the slight increase of the error in Table 4.13. For all combinations from LO to LO + SCP we observed this behavior, which shows that the energy can be further improved.

4.3.3 Limitations

As shown in Sections 4.3.1 and 4.3.2, our approach captures accurately the motion of hands tightly interacting either with each other or with a rigid or articulated object. However, for very fast motion like the “random” sequence of the Dexter dataset our approach fails. Furthermore, we assume that a hand model is given or can be acquired

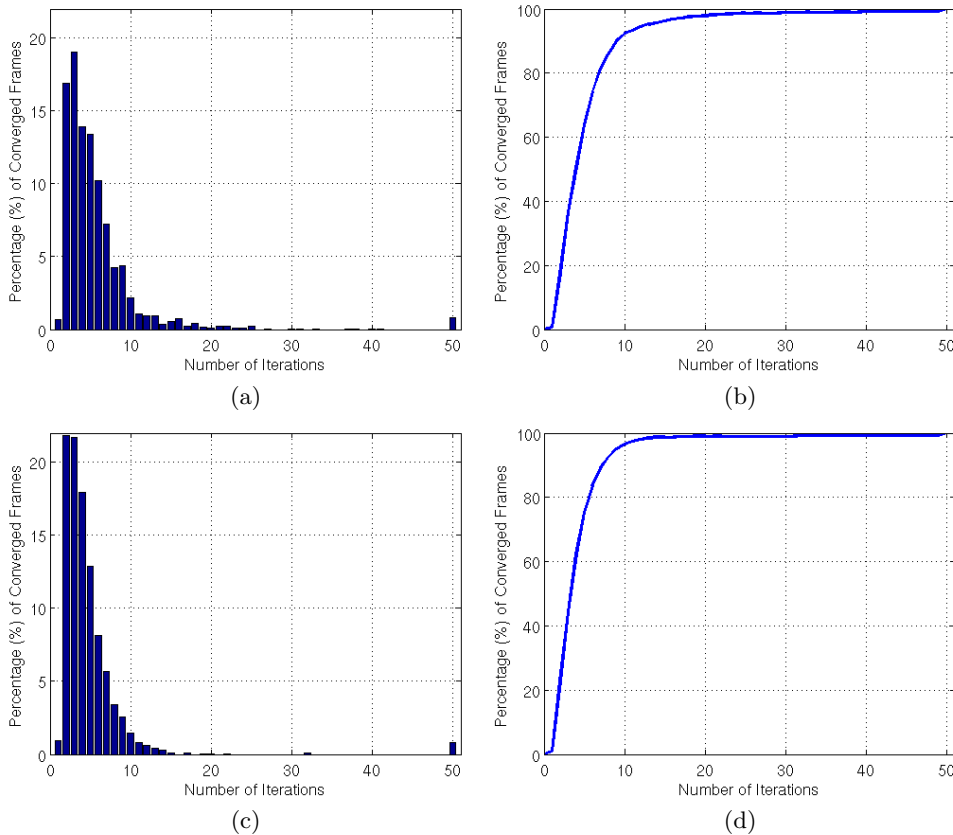


Figure 4.16: Number of iterations that are required to converge for LO + SCP (top) and LO (bottom). (a,c) Distribution of frames where the pose estimation converged after a given number of iterations. (b,d) Cumulative distribution

by an approach like [Taylor et al., 2014]. Figure 4.15 also visualizes an inaccurate hand pose of the lower hand due to missing depth data and two detections, which are not at the finger tips but located at other bones.

4.3.4 Multicamera RGB

We finally evaluated the approach for sequences captured using a setup of 8 synchronized cameras recording FullHD footage at 50 fps. To this end, we recorded 9 sequences that span a variety of hand-hand and hand-object interactions, namely: “*Praying*”, “*Fingertips Touching*”, “*Fingertips Crossing*”, “*Fingers Crossing and Twisting*”, “*Fingers Folding*”, “*Fingers Walking*” on the back of the hand, “*Holding and Passing a Ball*”, “*Paper Folding*” and “*Rope Folding*”. The length of the sequences varies from 180 to 1500 frames.

Figure 4.21 shows one frame from each of the tested sequences and the obtained results overlaid on the original frames from two different cameras. Visual inspection reveals that the proposed algorithm works also quite well for multiple RGB cameras

Table 4.15: Quantitative evaluation of the algorithm performance with respect to the used visual features: edges \mathcal{E} , collisions \mathcal{C} , optical flow \mathcal{O} , and salient points \mathcal{S} . LO stands for our local optimization approach, while HOPE64 and HOPE128 stand for our implementation of [Oikonomidis et al., 2011a] with 64 and 128 particles respectively, evaluated over 40 generations.

Used features	Mean	Std.Dev.	Max	
LO + \mathcal{E}	3.11	4.52	49.86	[mm]
LO + \mathcal{EC}	2.50	2.89	52.94	
LO + \mathcal{ECO}	2.38	2.25	16.84	
LO + \mathcal{ECOS}	1.49	1.44	13.27	
HOPE64 + \mathcal{ECOS}	4.86	3.69	31.05	
HOPE128 + \mathcal{ECOS}	4.67	3.28	41.11	

Used features	Mean	Std.Dev.	Max	
LO + \mathcal{E}	2.36	6.84	94.58	[deg]
LO + \mathcal{EC}	1.98	4.57	91.89	
LO + \mathcal{ECO}	1.84	3.81	60.09	
LO + \mathcal{ECOS}	1.88	3.90	44.51	
HOPE64 + \mathcal{ECOS}	4.35	7.11	58.61	
HOPE128 + \mathcal{ECOS}	4.73	7.46	78.65	

even in challenging scenarios of very closely interacting hands with multiple occlusions. The data and videos are available.³

4.3.4.1 Component Evaluation

As for the RGB-D sequences, we also evaluate the components of our approach. To this end, we synthesized two sequences: first, fingers crossing and folding, and second, holding and passing a ball, both similar to the ones captured in the real scenario. Videos were generated using a commercial rendering software. The pose estimation accuracy was then evaluated both in terms of error in the joints position, and in terms of error in the bones orientation.

Table 4.15 shows a quantitative evaluation of the algorithm performance with respect to the used visual features. It can be noted that each feature contributes to the accuracy of the algorithm and that the salient points \mathcal{S} clearly boost its performance. The benefit of the salient points is larger than for the RGB-D sequences since the localization of the finger tips from several high-resolution RGB cameras is more accurate than from a monocular depth camera with lower resolution. This is also indicated by the precision-recall curves in Figure 4.13a.

We also compared with [Oikonomidis et al., 2011a] on the synthetic data where we used an own implementation since the publicly available source code requires a single RGB-D sequence. We also added the salient points term and used two settings, namely 64 and 128 particles over 40 generations. The results in Table 4.15 show that

³<http://files.is.tue.mpg.de/dtzionas/hand-object-capture.html>

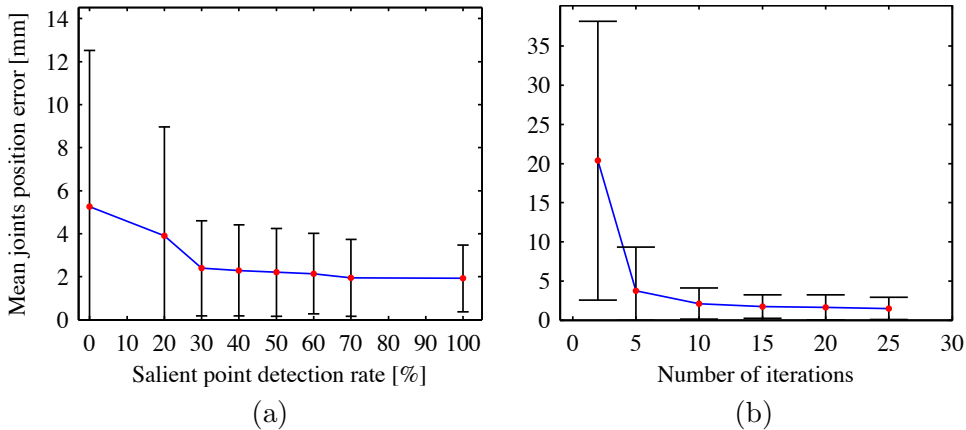


Figure 4.17: Quantitative evaluation of the algorithm performance on noisy data, with respect to the salient point detection rate (a), and the number of iterations (b). Black bars indicate the standard deviation of the obtained error.

our approach estimates the pose with a lower error and confirm the results for the RGB-D sequences reported in Table 4.8.

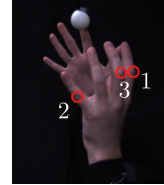
In order to make the synthetic experiments as realistic as possible, we simulated noise in all of the visual features. More precisely, edge detection errors were introduced by adding structural noise to the images, i.e. by adding and subtracting at random positions in each image 100 circles of radius varying between 10 and 30 pixels. The optical flow features corresponding to those circles were also not considered. Errors in the salient point detector were simulated by randomly deleting detections as well as by randomly adding outliers in a radius of 200 pixels around the actual features. Gaussian noise of 5 pixels was further introduced on the coordinates of the resulting salient points. Figure 4.17(a) shows the influence of the salient point detector on the accuracy of the pose estimation in case of noisy data. This experiment was run with a salient point false positive rate of 10%, and with varying detection rates. It is visible that the error quickly drops very close to its minimum even with a detection rate of only 30%.

Figure 4.17(b) shows the convergence rate for different numbers of iterations. It can be noted that the algorithm accuracy becomes quite reasonable after just 10 – 15 iterations, which is the same as for the RGB-D sequences.

We also annotated one of the captured sequences for evaluation. Since annotating joints in multiple RGB cameras is more time consuming than annotating joints in a single RGB-D camera, we manually labeled only three points on the hands in all camera views of the sequence *“Holding and Passing a Ball”*. Since we obtain 3d points by triangulation, we therefore use the 3d distance between these points and the corresponding vertices in the hand model as error metric. Table 4.16 shows the tracking accuracy obtained in this experiment. Overall, the median of the tracking error is at maximum 1cm.

Table 4.16: Results obtained on the manually marked data for the multicamera RGB sequences. The table reports the distance in mm between the manually tracked 3d points and the corresponding vertices on the hand model. The figure shows the positions of the tracked points on the hand.

Points	Median	Mean	Std.Dev.	Max
Point 1	06.98	07.98	3.54	20.53
Point 2	11.14	12.28	5.22	23.48
Point 3	10.91	10.72	4.13	24.68



4.4 Summary

In this chapter we have presented a framework that captures the articulated motion of hands and manipulated objects from monocular RGB-D videos as well as multiple synchronized RGB videos. Contrary to works that focus on gestures and single hands, we focus on the more difficult case of intense hand-hand and hand-object interactions. To address the difficulties, we have proposed an approach that combines in a single objective function a generative model with discriminatively trained salient points, collision detection and physics simulation. Although the collision and physics term reduce the pose estimation only slightly, they increase the realism of the captured motion, especially under occlusions and missing visual data. We performed qualitative and quantitative evaluations on 8 sequences captured with multiple RGB cameras and on 21 sequences captured with a single RGB-D camera. Comparisons with an approach based on particle swarm optimization [Oikonomidis et al., 2011a] for both camera systems revealed that our model achieves a higher accuracy for hand pose estimation. For the first time, we present successful tracking results of hands interacting with highly articulated objects.

Hand tracking can be an integral component for several novel applications. In Chapter 5 we present such an example application, where we incorporate the rich information of 3d hand motion in a 3d object reconstruction pipeline for the case of hand-object interaction scenarios.

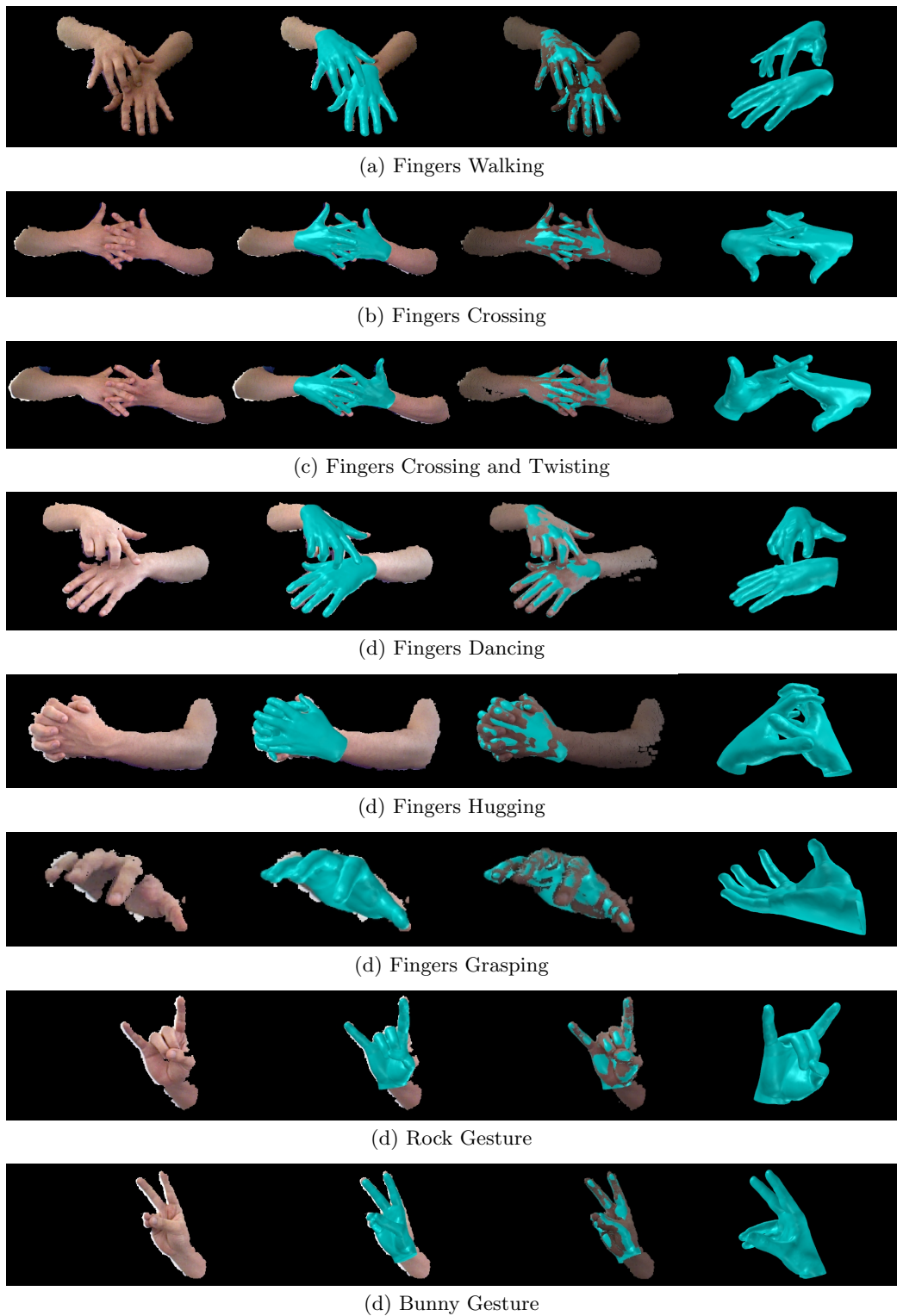


Figure 4.18: Some of the obtained results. (Left) Input RGB-D image. (Center-Left) Obtained results overlaid on the input image. (Center-Right) Obtained results fitted in the input point cloud. (Right) Obtained results from another viewpoint.

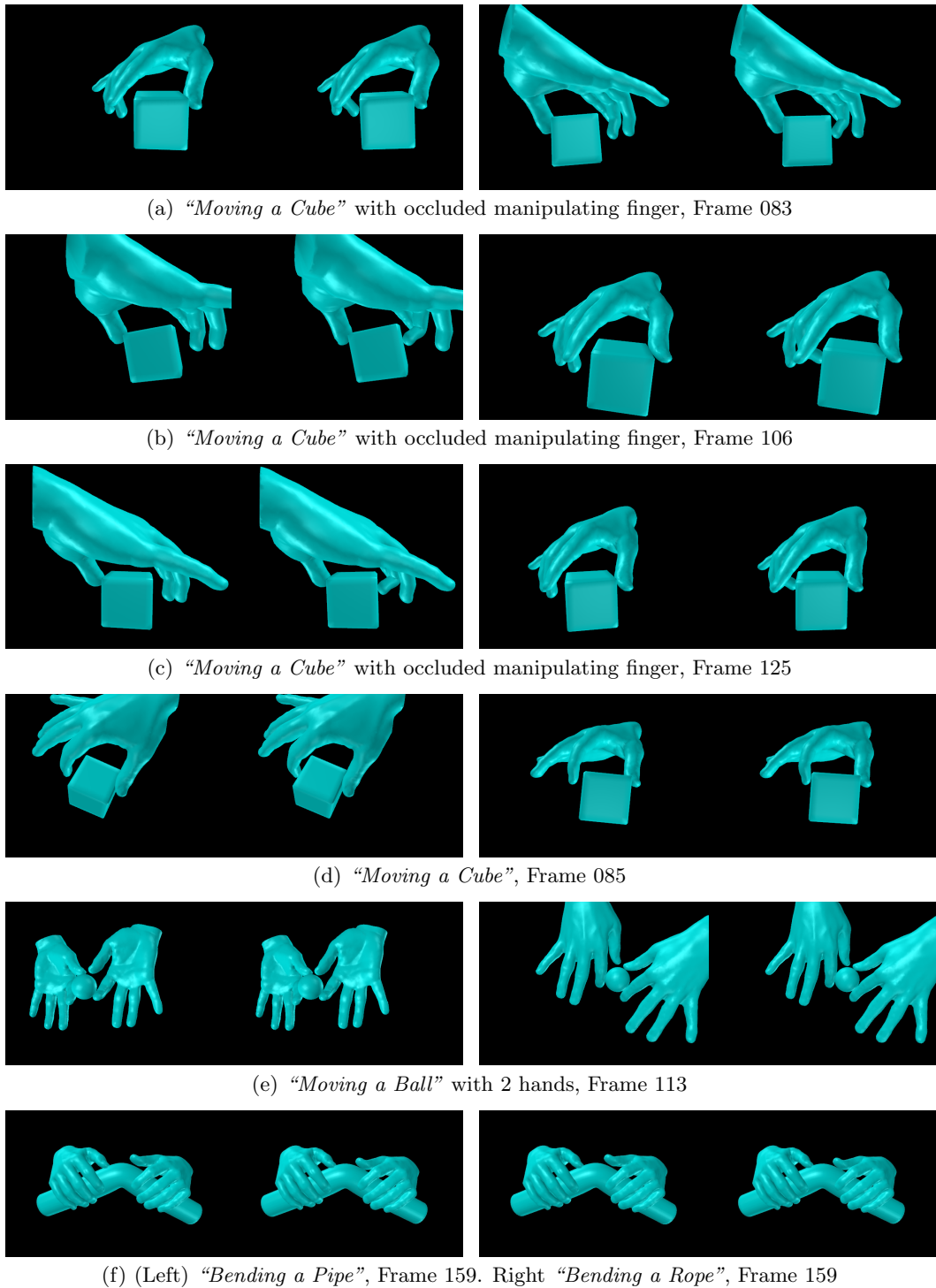


Figure 4.19: The impact of the physics component. For each image couple, the left image corresponds to $LO + SCx$ and the right one to $LO + SCP$. In the case of missing or ambiguous input visual data, as in sequences with occluded manipulating finger, the contribution of the physics component towards better physically plausible poses becomes more prominent

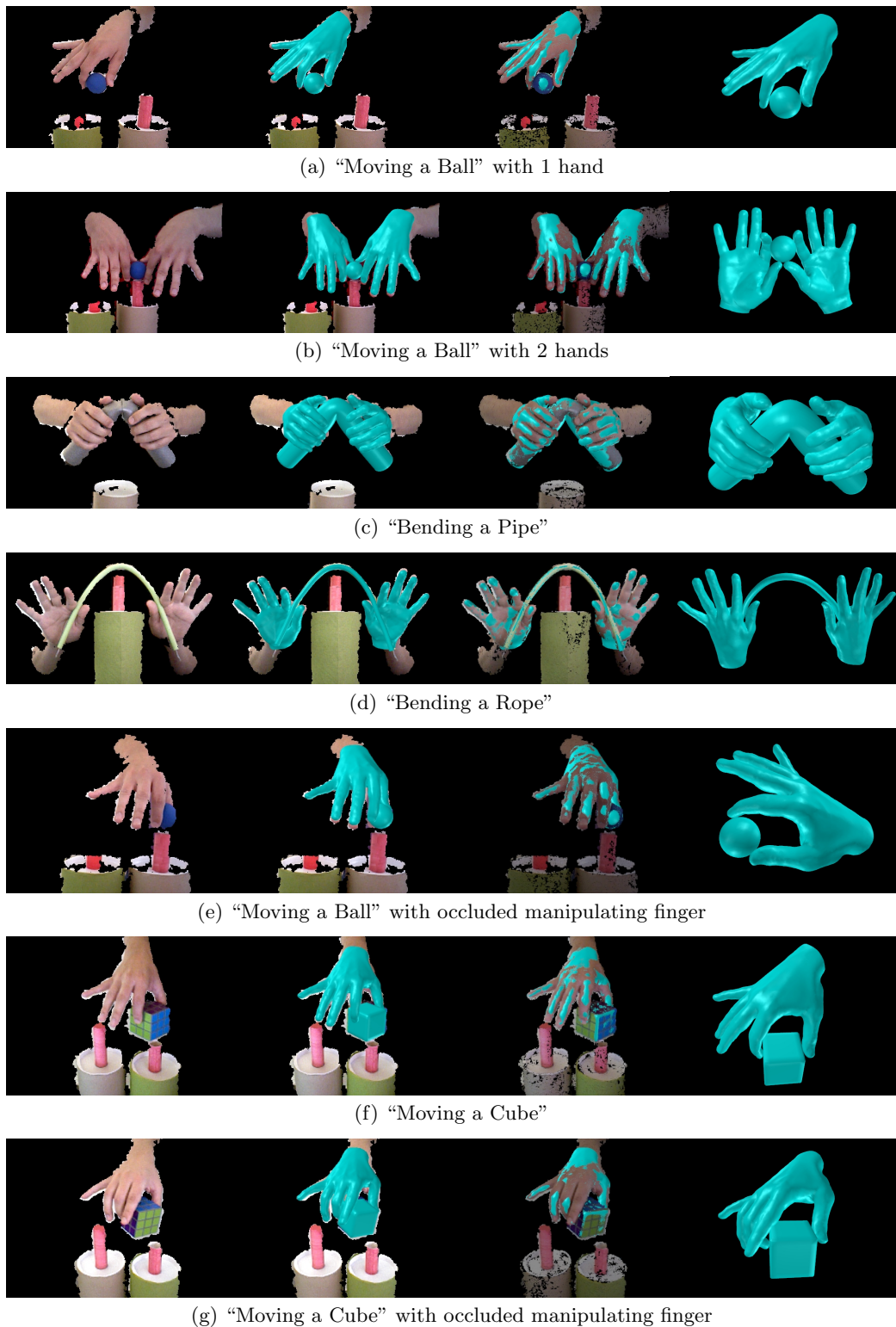


Figure 4.20: Some of the obtained results. (Left) Input RGB-D image. (Center-Left) Obtained results overlaid on the input image. (Center-Right) Obtained results fitted in the input point cloud. (Right) Obtained results from another viewpoint.

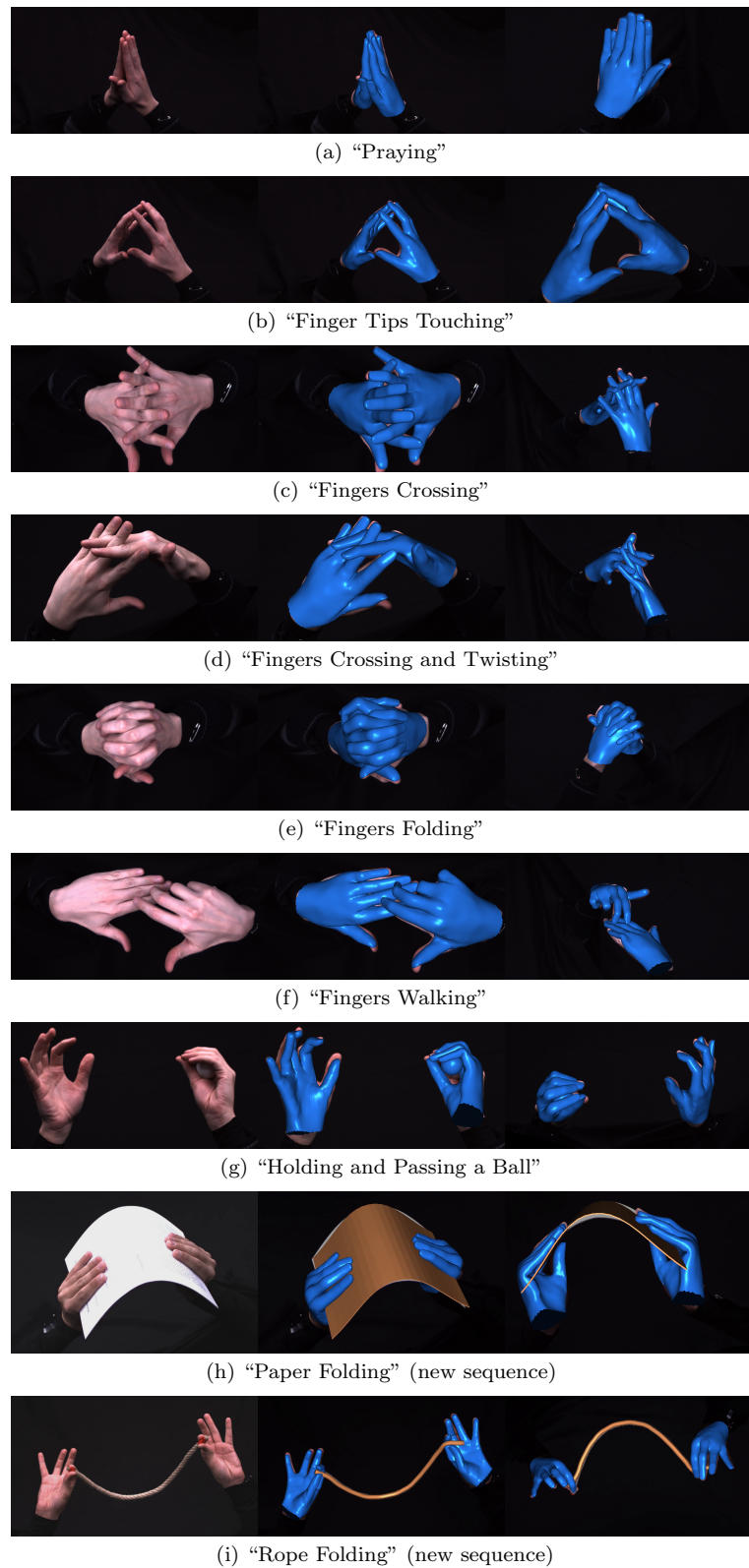


Figure 4.21: Some of the obtained results. (Left) One of the input RGB images. (Center) Obtained results overlaid on the input image. (Right) Obtained results from another viewpoint.

3D Object Reconstruction from Hand-Object Interactions

In Chapter 4 we presented an approach to effectively capture the motion of human hands in action. Hand tracking systems have the potential to be integral parts of a wide range of novel applications that incorporate the rich information of hand motion. In this chapter we show an example of such an application.

Recent advances have enabled 3d object reconstruction approaches using a single off-the-shelf RGB-D camera. Although these approaches are successful for a wide range of object classes, they rely on stable and distinctive geometric or texture features. Many objects like mechanical parts, toys, household or decorative articles, however, are textureless and characterized by minimalistic shapes that are simple and symmetric. Existing in-hand scanning systems and 3d reconstruction techniques fail for such symmetric objects in the absence of highly distinctive features. In this chapter, we show that extracting 3d hand motion for in-hand scanning effectively facilitates the reconstruction of even featureless and highly symmetric objects and we present an approach that fuses the rich additional information of hands into a 3d reconstruction pipeline, significantly contributing to the state-of-the-art of in-hand scanning.

Contents

5.1	Introduction	64
5.2	Related work	65
5.3	Hand motion capture for in-hand scanning	66
5.3.1	Preprocessing	66
5.3.2	Hand motion capture	67
5.4	Object reconstruction	67
5.4.1	Contact Points Computation	67
5.4.2	Reconstruction	68
5.5	Experiments	70
5.5.1	Synthetic data	70
5.5.2	Realistic data	71
5.6	Summary	76

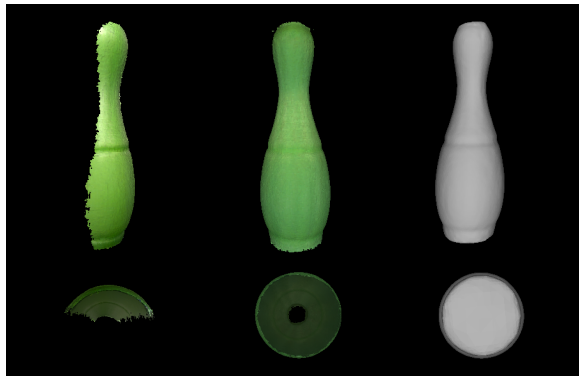


Figure 5.1: Reconstruction of a symmetric, textureless object. Both the front and the bottom view are provided for better visualization. Left: Existing in-hand scanning approaches fail for such objects. Middle and right: Successful reconstruction by the proposed in-hand scanning system that incorporates 3d hand motion capture.

5.1 Introduction

The advent of affordable RGB-D sensors has opened up a whole new range of applications based on the 3d perception of the environment by computers, which includes the creation of a virtual 3d representation of real objects. A moving camera can navigate in space observing the real world, while incrementally fusing the acquired frames into a 3d virtual model of it. Similarly, a static camera can observe a scene and dynamically reconstruct the observed moving objects. This domain has attracted much interest lately in the computer vision, the graphics and the robotics (SLAM) community, as it enables a plethora of other applications, facilitating among others 3d object detection, augmented reality, the internet of things, human-computer-interaction and the interaction of robots with the real world.

The field has matured [Salvi et al., 2007] since its beginning in the early 80s [Lucas and Kanade, 1981] and during the 90s [Besl and McKay, 1992, Blais and Levine, 1995, Pulli, 1999, Curless and Levoy, 1996]. Nowadays, several commercial solutions for 3d scanning with an off-the-shelf RGB-D camera have appeared, e.g. Fabletec [Sturm et al., 2013], Skanect [SKANECT], iSense [iSense], KScan3d [KSCAN3D], Shapify [Li et al., 2013] and Kinect-Fusion [Newcombe et al., 2011]. Several open-source projects like KinFu address the same problem, while other commercial solutions as MakerBot-Digitizer employ a laser scanning device along with sensorimotor information from a turntable.

Instead of a turntable, an object can also be rotated by hand in case of a static camera. This setting is very convenient for hand-sized objects since moving an object is more practical than moving a camera with a cable. Such a setup is also called *in-hand scanning* [Rusinkiewicz et al., 2002]. Weise et al. presented a real-time in-hand scanning system [Weise et al., 2008] that was later augmented with online loop closure [Weise et al., 2011]. Although the results are very convincing, the method

uses the hand only as a replacement of a turntable and discards the hand information. When the objects are textureless and contain very few geometric features, the in-hand scanning fails as illustrated in Figure 5.1.

In this chapter, we propose to use the hand motion for in-hand scanning as an additional cue to reconstruct also textureless objects. Instead of discarding the hands with the use of a black glove [Weise et al., 2011], we track the hand pose and use the captured hand motion together with texture and geometric features for object reconstruction as in Figure 5.3. Since the hand motion provides additional information about the object motion, we can reconstruct even textureless and symmetric objects as shown in Figure 5.1.

5.2 Related work

During the last decades several real-time in-hand scanning systems like [Rusinkiewicz et al., 2002, Weise et al., 2008, 2011] have been presented. Such systems are able to provide real-time registration of the input frames, while the interactivity enables the user to guide the reconstruction process. Assuming high temporal continuity and objects with rich geometric features, the quality of the final reconstruction can be sufficient. Some methods add an offline optimization step [Pulli, 1999] to solve the loop closure problem, but in this case the final result might differ from the intermediate result. In order to solve this issue, Weise et al. presented a real-time in-hand scanning system [Weise et al., 2008] that was later augmented with online loop closure [Weise et al., 2011]. They follow an as-rigid-as-possible (ARAP) approach based on surfels to minimize registration artifacts. Due to online loop closure the approach does not require any post-processing. A different approach is proposed in STAR3D [Yuheng Ren et al., 2013] where a 3d level-set function is used for simultaneous tracking and reconstruction of rigid objects. Similarly to in-hand scanning, this approach works only for objects with sufficient geometric or texture features. In order to reconstruct textureless and symmetric objects, additional information from sensors, markers [Mihalyi et al., 2013] or a robotic manipulator is needed [Kraft et al., 2008, Krainin et al., 2011].

In this chapter, we propose to extract the additional information directly from the hand within an in-hand scanning framework. Instead of simply discarding the hand [Rusinkiewicz et al., 2002, Weise et al., 2008, 2011], we capture the hand motion. In recent years, there has been a progress in hand motion capture. In particular, capturing of hand-object interactions has become of increasing interest [Romero et al., 2010, Hamer et al., 2009, 2010, Oikonomidis et al., 2011b, Kyriazis and Argyros, 2013, Ballan et al., 2012]. These approaches assume that a model of the object is given, while we aim to reconstruct the object during hand-object interactions. In [Michel et al., 2014] a rigid tool is tracked in a multicamera setup to reconstruct textureless and even transparent objects. Shape carving is in this case explicitly performed by the tool and the tool needs to be swept over the entire objects, which can be time-consuming. In contrast to in-hand scanning, this approach needs an additional tool. Static objects have also been used in [Salas-Moreno et al., 2013] to augment a SLAM system with



Figure 5.2: The hand tracker used in the in-hand scanning pipeline. The left image shows the raw depth input map, the middle image shows the hand pose overlaid on top of the RGB-D data, while the right image shows just the hand pose.

the pose of repetitive objects in a scene.

Recently [Panteleris et al., 2015] presented an in-hand scanning system that also captures the motion of the hands. However, they use the tracked hand pose only in order to segment the hand and the object. On the contrary, we track the hand in order to extract useful information for the 3d reconstruction part of our in-hand scanning framework.

5.3 Hand motion capture for in-hand scanning

As illustrated in Figure 5.2, we observe an RGB-D video where a hand is interacting with an object. The data is first preprocessed as described in Section 5.3.1 and the hand pose is estimated in each frame as described in Section 5.3.2. We then exploit the captured hand motion to reconstruct the object as shown in Figure 5.1. The reconstruction process is described Section 5.4.

5.3.1 Preprocessing

We first remove irrelevant parts of the RGB-D image D by thresholding the depth values in order to avoid unnecessary processing like normal computation for distant points. To this end, we keep only points within a specified volume. For the used Primesense Carmine 1.09 sensor, only points (x, y, z) within the volume $[-100mm, 100mm] \times [-140mm, 220mm] \times [400mm, 1000mm]$ are kept. Subsequently we apply skin color segmentation on the RGB image using the Gaussian-Mixtures-Model (GMM) of [Jones and Rehg, 2002] and get the masked RGB-D images D_o for

the object and D_h for the hand.

5.3.2 Hand motion capture

In order to capture the motion of a hand, we employ an approach similar to Chapter 4. The approach uses a hand template mesh and parameterizes the hand pose by a skeleton and linear blend skinning [Lewis et al., 2000]. For pose estimation, we minimize an objective function, which consists of three terms (to simplify notation we drop the terms $E_{anatomy}$ and $E_{regularization}$ of Equation (4.5)):

$$E(\theta, D) = E_{model \rightarrow data}(\theta, D_h) + E_{data \rightarrow model}(\theta, D_h) + \gamma_c E_{collision}(\theta) \quad (5.1)$$

where D_h is the current preprocessed depth image for the hand and θ are the pose parameters of the hand. The first two terms of Equation (5.1) minimize the alignment error between the input depth data and the hand pose. The alignment error is measured by $E_{model \rightarrow data}$, which measures how well the model fits the observed depth data, and $E_{data \rightarrow model}$, which measures how well the depth data is explained by the model. $E_{collision}$ penalizes intersections of fingers and enhances realism by ensuring physically plausible poses. The parameter γ_c is set to 10 as in Chapter 4. For simplicity we do not use the additional term $E_{salient}$ of Chapter 4 for the detected salient points. The overall hand tracking accuracy for the hand joints is approximately $17mm$.

5.4 Object reconstruction

In order to use the captured hand motion for 3d reconstruction, we have to infer the contact points with the object. This is described in Section 5.4.1. The reconstruction process based on the estimated hand poses and the inferred contact points is then described in Section 5.4.2.

5.4.1 Contact Points Computation

In order to compute the *contact points*, we use the high-resolution mesh of the hand, which has been used for hand motion capture. To this end, we compute for each vertex associated to each end-effector the distance to the closest point of the object point cloud D_o . We first count for each end-effector the number of vertices with a closest distance of less than $1mm$. If an end-effector has more than 40 *candidate contact vertices*, it is labeled as a contact bone and all vertices of the bone are labeled as *contact vertices*. If there are not at least 2 end-effectors selected, we iteratively increase the distance threshold by $0.5mm$ until we have at least two end-effectors. In our experiments, we observed that the threshold barely exceeds $2.5mm$. As a result, we obtain for each frame pair the set of *contact correspondences* $(X_{hand}, X'_{hand}) \in \mathcal{C}_{hand}(\theta, D_h, D_o)$, where (X_{hand}, X'_{hand}) is a pair of *contact vertices* in the *source* and *target* frame, respectively. Figure 5.3 depicts the *contact correspondences* for a frame pair.

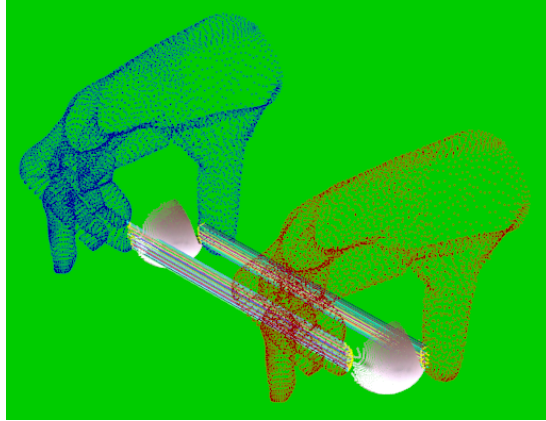


Figure 5.3: Illustration of the *contact correspondences* $(X_{hand}, X'_{hand}) \in \mathcal{C}_{hand}(\theta, D_h, D_o)$ between the *source frame* (red) and the *target frame* (blue). Although the correspondences are formed for all the vertices of the end-effectors of the manipulating fingers, we display only the detected *candidate contact points* to ease visualization. The candidate contact points are displayed with yellow color, while the multi-color lines show the *contact correspondences*. The white point cloud is a partial view of the unknown object whose shape is reconstructed during hand-object interaction.

5.4.2 Reconstruction

We use a feature-based approach for reconstruction, where we first align the currently observed point cloud (*source*) to the previous frame (*target*) and afterwards we align the transformed source by ICP to the previously accumulated transformed point cloud [Chen and Medioni, 1991] for refinement.

For pairwise registration, we combine features extracted from D_o and the *contact points*, which have been extracted from D_h and the hand pose θ . As a result, we minimize an energy function based on two weighted energies:

$$E(\theta, D_h, D_o, \mathbf{R}, \mathbf{t}) = E_{visual}(D_o, \mathbf{R}, \mathbf{t}) + \gamma_t E_{contact}(\theta, D_h, \mathbf{R}, \mathbf{t}) \quad (5.2)$$

where E is a measure of the discrepancy between the incoming and the already processed data, that needs to be minimized. In that respect, we seek the rigid transformation $T = (\mathbf{R}, \mathbf{t})$, where $\mathbf{R} \in SO(3)$ is a rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ is a translation vector, that minimizes the energy E by transforming the *source frame* accordingly.

The *visual energy* E_{visual} consists of two terms that are computed on the visual data of the object point cloud D_o :

$$E_{visual}(D_o, \mathbf{R}, \mathbf{t}) = E_{feat2d}(D_o, \mathbf{R}, \mathbf{t}) + E_{feat3d}(D_o, \mathbf{R}, \mathbf{t}) \quad (5.3)$$

The term E_{feat2d} is based on a sparse set of correspondences $\mathcal{C}_{feat2d}(D_o)$ using 2d SIFT [Lowe, 1999] features that are back-projected in 3d by the function $\varphi(x): \mathbb{R}^2 \rightarrow \mathbb{R}^3$, given the intrinsic parameters of the camera. The 2d SIFT keypoint correspondences in the source and target image respectively are denoted as $(x_{2d}, x'_{2d}) \in \mathcal{C}_{feat2d}(D_o)$, while $X_{2d} = \varphi(x_{2d})$ and $X'_{2d} = \varphi(x'_{2d})$ are the corresponding back-projected 3d points. E_{feat2d} is then formulated as

$$E_{feat2d}(D_o, \mathbf{R}, \mathbf{t}) = \sum_{(X_{2d}, X'_{2d}) \in \mathcal{C}_{feat2d}} \|X'_{2d} - (\mathbf{R}X_{2d} + \mathbf{t})\|^2. \quad (5.4)$$

In a similar manner, the term E_{feat3d} is based on a sparse set of correspondences $\mathcal{C}_{feat3d}(D_o)$. Instead of the image domain, we operate on the 3d point cloud by choosing ISS3D [Zhong, 2009] keypoints and the CSHOT [Tombari et al., 2011] feature descriptor, that augments the SHOT [Tombari et al., 2010] descriptor with texture information. This combination has been shown to work well for point clouds [Filipe and Alexandre, 2013, Tombari et al., 2013]. E_{feat3d} is then formulated as

$$E_{feat3d}(D_o, \mathbf{R}, \mathbf{t}) = \sum_{(X_{3d}, X'_{3d}) \in \mathcal{C}_{feat3d}} \|X'_{3d} - (\mathbf{R}X_{3d} + \mathbf{t})\|^2. \quad (5.5)$$

Finally, the term $E_{contact}$ depends on the current hand pose estimate θ and the hand point cloud D_h . Based on these, the *contact correspondences* $\mathcal{C}_{hand}(\theta, D_h, D_o)$ are computed as described in Section 5.4.1. Let $(X_{hand}, X'_{hand}) \in \mathcal{C}_{hand}(\theta, D_h, D_o)$ be the corresponding *contact points*, i.e. vertices, in the *source* and *target* frame respectively, then $E_{contact}(\theta, D_h, D_o)$ is written as

$$E_{contact}(\theta, D_h, D_o, \mathbf{R}, \mathbf{t}) = \sum_{(X_{hand}, X'_{hand}) \in \mathcal{C}_{hand}} \|X'_{hand} - (\mathbf{R}X_{hand} + \mathbf{t})\|^2. \quad (5.6)$$

The two terms in the energy function (5.2) are weighted since they have different characteristics. Although *visual correspondences* preserve local geometric or textural details better, they tend to cause a slipping of one frame upon another in case of textureless and symmetric objects. In this case, the *contact correspondences* ensure that the movement of the hand is taken into account. An evaluation of the weight γ_t is presented in Section 5.5.

The sparse correspondence sets \mathcal{C}_{feat2d} , \mathcal{C}_{feat3d} , and \mathcal{C}_{hand} provide usually an imperfect alignment of the *source* frame to the *target* frame either because of noise, ambiguities in the visual features or the pose, or a partial violation of basic assumptions like the rigid grasping of an object during interaction. For this reason, we refine the aligned source frame by finding a locally optimal solution based on dense ICP [Besl and McKay, 1992] correspondences. While for the sparse correspondences we align the current frame only to the previous one, during this refinement stage we align the current frame to the accumulation of all previously aligned frames [Chen and Medioni, 1991], i.e. the current partial reconstructed model. After finding a dense set $(X_{icp}, X'_{icp}) \in \mathcal{C}_{icp}(D_o)$ of ICP correspondences with maximum distance of 5mm, we

minimize the discrepancy between them

$$E_{icp}(D_o, \mathbf{R}, \mathbf{t}) = \sum_{(X_{icp}, X'_{icp}) \in \mathcal{C}_{icp}} \|X'_{icp} - (\mathbf{R}X_{icp} + \mathbf{t})\|^2. \quad (5.7)$$

5.4.2.1 Surface model

To obtain a mesh representation of the reconstructed object, we first employ a truncated signed distance function (TSDF) [Curless and Levoy, 1996, Newcombe et al., 2011] to get a volumetric representation. The TSDF volume has a dimension of $350mm$ for all objects with 256 voxel resolution and $6mm$ maximum voxel size. Subsequently we apply the marching-cubes [Lorensen and Cline, 1987] method to extract a mesh and remove tiny disconnected components. The final mesh is then obtained by Laplacian smoothing [Vollmer et al., 1999] followed by Poisson reconstruction [Kazhdan et al., 2006] with an octree with 10 layers in order to get a smooth, water-tight mesh with preserved details.

5.5 Experiments

In this section we show that although existing in-hand scanning pipelines fail for symmetric and textureless objects, the incorporation of hand motion capture can effectively improve the reconstruction, enabling the efficient and full reconstruction of such objects without the use of additional intrusive markers or devices in the scene. We present thus for the first time the effective reconstruction of 4 symmetric objects with an in-hand scanning system, which cannot be reconstructed by two state-of-the-art reconstruction systems. Furthermore, we perform an experiment with synthetic data, showing that the pipeline can also be applied to multicamera RGB videos.

The recorded sequences, calibration data, hand motion data, as well as video results, the resulting meshes and the source code for reconstruction are publicly available¹.

5.5.1 Synthetic data

In order to generate synthetic data we use the publicly available² data of a multicamera RGB hand tracker [Ballan et al., 2012]. We use the frames 180-203 of the sequence in which a hand interacts with a rigid ball. We generate synthetic point clouds by rendering the moving meshes. We then apply the pipeline described in Section 5.4 to the rendered point clouds and use the hand meshes and motion data of [Ballan et al., 2012].

The resulting accumulated and aligned point cloud is depicted in Figure 5.4. Figure 5.4(a) shows the reconstruction without hand motion data, while Figure 5.4(b) shows the reconstruction after the incorporation of hand motion into the in-hand scanning

¹<http://files.is.tue.mpg.de/dtzionas/ihScanning>

²<http://cvg.ethz.ch/research/ih-mocap>

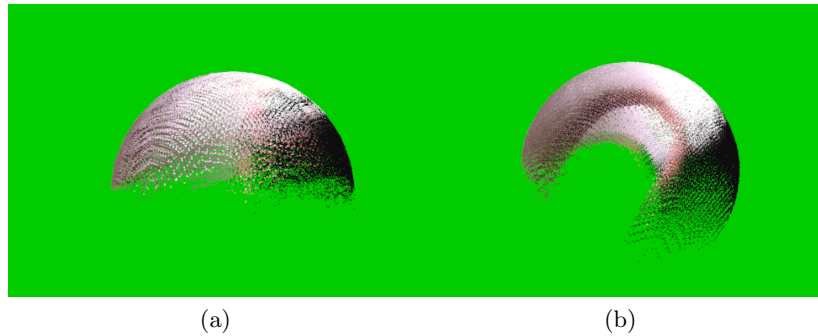


Figure 5.4: Reconstruction without (a) and with (b) hand motion capture on synthetic data generated from [Ballan et al., 2012]. Since the *visual correspondences* alone are not descriptive enough for symmetric objects, the reconstruction collapses to a hemisphere (a). On the contrary, the use of hand motion capture gives meaningful *contact correspondences*, successfully driving the reconstruction process (b). The clear observation of the occlusions by the manipulating fingers (b) indicates a sensible registration. The motion includes some notable translation and rotation, but the object is not fully rotated in order to allow for a complete reconstruction.

system. The reconstruction without the hand motion data collapses to a degenerate hemisphere, while it is clear that the hand motion data significantly contributes towards the effective reconstruction of the manipulated object. Parts of the object are never visible in the sequence. The occlusions caused by the manipulating fingers can be clearly observed, verifying the correct registration of the camera frames.

5.5.2 Realistic data

For our experiments with realistic data, we use a Primesense Carmine 1.09 short-range, structured-light RGB-D camera. Structured light sensors may not be optimal for hand pose estimation, in contrast to time-of-flight sensors, because significant parts of the hand completely disappear from the depth image in case of reflections or at some viewing angles. Nevertheless, the used hand tracker worked well with the sensor.

In order to perform both a qualitative and a quantitative evaluation, we have captured new sequences for the four objects depicted in Figure 5.5. As seen in Table 5.1, the size of the objects varies in order to be representative of several everyday objects. However, all objects have in common the high symmetry and the lack of distinctive geometrical and textural features, that renders them especially challenging for existing in-hand scanning systems.

In the following we show the successful reconstruction of these objects for the first time with an in-hand scanning system, while we systematically evaluate the performance of our pipeline both with respect to existing baselines, ground-truth object dimensions as well as state-of-the-art systems.

Table 5.1: Quantitative evaluation of the captured object shapes. The ground-truth parameters, estimated parameters and errors are given. We compare our proposed setup with $\gamma_t = 15$ with the methods KinFu and Skanect. For the methods highlighted with (*), we perform a reconstruction three times and we report the best results of them.

Dimensions Comparison	G.Truth	Ours $\gamma_t = 15$		KinFu (*)		Skanect (*)		Detect.Baseline		Enriched Texture	
		Capture	Diff.	Capture	Diff.	Capture	Diff.	Capture	Diff.	Capture	Diff.
Water-bottle diameter	73	82.3	9.3	66.2	6.8	64.3	8.7	86.6	13.6		
Water-bottle height	218	225.4	7.4	195.7	22.3	222.1	4.1	237.4	19.4		
Bowling-pin head diameter	50	50.8	0.8	54.1	4.1	39.0	11.0	48.7	1.3	49.8	0.2
Bowling-pin body diameter	82	90.0	8.0	70.9	11.1	63.8	18.2	93.2	11.2	89.4	7.4
Bowling-pin height	268	275.2	7.2	239.3	28.7	270.9	2.9	272.4	4.4	267.7	0.3
Small-bottle diameter	52	57.7	5.7	45.6	6.4	39.5	12.5	61.6	9.6		
Small-bottle height	80	89.5	9.5	78.1	1.9	84.9	4.9	95.0	15.0		
Sphere diameter	70	71.4	1.4	46.9	23.1	43.8	26.2	72.2	2.2		
Average			6.1625		13.05		11.0625		9.5875		
Sphere volume	179503	190490	10987	53988	125515	43974	135529	196965	17462		

mm
mm³



Figure 5.5: The objects to be reconstructed (left to right): a *water-bottle*, a *bowling-pin*, a *small-bottle* and a *sphere*. The dimensions of the objects are summarized in Table 5.1. All four objects are characterized by high symmetry and lack of distinctive geometrical and textural features, causing existing in-hand pipelines to fail. We perform successful reconstruction of all four objects.

5.5.2.1 Quantitative evaluation

Acquiring a ground-truth measure is difficult for most objects, however, for symmetric ones it is easy to measure the dimensions of some distinctive areas. We therefore measure manually the distinctive dimensions of the objects depicted in Figure 5.5 in order to quantitatively evaluate the proposed setup. The ground-truth dimensions, along with the measured ones by our approach and the measurement error, are presented in Table 5.1. Especially for the case of the sphere, we can easily acquire a ground truth value for its volume, that is less prone to measurement errors introduced by human factors.

During quantitative evaluation, we measured the distinctive dimensions of the water-tight meshes that are reconstructed by our pipeline. We then evaluate the most important parameter of our pipeline, namely the weight γ_t that steers the influence of the *contact correspondences* in the in-hand scanning system. The results of our

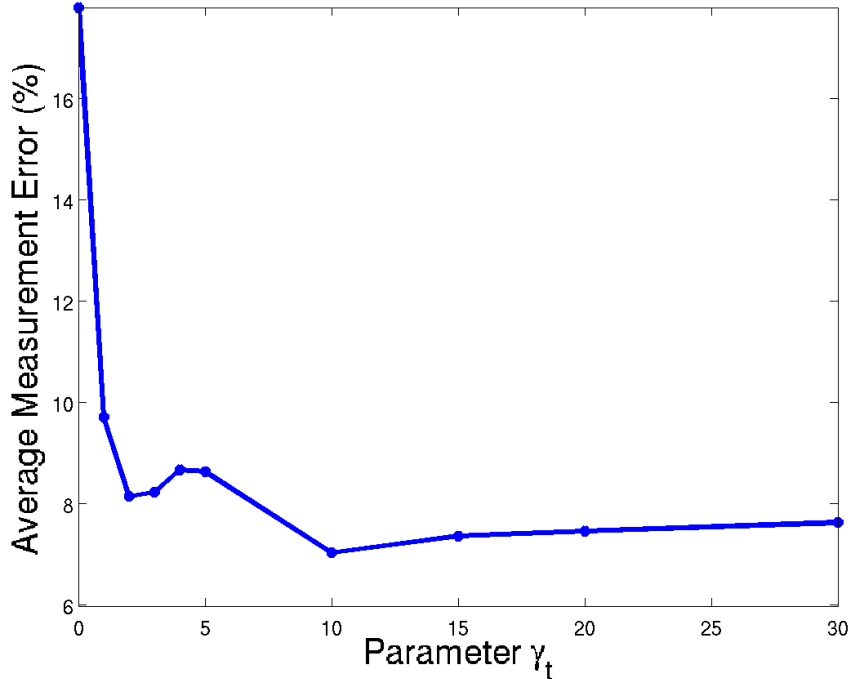


Figure 5.6: Quantitative evaluation of the weight γ_t of the energy function (5.2) based on the ground truth dimensions of the objects presented in Table 5.1. The error for each parameter is normalized by the ground-truth value.

experiments are summarized in Figure 5.6, which plots the mean accumulated estimation errors of the eight parameters p normalized by the ground-truth values, i.e., $\frac{1}{8} \sum_p \frac{|p_{est} - p_{GT}|}{p_{GT}}$. Without the *contact correspondences* and using only *visual features* ($\gamma_t = 0$), the error is relatively high since the reconstruction for symmetric objects fails in this case. Even small values for γ_t result in an abrupt drop in the error metric, however, the influence of the *contact correspondences* starts becoming more apparent for values above 5. In all subsequent experiments, we use $\gamma_t = 15$.

The performance of our setup is described in more details in Table 5.1 which provides a direct comparison to the measured object dimensions. The average error is only $6mm$, comparable to the noise of commodity RGB-D sensors, showing the potential of such a system for a wide range of everyday applications.

For evaluation against a reference reconstructed shape, we also add textural features on the *bowling-pin* in the form of stickers, as depicted in Figure 5.7, without altering the geometrical shape of the object. We then perform the reconstruction by rotating the object on a turntable. The resulting reconstruction is illustrated in Figure 5.7, while quantitative measures are provided for comparison in Table 5.1.

We further test the effectiveness of our system in comparison to two state-of-the-art systems, namely *KinFu*, [KINFU], an open-source implementation of Kinect-Fusion [Newcombe et al., 2011], and the similar commercial system *Skaneet* [SKANEET] For



Figure 5.7: The *bowling-pin* object enriched with 2d texture using stickers (left). The added texture allows for the reconstruction of a reference ground truth shape (middle, right) facilitating quantitative evaluation.

technical reasons, we use *KinFu* with a Kinect and *Skanect* with the Structure-IO camera. Existing in-hand scanning approaches are expected to have a performance similar to these systems. For comparison, we use *KinFu* and *Skanect* to reconstruct the four objects depicted in Figure 5.5. A turntable rotates each object for approximately 450 degrees in front of a static camera, while we repeat the process three times and report only the best run in order to assure objectiveness. Quantitative performance measures are provided for these methods in Table 5.1.

In order to show the important role of the hand pose in our reconstruction pipeline, we replace the contact correspondences \mathcal{C}_{hand} based on *contact vertices* with correspondences \mathcal{C}_{detect} based on a *contact detector*. In that respect we train a Hough forest [Gall et al., 2011b] detector that detects finger-object contacts in RGB images. We then establish correspondences $(X_{det}, X'_{det}) \in \mathcal{C}_{detect}$ between the points enclosed by the detection bounding boxes in the *source* and *target* frames simply by associating points with the same 2d coordinates inside the fixed-sized bounding boxes. In that respect, the term $E_{contact}(\theta, D_h, D_o, \mathbf{R}, \mathbf{t})$ in the objective function (5.2) is replaced by the term

$$E_{detector}(D_h, D_o, \mathbf{R}, \mathbf{t}) = \sum_{(X_{det}, X'_{det}) \in \mathcal{C}_{detect}} \|X'_{det} - (\mathbf{R}X_{det} + \mathbf{t})\|^2. \quad (5.8)$$

The results of the reconstruction in Figure 5.9 show that the reconstruction is either incomplete or it has major flaws, which is supported by the numbers in Table 5.2.

In order to measure the accuracy of the contact correspondences obtained by the *hand pose* or the *contact detector*, we manually annotated two points for each of the two manipulating fingers for pairs of consecutive frames and we do so for every 10th frame in our four sequences. We then measure the pairwise registration error for each annotated pair (X_{gt}, X'_{gt}) by $\|X'_{gt} - (\mathbf{R}X_{gt} + \mathbf{t})\|$. The results are summarized in Table 5.2 and show that the hand tracker is more accurate for pairwise registration than a detection based approach.

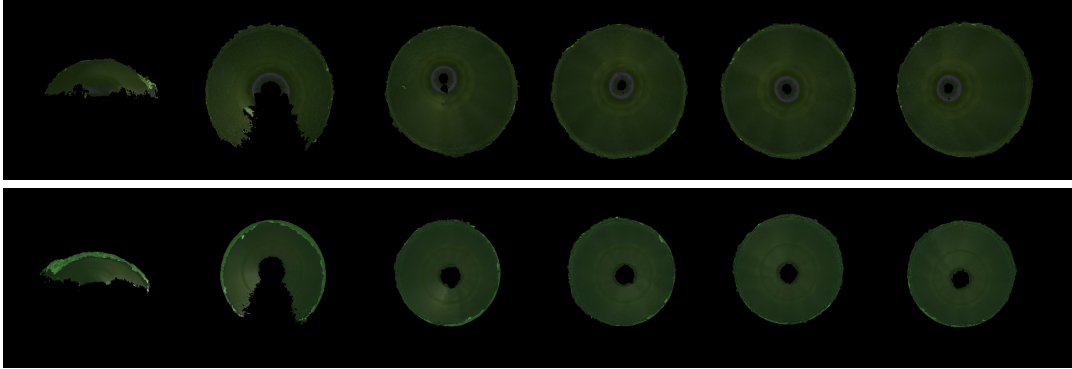


Figure 5.8: Qualitative evaluation of the weight γ_t of the energy function (5.2). The images show the reconstruction of the objects *water-bottle* and *bowling-pin* (bottom view) for the weights γ_t : 0, 1, 5, 10, 15 and 20 (from left to right).

Table 5.2: Quantitative evaluation of the pairwise registration based on annotated pairs of frames. We assess the performance of both the proposed pipeline based on hand pose $E_{contact}$ as described in Equation (5.6), as well as the detector-based baseline E_{detect} as described in Equation (5.8). We report the mean and the standard deviation over all the sampled frame pairs of all sequences in millimeters.

Energy	Mean	Std.Dev.	mm
$E_{contact} + E_{visual}$	1.67	0.95	
$E_{contact}$	1.64	0.88	
$E_{detector} + E_{visual}$	1.73	1.08	
$E_{detector}$	1.80	1.12	

5.5.2.2 Qualitative evaluation

Although the quantitative evaluation is informative, a qualitative evaluation can give further intuition about the effectiveness of the system and the influence of its parameters. We therefore show in Figure 5.8 the mesh extracted from the TSDF volume of our pipeline for a number of different values for the weight γ_t . The experiment is done for the two objects where the influence of γ_t can be easily observed visually. As expected, the reconstruction without the use of hand motion capture results in a degenerate alignment. The incorporation of *contact correspondences* immediately improves the reconstruction, driving the alignment process according to the spatiotemporal movement of the contact fingers. A low value, however, leads only to a partial reconstruction. A sensible choice seems to be a value between 10 and 30, while for bigger values some small alignment artifacts appear. For our experiments we choose $\gamma_t = 15$.

While Figure 5.9 shows the reconstruction when the hand tracker is replaced by a detector, Figure 5.10 shows the reconstruction of the *bowling-pin* when ICP, as

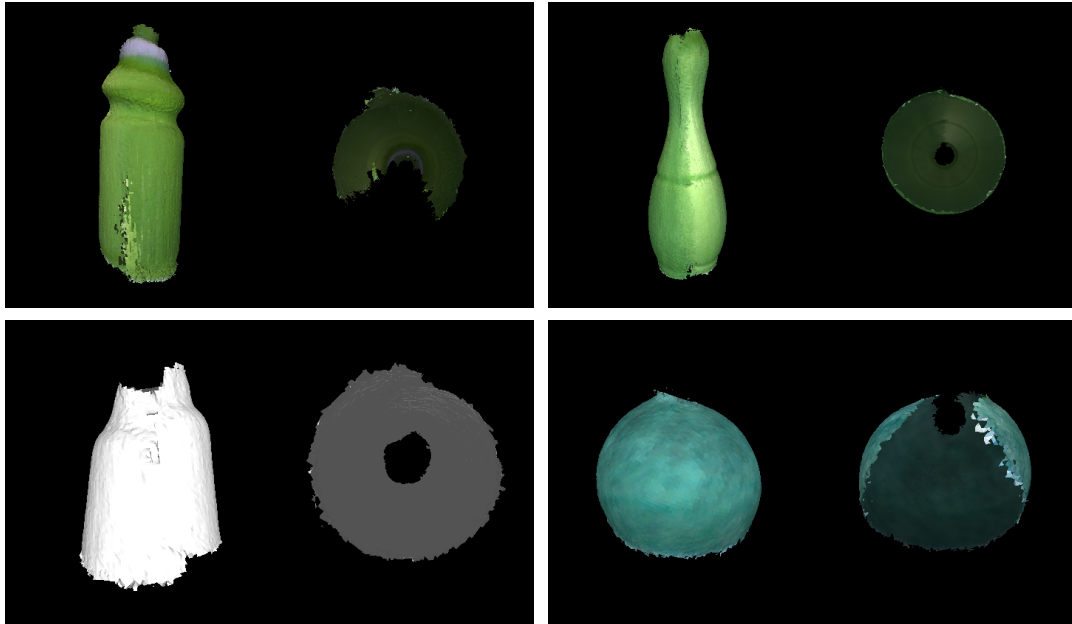


Figure 5.9: When a *contact detector* is used instead of the contact points based on a *hand tracker*, the reconstruction fails. For each object the front and the bottom view are shown.

described in Equation (5.7), is not used. In both cases, the point clouds are not well aligned.

Figure 5.11 shows the best reconstruction of three runs by *KinFu* and *Skaneet* in comparison to our pipeline, both with and without the use of hands and hand motion data. The images show that the reconstruction without hands is similar across different systems and results in a degenerate 3d representation of the object. The incorporation of hand motion capture in the reconstruction plays clearly a vital role, leading to the effective reconstruction of the full surface of the object.

Although Figure 5.11 compares the TSDF meshes, more detailed results are shown in Figure 5.12. The camera poses are reconstructed effectively, showing not only the rotational movement during the scanning process, but also the type and intensity of hand-object interaction. The water-tight meshes that are shown compose the final output of our system. The resulting reconstruction renders our approach the first in-hand scanning system to cope with the reconstruction of symmetric objects, while also showing prospects of future practical applications.

5.6 Summary

While existing in-hand scanning systems discard information originating from the hand, we have proposed an approach that successfully incorporates the 3d motion

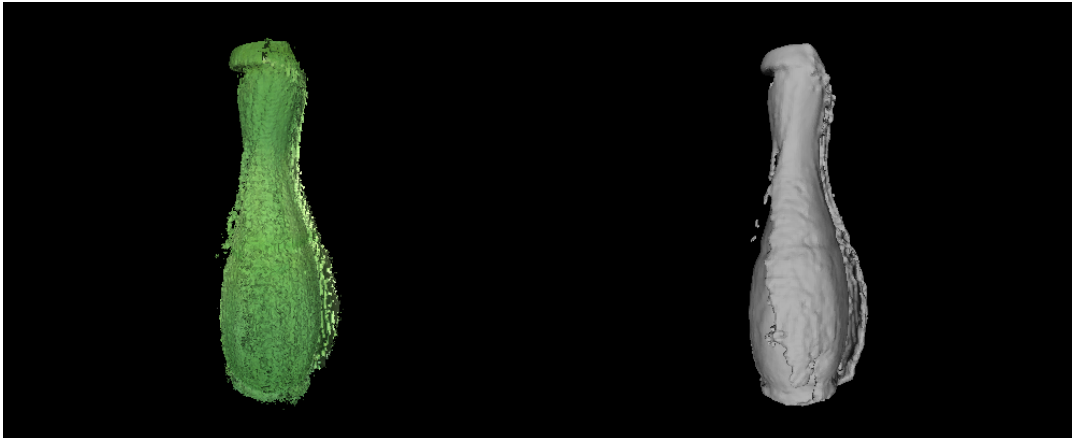


Figure 5.10: Qualitative results for the *bowling-pin* object without the term E_{icp} of Equation (5.7). In this case the point clouds are not well aligned.

information of the manipulating hand for 3d object reconstruction. In that respect, the visual correspondences based on geometric and texture features are combined with contact correspondences that are inferred from the manipulating hand. In our quantitative and qualitative experiments we show that our approach successfully reconstructs the 3d shape of four highly symmetric and textureless objects.

Reconstruction of the 3d shape alone is enough for rigid objects like the ones used in this chapter. However for articulated objects we have to further reconstruct the 3d kinematic model. In this direction in Chapter 6 we present a system that reconstructs the skeleton of an articulated object to get a fully rigged model for tracking or animation.

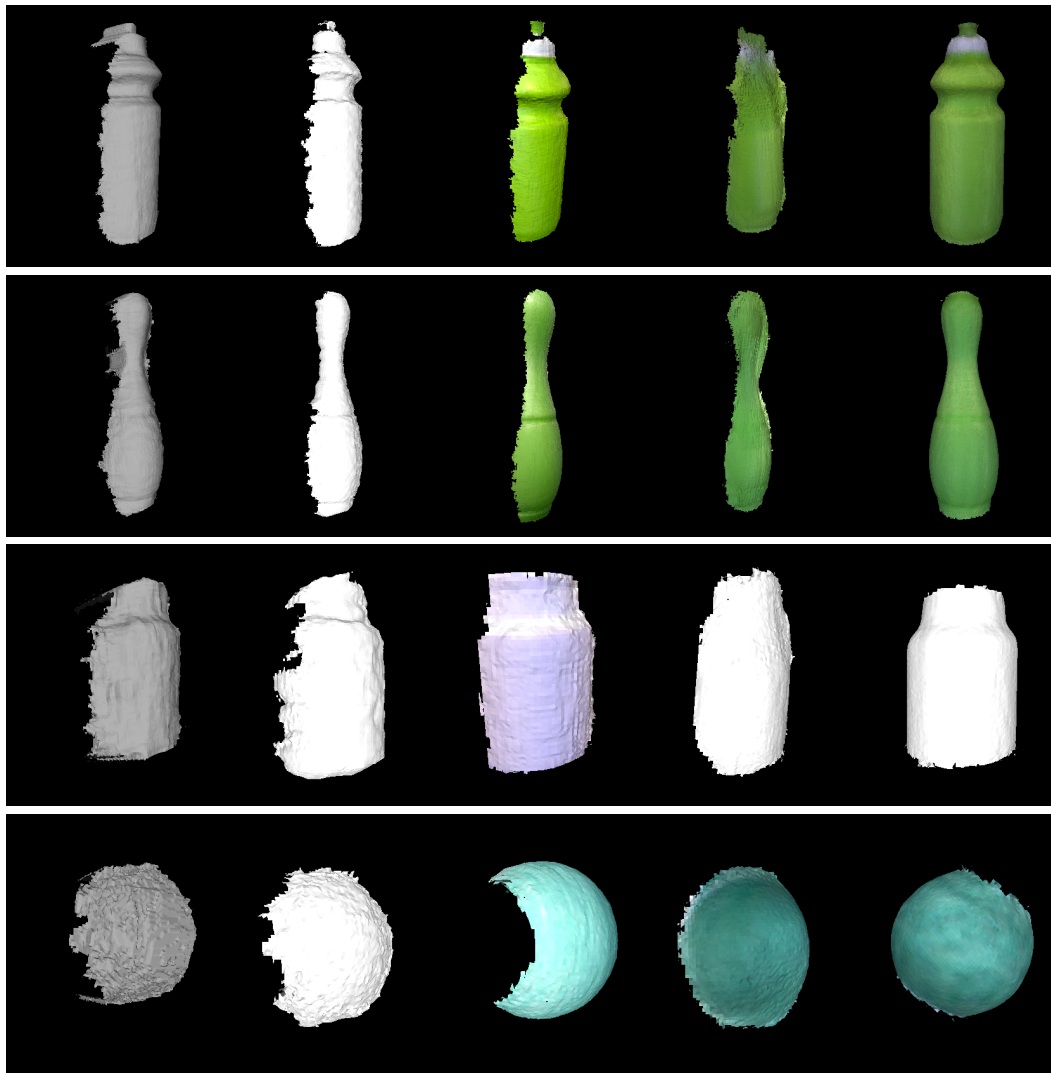


Figure 5.11: Qualitative comparison of different in-hand scanning systems for all four objects of Figure 5.5. We visualize the meshes extracted from the TSDF volume. From left to right, each row contains the result of: (a) KinFu, (b) Skanect, (c) Our pipeline with a turntable and without hand motion data, (d) Our pipeline with in-hand scanning but without hand motion data, (e) Our pipeline with in-hand scanning that includes hand motion data (the proposed setup). Only the combination of in-hand scanning with hand motion data succeeds in reconstructing all symmetric objects.

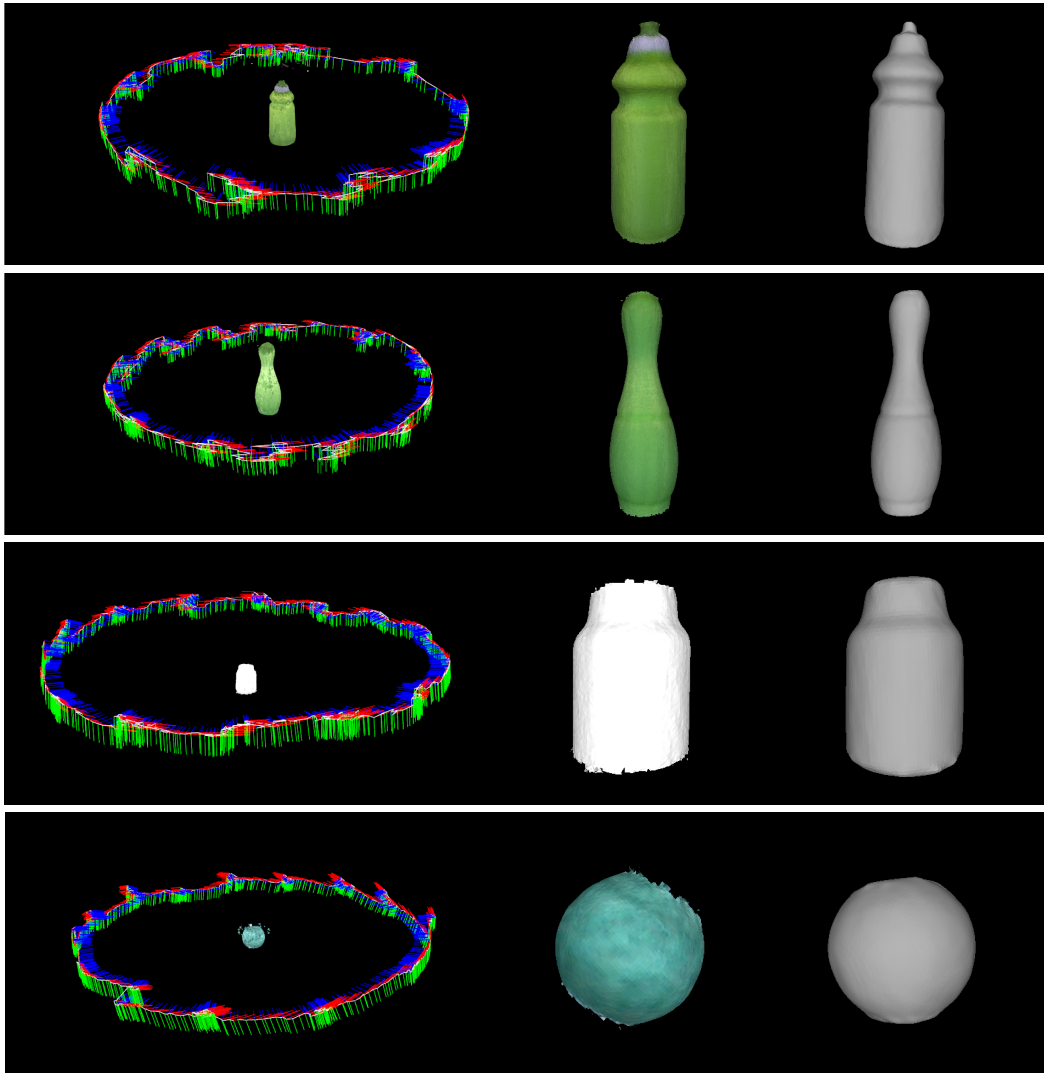


Figure 5.12: Qualitative results of our pipeline for all four objects of Figure 5.5 when a hand rotates the object in front of the camera. The left images show the reconstructed camera poses. The poses follow a circular path, whose shape signifies the type of hand-object interaction during the rotation. The middle images show the mesh that is acquired with marching cubes from the TSDF volume, while the right ones show the final water-tight mesh that is acquired with Poisson reconstruction.

Reconstructing Articulated Rigged Models from RGB-D Videos

After the advent of commodity RGB-D sensors there is a plethora of commercial and open-source software to reconstruct the 3d shape of a rigid object with rich geometric or texture features. Highly symmetric or featureless rigid objects can be reconstructed with the approach presented in Chapter 5, by tracking the hands rotating the object and incorporating the hand motion information in the reconstruction pipeline. However, there is a lack of tools that build rigged models of articulated objects that deform realistically and can be used for tracking or animation.

In this chapter, we fill this gap and propose a method that creates a fully rigged model of an articulated object from depth data of a single sensor. To this end, we combine deformable mesh tracking, motion segmentation based on spectral clustering and skeletonization based on mean curvature flow. The fully rigged model then consists of a watertight mesh, embedded skeleton, and skinning weights.

Contents

6.1	Introduction	81
6.2	Related work	82
6.3	Mesh motion	83
6.3.1	Preprocessing	84
6.3.2	Mesh tracking	84
6.4	Kinematic model acquisition	85
6.4.1	Motion segmentation	86
6.4.2	Kinematic topology	87
6.5	Experiments	89
6.6	Summary	92

6.1 Introduction

With the increasing popularity of depth cameras, the reconstruction of rigid scenes or objects at home has become affordable for any user [Newcombe et al., 2011] and

together with 3d printers allows novel applications [Sturm et al., 2013]. Many objects, however, are non-rigid and their motion can be modeled by an articulated skeleton. Although articulated models are highly relevant for computer graphic applications [Baran and Popović, 2007] including virtual or augmented reality and robotic applications [Pillai et al., 2014], there is no approach that builds from a sequence of depth data a fully rigged 3d mesh with a skeleton structure that describes the articulated deformation model.

In the context of computer graphics, methods for automatic rigging have been proposed. In [Baran and Popović, 2007], for instance, the geometric shape of a static mesh is used to fit a predefined skeleton into a static mesh. More detailed human characters including cloth simulation have been reconstructed from multi-camera video data in [Stoll et al., 2010]. Both approaches, however, assume that the skeleton structure is given. On the contrary, the skeleton structure can be estimated from high-quality mesh animations [De Aguiar et al., 2008]. The approach, however, cannot be applied to depth data. At the end, we have a typical chicken-and-egg problem. If a rigged model with predefined skeleton is given the mesh deformations can be estimated accurately [Liu et al., 2013] and if the mesh deformations are known the skeleton structure can be estimated [De Aguiar et al., 2008].

In this chapter, we propose an approach to address this dilemma and create a fully rigged model from depth data of a single sensor. To this end, we first create a static mesh model of the object. We then reconstruct the motion of the mesh in a sequence captured with a depth sensor by deformable mesh tracking. Standard tracking, however, fails since it maps the entire mesh to the visible point cloud. As a result, the object is squeezed as shown in Figure 6.4. We therefore reduce the thinning artifacts by a strong regularizer that prefers smooth mesh deformations. Although the regularizer also introduces artifacts by oversmoothing the captured motion, in particular at joint positions as shown for the pipe sequence in Figure 6.1, the mesh can be segmented into meaningful parts by spectral clustering based on the captured mesh motion as shown in Figure 6.5. The skeleton structure consisting of bones and limbs is then estimated based on the mesh segments and mean curvature flow.

As a result, our approach is the first method that creates a fully rigged model of an articulated object consisting of a watertight mesh, embedded skeleton, and skinning weights from depth data. Such models can be used for animation, virtual or augmented reality, or in the context robot-object manipulation. We perform a quantitative evaluation with five objects of varying size and deformation characteristics and provide a thorough analysis of the parameters.

6.2 Related work

Reconstructing articulated objects has attracted a lot of interest during the past decade. Due to the popularity of different image sensors over the years, research focus has gradually shifted from reconstructing 2d skeletons from RGB data [Yan and Pollefeys, 2006, 2008, Ross et al., 2010, Chang and Demiris, 2015] to 3d skeletons from

RGB [Fayad et al., 2011, Sturm et al., 2009, 2011, Yücer et al., 2015] or RGB-D data [Katz et al., 2013, Pillai et al., 2014, Martín-Martín et al., 2016].

A popular method for extracting 2d skeletons from videos uses a factorization-based approach for motion segmentation. In [Yan and Pollefeys, 2006, 2008] articulated motion is modeled by a set of independent motion subspaces and the joint locations are obtained from the intersections of connected motion segments. A probabilistic graphical model has been proposed in [Ross et al., 2010]. The skeleton structure is inferred from 2d feature trajectories by maximum likelihood estimation and the joints are located in the center of the motion segments. Recently, [Chang and Demiris, 2015] combine a fine-to-coarse motion segmentation based on iterative randomized voting with a distance function based on contour-pruned skeletonization. The kinematic model is inferred with a minimum spanning tree approach.

In order to obtain 3d skeletons from RGB videos, structure-from-motion (SfM) approaches can be used. [Fayad et al., 2011] perform simultaneous segmentation and sparse 3d reconstruction of articulated motion with a cost function minimizing the re-projection error of sparse 2d features, while a spatial prior favors smooth segmentation. The method is able to compute the number of joints and recover from local minima, while occlusions are handled by incorporating partial sequences into the optimization. In contrast to [Tresadern and Reid, 2005], it is able to reconstruct complex articulated structures. [Yücer et al., 2015] use ray-space optimization to estimate 3d trajectories from 2d trajectories. The approach, however, assumes that the number of parts are known. In [Sturm et al., 2009, 2011] markers are attached to the objects to get precise 3d pose estimations of object parts. They use a probabilistic approach with a mixture of parametrized and parameter-free representations based on Gaussian processes. The skeleton structure is inferred by computing the minimum spanning tree over all connected parts.

The recent advances in RGB-D sensors allow to work fully in 3d. An early approach [Katz et al., 2013] uses sparse KLT and SIFT features and groups consistent 3d trajectories with a greedy approach. The kinematic model is inferred by sequentially fitting a prismatic and a rotational joint with RANSAC. In [Pillai et al., 2014] the 3d trajectories are clustered by density-based spatial clustering. For each cluster, the 3d pose is estimated and the approach [Sturm et al., 2011] is applied to infer the skeleton structure. Recently, [Martín-Martín et al., 2016] presented a method that combines shape reconstruction with the estimation of the skeleton structure. While these approaches operate only with point clouds, our approach generates fully rigged models consisting of a watertight mesh, embedded skeleton, and skinning weights.

6.3 Mesh motion

Our approach consists of three steps. We first create a watertight mesh of the object using a depth sensor that is moved around the object while the object is not moving. Creating meshes from static objects can be done with standard software. In our experiments, we use Skanect [SKANECT] with optional automatic mesh cleaning using

MeshLab [MESHLAB]. In the second step, we record a sequence where the object is deformed by hand-object interaction and track the mesh to obtain the mesh motion. In the third step, we estimate the skeleton structure and rig the model. The third step will be described in Section 6.4.

6.3.1 Preprocessing

For tracking, we preprocess each frame of the RGB-D sensor. We first discard points that are far away and only keep points that are within a 3d volume. This is actually not necessary but it avoids unnecessary processing like normal computation for irrelevant points. Since the objects are manipulated by hands, we discard the hands by skin color segmentation on the RGB image using a Gaussian mixtures model (GMM) [Jones and Rehg, 2002]. The remaining points are then smoothed by a bilateral filter [Paris and Durand, 2009] and normals are computed as in [Holzer et al., 2012].

6.3.2 Mesh tracking

For mesh tracking, we capitalize on a Laplacian deformation framework similar to [Botsch and Sorkine, 2008]. While in [Gall et al., 2009, Liu et al., 2013] a Laplacian deformation framework was combined with skeleton-based tracking in the context of a multi-camera setup, we use the Laplacian deformation framework directly for obtaining the mesh motion of an object with unknown skeleton structure. Since we use only one camera and not an expensive multi-camera setup, we observe only a portion of the object and the regularizer will be very important as we will show in the experiments.

For mesh tracking, we align the mesh \mathcal{M} with the preprocessed depth data D by minimizing the objective function

$$E(\mathcal{M}, D) = \mathcal{E}_{smooth}(\mathcal{M}) + \gamma_{def} \left(\mathcal{E}_{model \rightarrow data}(\mathcal{M}, D) + \mathcal{E}_{data \rightarrow model}(\mathcal{M}, D) \right). \quad (6.1)$$

The objective function consists of a smoothness term \mathcal{E}_{smooth} that preserves geometry by penalizing changes in surface curvature, as well as two data terms $\mathcal{E}_{model \rightarrow data}$ and $\mathcal{E}_{data \rightarrow model}$ that align the mesh model to the observed data and the data to the model, respectively. The impact of the smoothness term and the data terms is steered by the parameter γ_{def} .

For the data terms, we use the same terms that are used for articulated hand tracking in Chapter 4. For the first term

$$\mathcal{E}_{model \rightarrow data}(\mathcal{M}, D) = \sum_i \|\mathbf{V}_i - \mathbf{X}_i\|_2^2 \quad (6.2)$$

we establish correspondences between the visible vertices \mathbf{V}_i of the mesh \mathcal{M} and the closest points \mathbf{X}_i of the point cloud D and minimize the distance. We discard correspondences for which the angle between the normals of the vertex and the closest point is larger than 45° or the distance between the points is larger than 10 mm.

The second data term

$$\mathcal{E}_{data \rightarrow model}(\mathcal{M}, D) = \sum_i \|\mathbf{V}_i \times \mathbf{d}_i - \mathbf{m}_i\|_2^2 \quad (6.3)$$

minimises the distance between a vertex \mathbf{V}_i and the projection ray of a depth discontinuity observed in the depth image. To compute the distance, the projection ray of a 2d point is expressed by a Plücker line [Pons-Moll and Rosenhahn, 2011] with direction \mathbf{d}_i and moment \mathbf{m}_i . The depth discontinuities are obtained as in Chapter 4 by an edge detector applied to the depth data and the correspondence between a depth discontinuity and a vertex are obtained by searching the closest projected vertex for each depth discontinuity.

Due to the partial view of the object, the data terms are highly under-constrained. This is compensated by the smoothness term that penalizes changes of the surface curvature [Botsch and Sorkine, 2008]. The term can be written as

$$\mathcal{E}_{smooth}(\mathcal{M}) = \sum_i \|\mathbf{L}\mathbf{V}_i - \mathbf{L}\mathbf{V}_{i,t-1}\|_2^2 \quad (6.4)$$

where $\mathbf{V}_{i,t-1}$ is the previous vertex position. In order to model the surface curvature, we employ the cotangent Laplacian [Botsch and Sorkine, 2008] matrix \mathbf{L} given by

$$L_{ij} = \begin{cases} \sum_{\mathbf{V}_k \in \mathcal{N}_1(\mathbf{V}_i)} w_{ik} & , i = j \\ -w_{ij} & , \mathbf{V}_j \in \mathcal{N}_1(\mathbf{V}_i) \\ 0 & , \text{otherwise} , \end{cases} \quad \text{where} \quad w_{ij} = \frac{1}{2|A_i|} (\cot \alpha_{ij} + \cot \beta_{ij}) \quad (6.5)$$

where $\mathcal{N}_1(\mathbf{V}_i)$ denotes the set of one-ring neighbor vertices of vertex \mathbf{V}_i . The weight w_{ij} for an edge in the triangular mesh between two vertices \mathbf{V}_i and \mathbf{V}_j depends on the cotangents of the two angles α_{ij} and β_{ij} opposite of the edge (i, j) and the size of the Voronoi cell $|A_i|$ that is efficiently approximated by half of the sum of the triangle areas defined by $\mathcal{N}_1(\mathbf{V}_i)$.

We minimize the least squares problem (6.1) by solving a large but highly sparse linear system using sparse Cholesky decomposition. For each frame, we use the estimate of the previous frame for initialization.

6.4 Kinematic model acquisition

After having estimated the mesh motion as described in Section 6.3, we have for each vertex the trajectory \mathcal{T}_i . We use the trajectories together with the shape of the mesh \mathcal{M} to reconstruct the underlying skeleton. To this end, we first segment the trajectories as described in Section 6.4.1 and then infer the skeleton structure, which will be explained in Section 6.4.2.

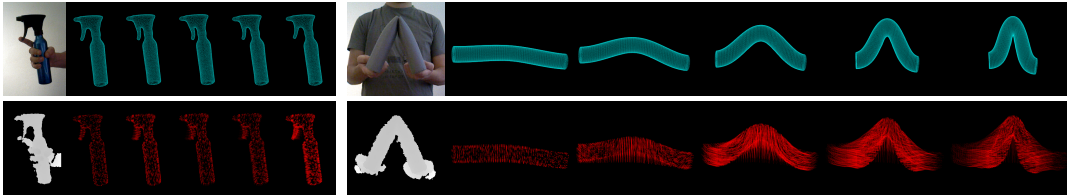


Figure 6.1: Tracked mesh with the deformable tracker presented in Section 6.3.2 and the corresponding 3d vertex trajectories. We present images for the sequences “spray” and “pipe 1/2” showing the temporal evolution at 20%, 40%, 60%, 80% and 100% of the sequence.

6.4.1 Motion segmentation

In contrast to feature based trajectories, the mesh motion provides trajectories of the same length and a trajectory for each vertex, even if the vertex has never been observed in the sequence due to occlusions. This means that clustering the trajectories also segments the mesh into rigid parts.

Similar to 2d motion segmentation approaches for RGB videos [Brox and Malik, 2010], we define an affinity matrix based on the 3d trajectories and use spectral clustering for motion segmentation. The affinity matrix

$$\Phi_{ij} = \exp(-\lambda d(\mathcal{T}_i, \mathcal{T}_j)) \quad (6.6)$$

is based on the pairwise distance between two trajectories \mathcal{T}_i and \mathcal{T}_j . $\Phi_{ij} = 1$ if the trajectories are the same and close to zero if the trajectories are very dissimilar. As in [Brox and Malik, 2010], we use $\lambda = 0.1$.

To measure the distance between two trajectories \mathcal{T}_i and \mathcal{T}_j , we measure the distance change of two vertex positions \mathbf{V}_i and \mathbf{V}_j within a fixed time interval. We set the length of the time interval proportional to the observed maximum displacement, i.e.

$$dt = 2 \max_{i,t} \|\mathbf{V}_{i,t} - \mathbf{V}_{i,t-1}\|_2. \quad (6.7)$$

Since the trajectories are smooth due to the mesh tracking as described in Section 6.3.2, we do not have to deal with outliers and we can take the maximum displacement over all vertices. The object, however, might be deformed only at a certain time interval of the entire sequence. We are therefore only interested in the maximum distance change over all time intervals, i.e.

$$d^v(\mathcal{T}_i, \mathcal{T}_j) = \max_t \left| \|\mathbf{V}_{i,t} - \mathbf{V}_{j,t}\|_2 - \|\mathbf{V}_{i,t-dt} - \mathbf{V}_{j,t-dt}\|_2 \right|. \quad (6.8)$$

This means that if two vertices belong to the same rigid part, the distance between them should not change much over time. In addition, we take the change of the angle between the vertex normals \mathbf{N} into account. This is measured in the same way as

maximum over the intervals

$$d^n(\mathcal{T}_i, \mathcal{T}_j) = \max_t \left| \arccos(\mathbf{N}_{i,t}^T \mathbf{N}_{j,t}) - \arccos(\mathbf{N}_{i,t-dt}^T \mathbf{N}_{j,t-dt}) \right|. \quad (6.9)$$

The two distance measures are combined by

$$d(\mathcal{T}_i, \mathcal{T}_j) = (1 + d^n(\mathcal{T}_i, \mathcal{T}_j)) d^v(\mathcal{T}_i, \mathcal{T}_j). \quad (6.10)$$

The distances are measured in mm and the angles in rad. Adding 1 to d^n was necessary since d^n can be close to zero despite of large displacement changes.

Based on (6.6), we build the normalized Laplacian graph [Ng et al., 2002]

$$\mathcal{L} = D^{-\frac{1}{2}}(D - \Phi)D^{-\frac{1}{2}} \quad (6.11)$$

where D is an $n \times n$ diagonal matrix with

$$D_{ii} = \sum_j \Phi_{ij} \quad (6.12)$$

and perform eigenvalue decomposition of \mathcal{L} to get the eigenvalues $\lambda_1, \dots, \lambda_n$, ($\lambda_1 \leq \dots \leq \lambda_n$), as well as the corresponding eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_n$. The number of clusters k is determined by the number of eigenvalues below a threshold λ_{thresh} and the final clustering of the trajectories is then obtained by k -means clustering Ng et al. [2002] on the rows of the $n \times k$ matrix

In practice, we sample uniformly 1000 vertices from the mesh to compute the affinity matrix. This turned out to be sufficient while reducing the time to compute the matrix. For each vertex that has not been sampled, we compute the closest sampled vertex on the mesh and assign it to the same cluster. This results in a motion segmentation of the entire mesh as shown in Figure 6.5.

6.4.2 Kinematic topology

Given the segmented mesh, it remains to determine the joint positions and topology of the skeleton. To obtain a bone structure, we first skeletonize the mesh by extracting the mean curvature skeleton (MCS) based on the mean curvature flow [Tagliasacchi et al., 2012] that captures effectively the topology of the mesh by iteratively contracting the triangulated surface. The red 3d curve in Figure 6.2 shows the mean curvature skeleton for an object. In order to localize the joints, we compute the intersecting boundary of two connected mesh segments using a half-edge representation. For each intersecting pair of segments, we compute the centroid of the boundary vertices and find its closest 3d point on the mean curvature skeleton. In this way, the joints are guaranteed to lie inside the mesh. In order to create the skeleton structure with bones, we first create auxiliary joints without any degree of freedom at the points where the mean curvature skeleton branches or ends as shown in Figure 6.2. After all 3d joints on the skeleton are determined, we follow the mean curvature skeleton and connect the detected joints accordingly to build a hierarchy of bones that defines the topology

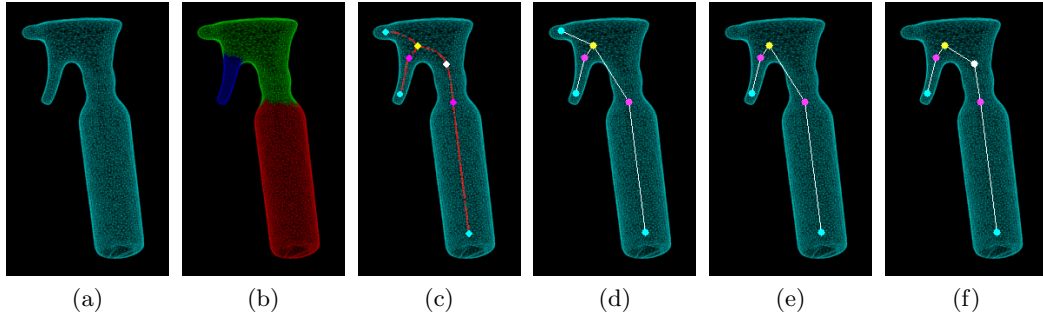


Figure 6.2: The steps of our pipeline. (a) *Initial mesh* (b) *Motion segments* (c) *Mean curve skeleton* where the endpoints are shown with cyan, the junction points with yellow, the virtual point due to collision with white and the motion joints with magenta (d) *Initial skeleton* (e) *Refined skeleton* after removal of redundant bone (f) *Final skeleton* after replacement of the colliding bone with two non-colliding ones and a virtual joint.

Algorithm 2: Overview of the steps of our algorithm.

Deformable motion capture

- └ - Perform *deformable* tracking of the object Section 6.3.2 - Eq. (6.1)

Motion segmentation of the object

- └ - Generate dense vertex *trajectories* from tracking result Section 6.4.1
- └ - Sample 1000 trajectories for tractability Section 6.4.1
- └ - Build an *affinity matrix* of vertex trajectories Section 6.4.1 - Eq. (6.6-6.10)
- └ - Segment mesh by *spectral clustering* Section 6.4.1 - Eq. (6.11-6.12)

Kinematic model acquisition for the object

- └ - Infer *joints* at intersections of mesh segments Section 6.4.2
 - └ - Infer *skeleton topology* Section 6.4.2
 - └ - Compute *skinning weights* Section 6.4.2
-

of a skeleton structure.

Although the number of auxiliary joints usually does not matter, we reduce the number of auxiliary joints and irrelevant bones by removing bones that link an endpoint with another auxiliary joint if they belong to the same motion segment. The corresponding motion segment for each joint can be directly computed from the mean curvature flow [Tagliasacchi et al., 2012]. We finally ensure that each bone is inside the mesh. To this end, we detect bones colliding with the mesh with a collision detection approach based on bounding volume hierarchies. We then subdivide each colliding bone in two bones by adding an additional auxiliary joint at the middle of the mean curvature skeleton that connects the endpoints of the colliding bone. The process is repeated until all bones are inside the mesh. In our experiments, however, one iteration was enough. This procedure defines the refined topology of the skeleton that is already embedded in the mesh. The skinning weights are then computed as in [Baran and Popović, 2007].

As a result, we obtain a fully rigged model consisting of a watertight mesh, an

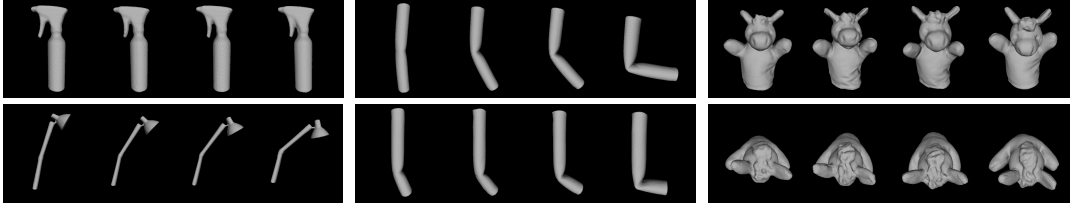


Figure 6.3: Each object is scanned in four target poses with escalating difficulty and pose estimation from an initial state is performed for evaluation while spanning the parameter space of $(\gamma_{def}, \lambda_{thresh})$. For the “donkey” object both a front and a top view is presented.

embedded skeleton structure, and skinning weights. The entire steps of the approach are summarized in Algorithm 2. Results for a few objects are shown in Figure 6.5.

6.5 Experiments

We quantitatively evaluate our approach for five different objects shown in Table 6.1, the “spray”, the “donkey”, the “lamp”, as well as the “pipe 1/2” and “pipe 3/4” which have a joint at 1/2 and 3/4 of their length, respectively. We acquire a 3d template mesh using the commercial software *skanect* [SKANECT] for the first three, while for the pipe we use the public template model presented in Chapter 4. All objects have the same number of triangles, so the average triangle size varies from $3.7mm^2$ for the “spray”, 13.8 for the “donkey”, 24.8 for the “lamp” and 4.4 for the “pipe” models. We captured sequences of the objects while deforming them using a Primesense Carmine 1.09 RGB-D sensor. The recorded sequences, calibration data, scanned 3d models, deformable motion data, as well as the resulting models and respective videos will be publicly available.

We perform deformable tracking (Section 6.3.2) to get 3d dense vertex trajectories as depicted in Figure 6.1. Deformable tracking depends on the weight γ_{def} in the objective function (6.1) that steers the influence of the smoothness and data terms. As depicted in Figure 6.4, a very low γ_{def} gives too much weight to the smoothness term and prevents an accurate fitting to the input data, while a big γ_{def} results in over-fitting to the partial visible data and a strong thinning effect can be observed. The thinning gets more intense for an increasing γ_{def} .

Despite of γ_{def} , our approach also depends on the eigenvalue threshold λ_{thr} for spectral clustering. To study the effect of the parameters, we created a test dataset. For each object, we scanned the objects in four different poses. To this end, we fixed the object in a pose with adhesive tape and reconstructed it by moving the camera around the object. The target poses of the objects are shown in Figure 6.3. To measure the quality of a rigged model for a parameter setting, we align the model $\mathcal{M}(\theta)$ parametrized by the rotations of the joints and the global rigid transformation to the reconstructed object \mathcal{O} from an initial pose. For the alignment, we use only the

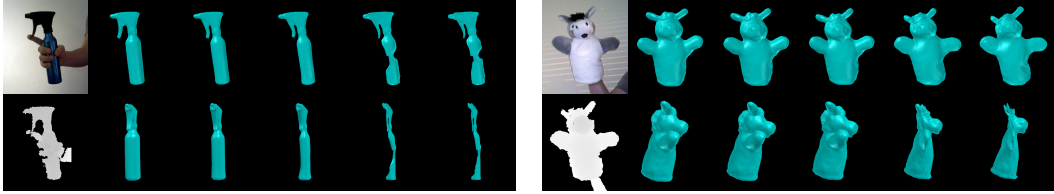


Figure 6.4: Deformable tracking for $\gamma_{def} = 0.001, 0.005, 0.01, 0.05, 0.1$ (from left to right) that steers the influence of the smoothness and data terms in Equation (6.1). We depict the front (top) and side view (bottom) for the last frame of the sequences “spray” and “donkey”.

inferred articulated model, i.e. we estimate the rigid transformation and the rotations of the joints of the inferred skeleton. As data term, we use

$$\frac{1}{|\mathcal{M}(\theta)| + |\mathcal{O}|} \left(\sum_{\mathbf{v}(\theta) \in \mathcal{M}(\theta)} \|\mathbf{V}(\theta) - \mathbf{v}_{\mathcal{O}}\|_2^2 + \sum_{\mathbf{v}_{\mathcal{O}} \in \mathcal{O}} \|\mathbf{v}_{\mathcal{O}} - \mathbf{V}(\theta)\|_2^2 \right). \quad (6.13)$$

based on the closest vertices from mesh $\mathcal{M}(\theta)$ to \mathcal{O} and vice versa. This measure is also used to measure the 3d error in mm after alignment.

Table 6.1 summarizes the average 3d vertex error for various parameter settings, with the highlighted values indicating the best qualitative results for each object, while Figure 6.5 shows the motion segments and the acquired skeletons for the best configuration. The optimal parameter γ_{def} seems to depend on the triangle size since the smoothness term is influenced by the areas of the Voronoi cells $|A_i|$ (6.5) and therefore by the areas of the triangles. The objects “Donkey” and “Lamp” have *large triangles* ($> 10mm^2$) and prefer $\gamma_{def} = 0.05$, while the objects with small triangles ($< 10mm^2$) perform better for $\gamma_{def} = 0.005$. Spectral clustering on the other hand works well for $\lambda_{thr} = 0.7$ when *reasonably sized parts* undergo a *pronounced movement*, however, a higher value of $\lambda_{thr} = 0.98$ is better for *small parts* undergoing a *small motion* compared to the size of the object like the handle of the “spray”. As shown in Figure 6.6, a high threshold results in an over-segmentation and increases the number of joints. An over-segmentation is often acceptable as we see for example in Figure 6.2b or in Figure 6.5 for the “spray” and the “lamp”. In general, a slight over-segmentation is not problematic for many applications since joints can be disabled or ignored for instance for animation. A slight increase of the degrees of freedom also does not slow down articulated pose estimation, it even yields sometimes a lower alignment error as shown in Table 6.1

We also evaluated our method on the public sequences “*Bending a Pipe*” and “*Bending a Rope*” of Chapter 4, in which the skeleton was manually modeled with 1 and 35 joints, respectively. As input we use the provided mesh of each object and the RGB-D sequences to infer the skeleton. We use the tracked object meshes of Chapter 4 as ground-truth and measure the error as in (6.13), but averaged over all frames. We first evaluate the accuracy of the deformable tracking in Table 6.2, which performs

Table 6.1: Evaluation of our approach using the target poses in Figure 6.3. We create a rigged model while spanning the parameter space for the deformable tracking weight γ_{def} and the spectral clustering threshold λ_{thr} . The rigged model is aligned to the target poses by articulated pose estimation. We report the average vertex error in *mm*.


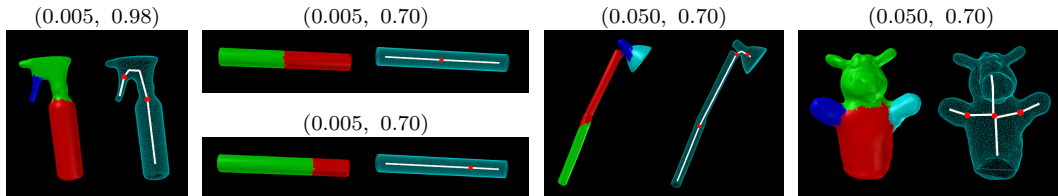
		λ_{thr}	0.40	0.50	0.60	0.70	0.80	0.90	0.95	0.98			λ_{thr}	0.40	0.50	0.60	0.70	0.80	0.90	0.95	0.98		
		γ_{def}												γ_{def}									
Spray	0.001	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9	Lamp	0.001	12.9	12.9	12.9	12.9	12.9	12.9	11.8	11.8			
	0.005	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.4		0.005	8.2	6.1	6.0	4.7	5.1	4.9	4.6	4.6			
	0.01	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.4		0.01	6.0	6.0	4.6	5.0	5.0	4.7	4.7	4.6			
	0.05	1.9	1.9	1.9	1.9	1.9	1.9	1.5	1.5	0.05		11.8	4.7	4.7	4.7	4.7	4.7	5.2	4.8				
	0.1	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9	0.1		12.6	12.8	5.2	5.3	4.7	4.7	4.6	4.6				
Pipe 1/2	0.001	10.0	2.4	2.4	2.4	4.5	3.4	3.3	3.6	Donkey	0.001	6.7	6.7	6.7	6.7	6.7	6.7	6.7	6.7				
	0.005	2.4	2.4	2.4	2.4	2.4	4.6	3.8	2.6		0.005	6.7	6.7	6.7	6.7	6.7	6.7	6.7	5.7				
	0.01	2.7	2.7	2.7	4.7	3.4	3.7	4.3	4.4		0.01	6.7	6.7	6.7	6.7	5.8	5.8	4.8	4.1				
	0.05	2.6	2.6	3.5	2.7	3.6	3.6	3.6	3.6		0.05	4.6	5.1	5.0	4.5	4.4	3.9	3.6	3.6				
	0.1	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0		0.1	6.3	5.1	5.0	5.1	3.8	4.0	4.0	4.0				
Pipe 3/4	0.001	8.3	5.1	5.1	5.1	2.5	3.0	2.8	2.4														
	0.005	2.4	2.4	2.4	2.4	3.6	2.5	2.6	2.4														
	0.01	2.4	2.4	2.4	2.4	2.8	2.4	2.4	2.4														
	0.05	8.3	8.3	8.3	8.3	8.3	8.3	8.3	8.3														
	0.1	8.3	8.3	8.3	8.3	8.3	8.3	8.3	8.3														

Figure 6.5: Results for the best configuration $(\gamma_{def}, \lambda_{thr})$ for each object. The images show the motion segments and the inferred 3d skeleton, where the joints with DoF are depicted with red color.

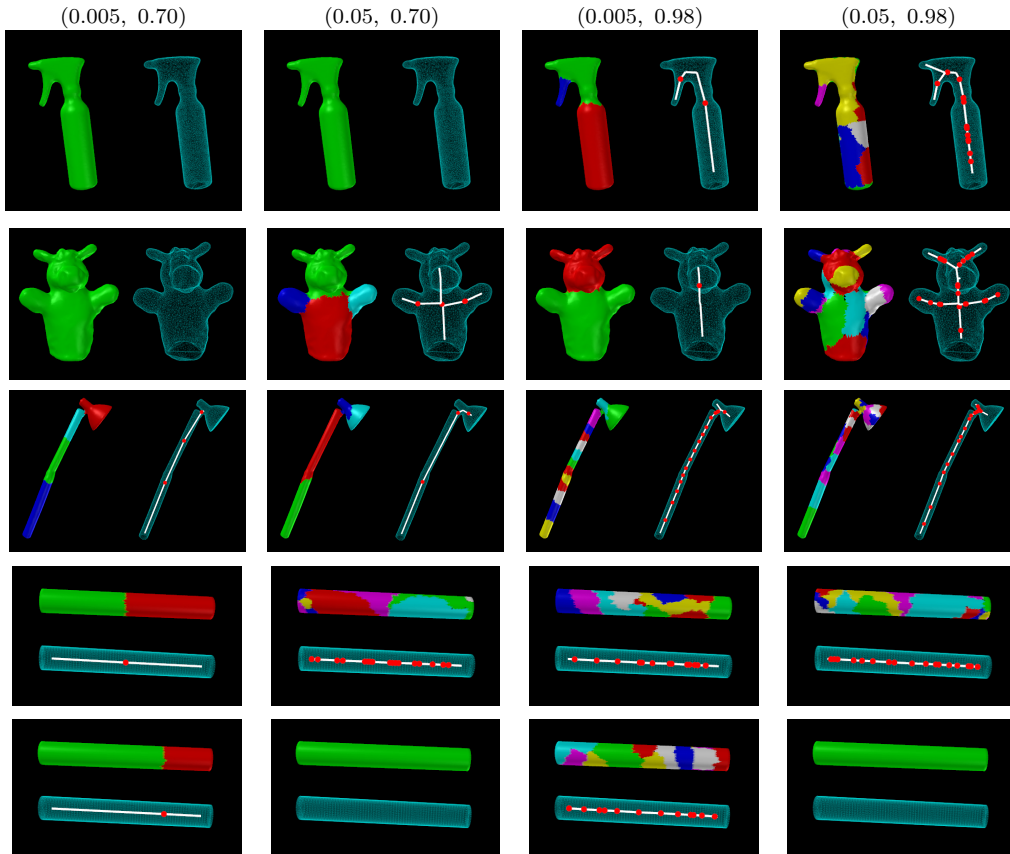


best with $\gamma_{def} = 0.005$ as in the previous experiments. If we track the sequence with the inferred articulated model using a point-to-plane metric as in Chapter 4, the error decreases. While the best spectral clustering threshold λ_{thr} for the pipe is again 0.70, the rope performs best for 0.98 due to the small size of the motion segments and the smaller motion differences of neighboring segments. We also report the error when the affinity matrix is computed only based on d^v without d^n (6.10). This slightly increases the error for the pipe with optimal parameters. The motion segments and the acquired skeletons for the best configurations are also depicted in Table 6.2.

The supplementary video¹ shows for each object the results of the deformable tracking, which is used to construct the rigged articulated models, as well as the

¹<https://youtu.be/EfG61jPK7qs>

Figure 6.6: Results for the four configurations $(\gamma_{def}, \lambda_{thr})$ that arise from the proposed parameters. The images show for each object the motion segments and the inferred 3d skeleton, where the joints with DoF are depicted with red color.



results of articulated tracking with the reconstructed skeleton. Furthermore, Figures 6.7 - 6.16 present results similar to Figures 6.5 and 6.6 for additional parameter pairs $(\gamma_{def}, \lambda_{thr})$.

6.6 Summary

We presented an approach that generates fully rigged models consisting of a watertight mesh, an embedded skeleton and skinning weights that can be used out of the box for articulated tracking or animation. In that respect we operate fully in 3d capitalizing on deformable tracking, spectral clustering and skeletonization based on mean curvature flow. The thorough evaluation of the parameters provides a valuable intuition about the important factors and opens up possibilities for further generalization in future work. Furthermore, we have shown in our experiments that the proposed approach generates nicely working rigged models and has prospects for future practical applications.

Table 6.2: Evaluation of our method and resulting kinematic models for the public sequences “Bending a Pipe” and “Bending a Rope” of Chapter 4. We report the average vertex error in mm .

		λ_{thr}					
		γ_{def}		0.70	0.98	0.70	0.98
Pipe	0.005	2.6	26.7	2.9	22.1	4.5	
	0.05	12.6	12.6	12.7	12.7	15.9	
		articulated with d^n		articulated without d^n		deform.	



		λ_{thr}					
		γ_{def}		0.70	0.98	0.70	0.98
Rope	0.005	2.5	1.1	2.4	1.1	2.6	
	0.05	141.0	141.0	193.8	193.8	nan	
		articulated with d^n		articulated without d^n		deform.	

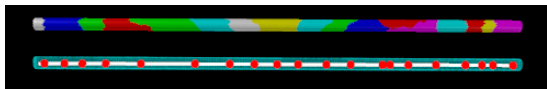


Figure 6.7: Results for all configurations (γ_{def} , λ_{thr}) spanning the parameter space. The images show for the object “spray” the motion segments.

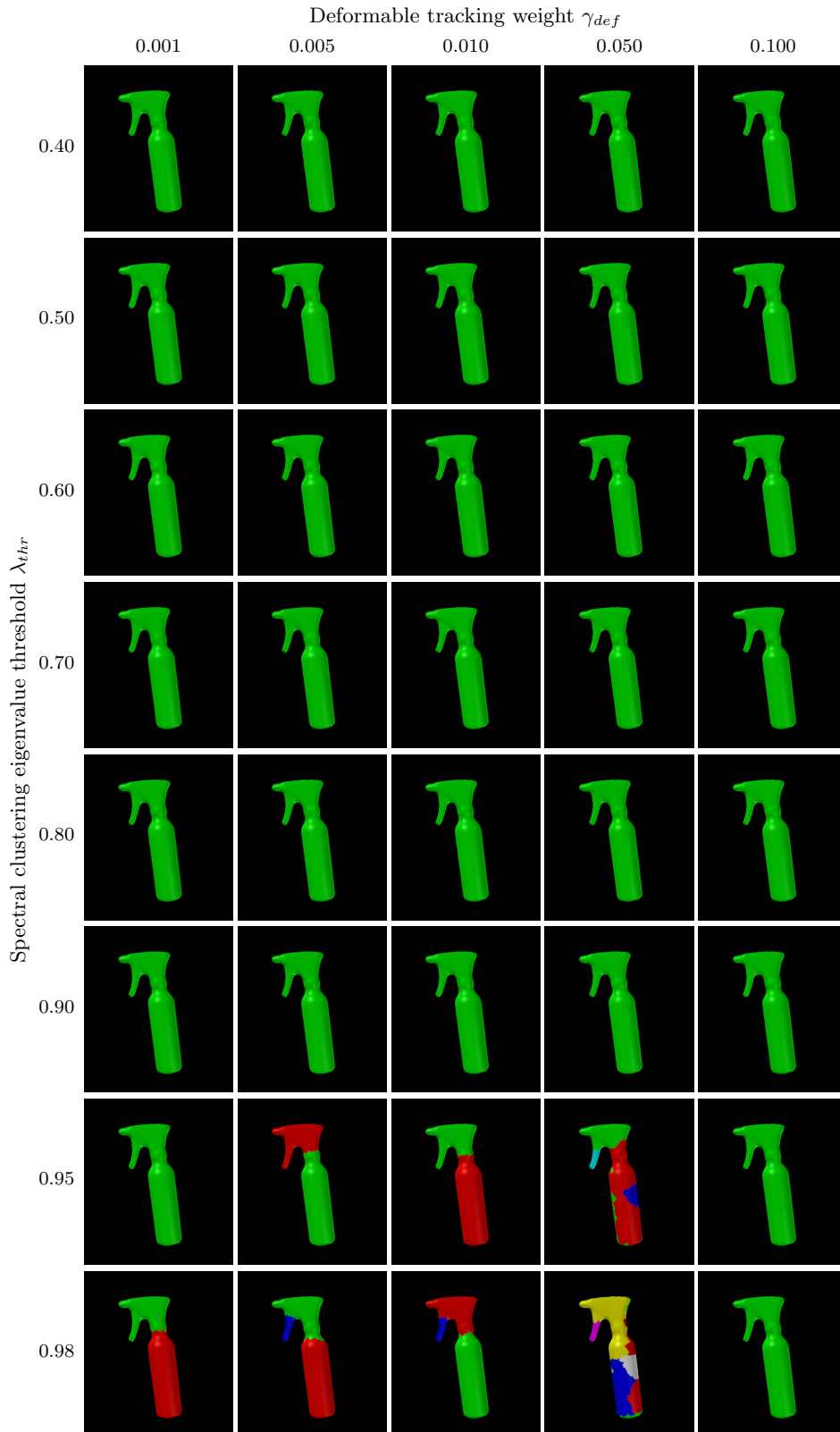


Figure 6.8: Results for all configurations $(\gamma_{def}, \lambda_{thr})$ spanning the parameter space. The images show for the object “donkey” the motion segments.

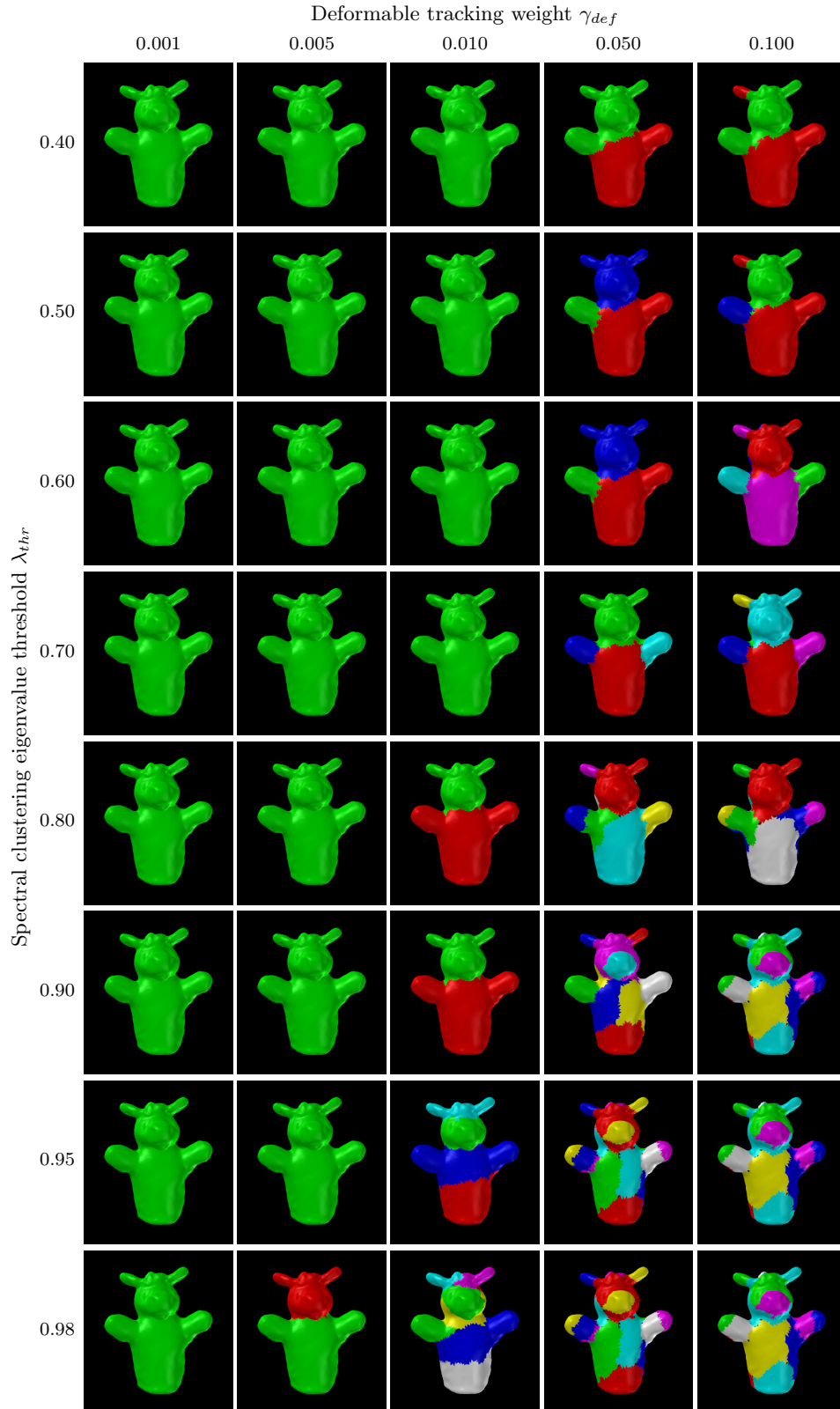


Figure 6.9: Results for all configurations $(\gamma_{def}, \lambda_{thr})$ spanning the parameter space. The images show for the object “lamp” the motion segments.

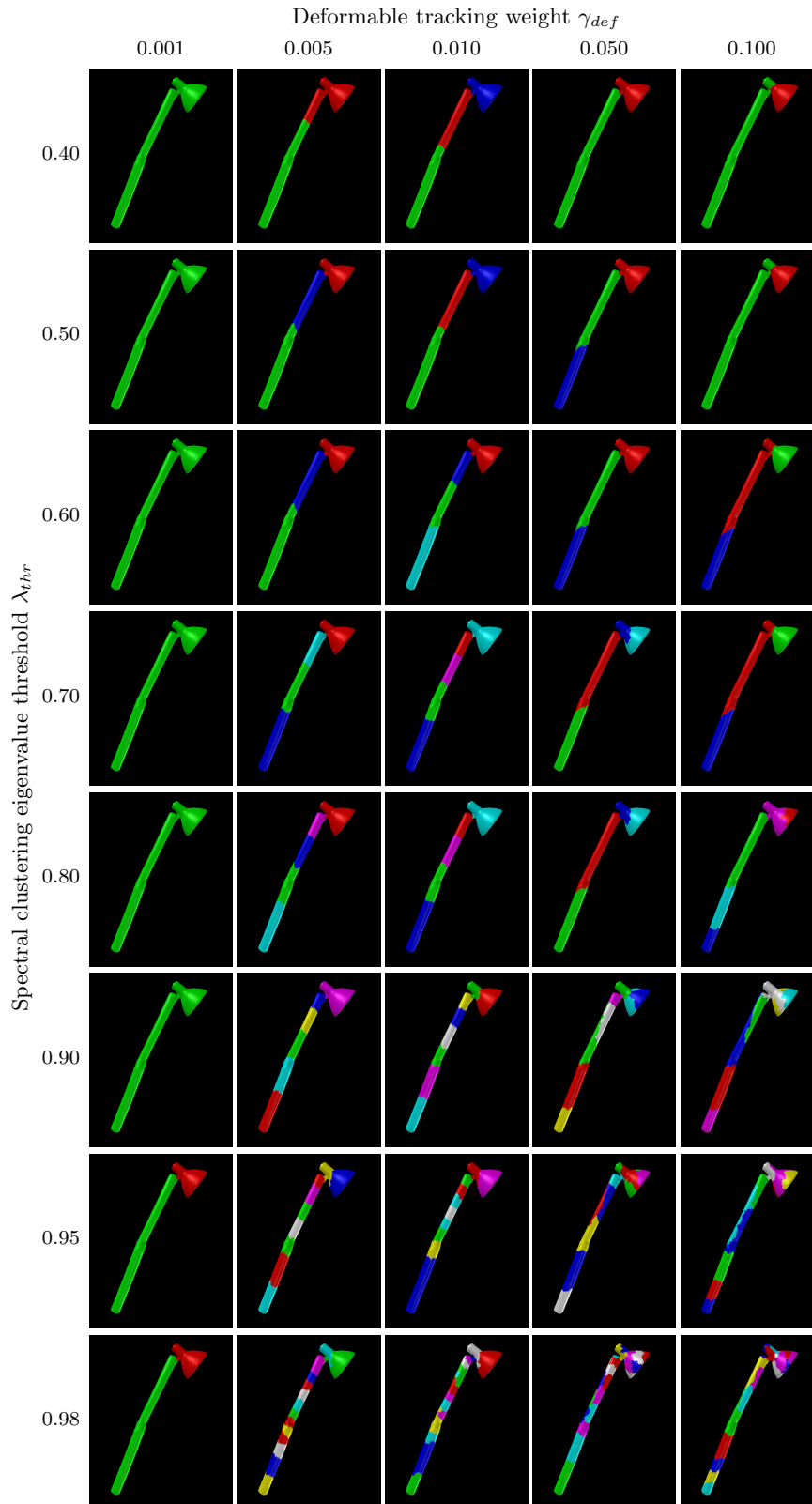


Figure 6.10: Results for all configurations $(\gamma_{def}, \lambda_{thr})$ spanning the parameter space. The images show for the object “pipe 1/2” the motion segments.

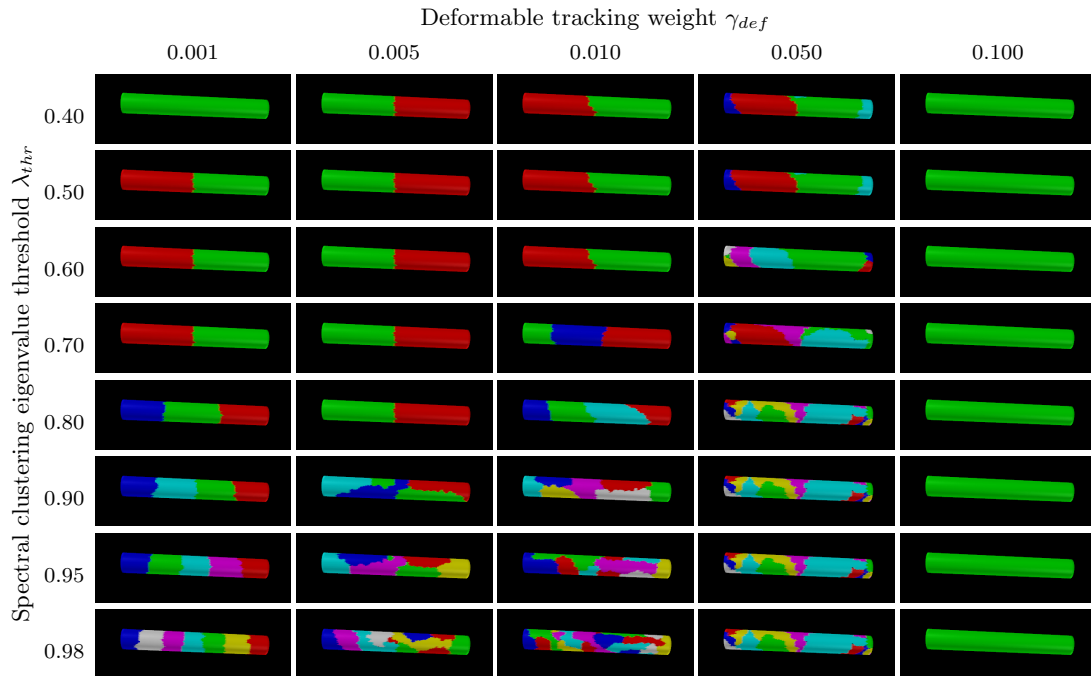


Figure 6.11: Results for all configurations $(\gamma_{def}, \lambda_{thr})$ spanning the parameter space. The images show for the object “pipe 3/4” the motion segments.

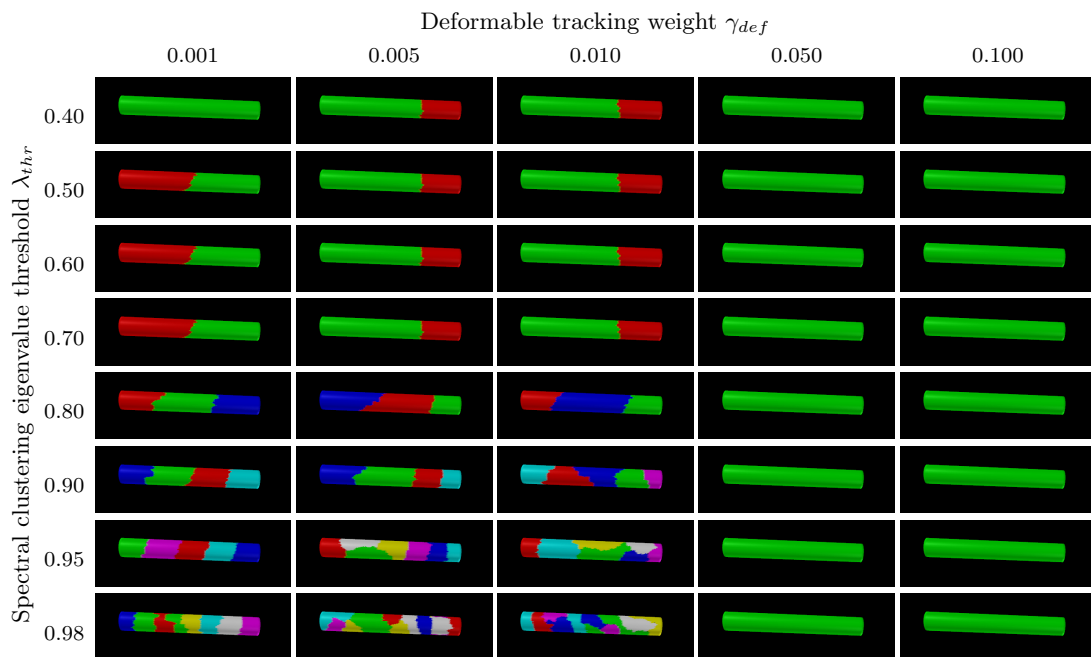


Figure 6.12: Results for all configurations $(\gamma_{def}, \lambda_{thr})$ that arise from the proposed parameters. The images show for the object “spray” the inferred 3d skeleton, where the joints with DoF are depicted with red color.

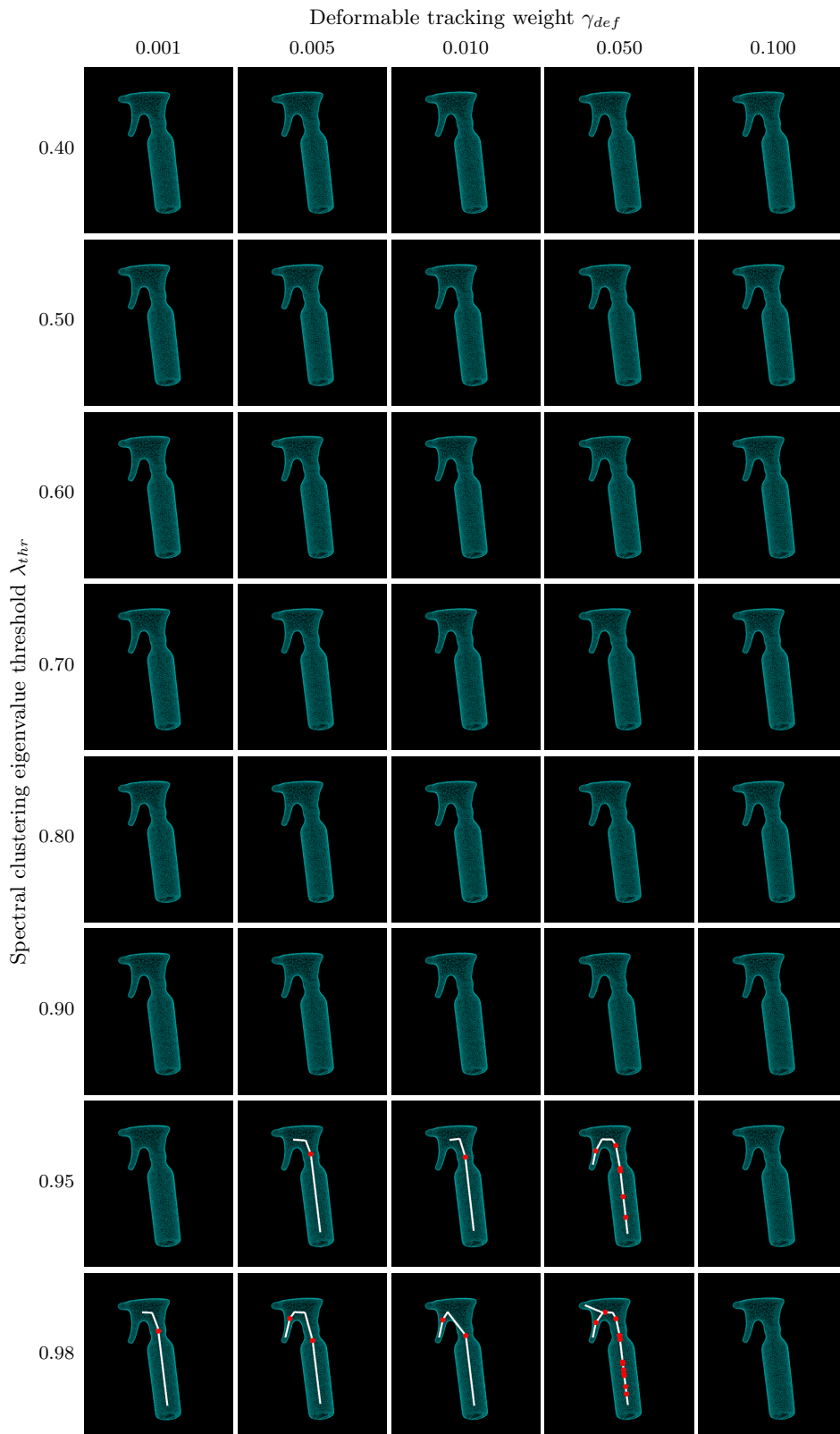


Figure 6.13: Results for all configurations $(\gamma_{def}, \lambda_{thr})$ that arise from the proposed parameters. The images show for the object “donkey” the inferred 3d skeleton, where the joints with DoF are depicted with red color.

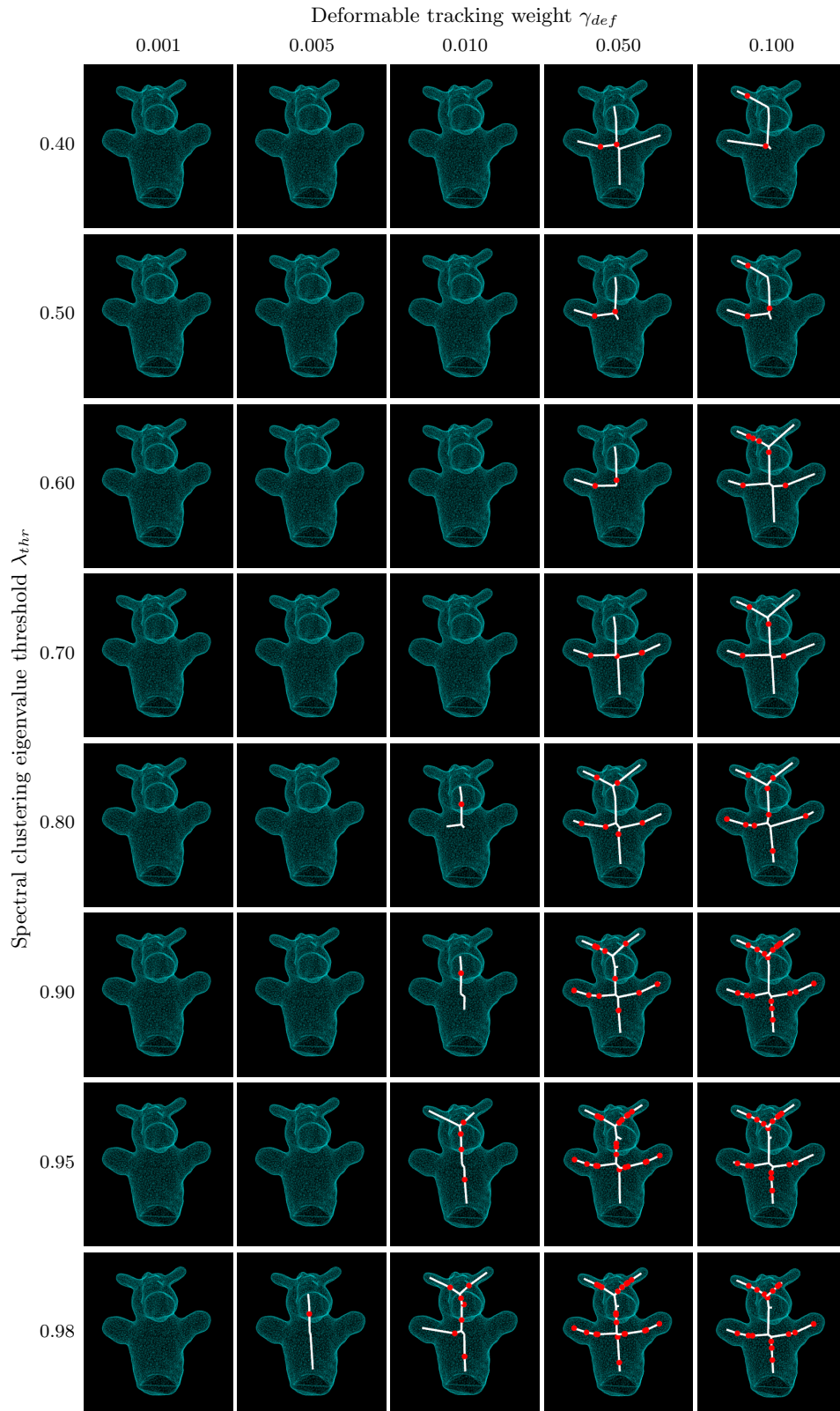


Figure 6.14: Results for all configurations $(\gamma_{def}, \lambda_{thr})$ that arise from the proposed parameters. The images show for the object “lamp” the inferred 3d skeleton, where the joints with DoF are depicted with red color.

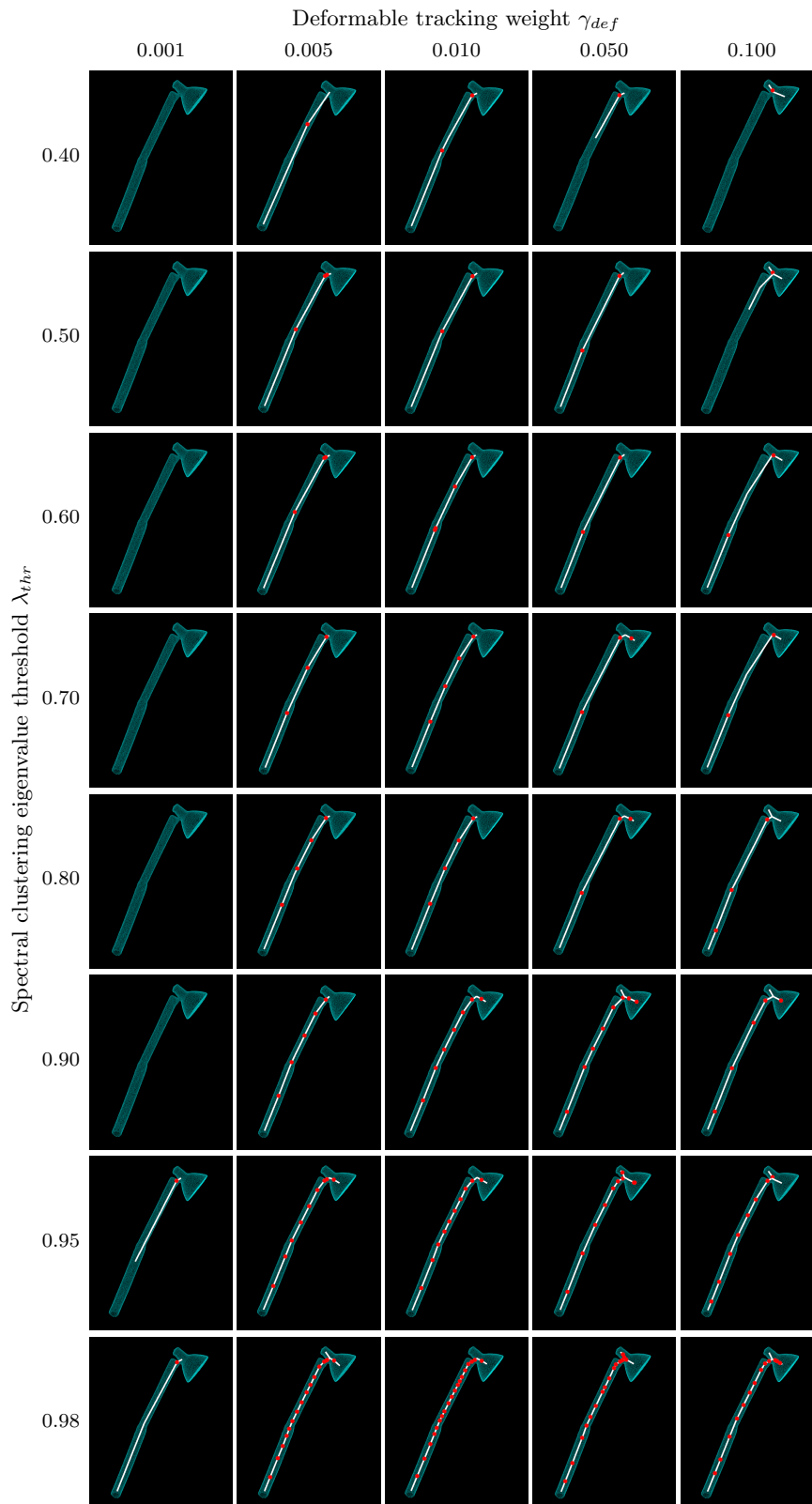


Figure 6.15: Results for all configurations $(\gamma_{def}, \lambda_{thr})$ that arise from the proposed parameters. The images show for the object “pipe 1/2” the inferred 3d skeleton, where the joints with DoF are depicted with red color.

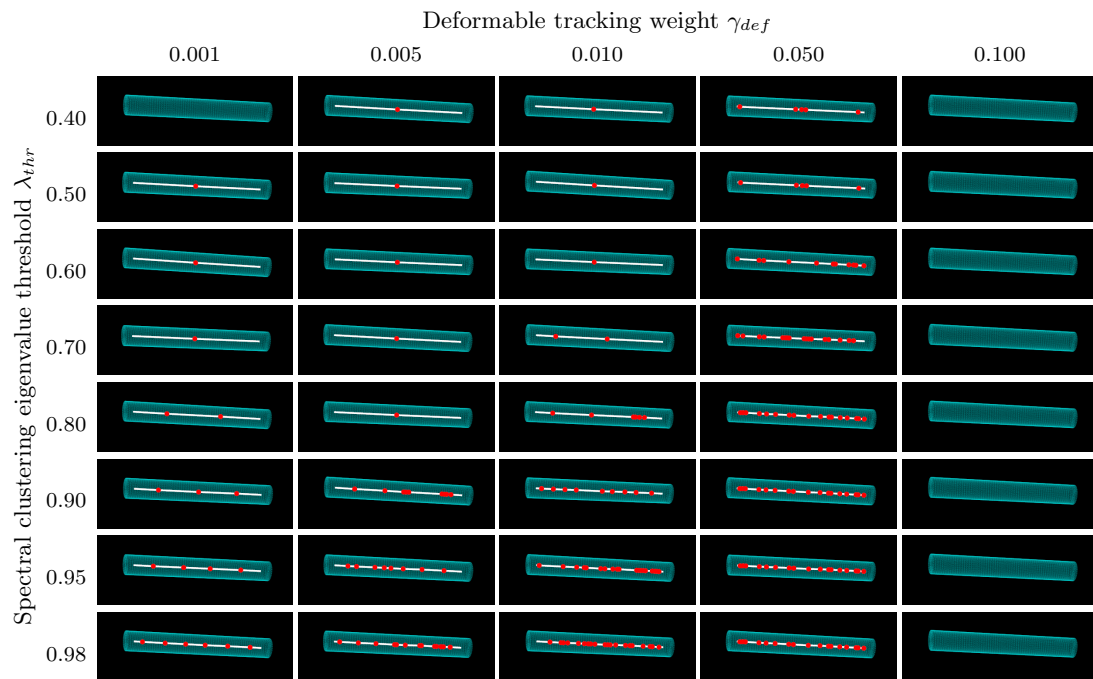


Figure 6.16: Results for all configurations $(\gamma_{def}, \lambda_{thr})$ that arise from the proposed parameters. The images show for the object “pipe 3/4” the inferred 3d skeleton, where the joints with DoF are depicted with red color.



Conclusions

Contents

7.1 Overview	103
7.2 Contributions and Discussion	103
7.2.1 Evaluation for Hand Pose Estimation	104
7.2.2 Capturing Hands in Action	104
7.2.3 3D Object Reconstruction from Hand-Object Interactions	105
7.2.4 Reconstructing Object Skeletons from RGB-D Videos	106
7.3 Future Work	106
7.4 Summary	109

7.1 Overview

Hand motion capture has been a popular research field for decades, while recently it gained further attention with the advent of commodity RGB-D sensors. Due to the challenges involved though, even most recent approaches focus on the case of a single isolated hand.

We focus instead on the more challenging problem of hands in action. In this direction we present a framework that captures the motion of hands interacting with other hands or with a known object, either rigid or articulated.

In case of an unknown object, existing approaches for 3d shape reconstruction need distinctive features and fail for textureless and highly symmetric objects. We show that reconstruction of such objects is feasible by incorporating 3d hand motion information in an in-hand scanning pipeline.

For articulated objects though also the skeleton of the object has to be reconstructed. We present an approach that builds rigged models of articulated objects from RGB-D videos where they deform realistically.

7.2 Contributions and Discussion

The work presented in this thesis can be clustered in four main categories, as presented below.

7.2.1 Evaluation for Hand Pose Estimation

Initially we propose a new benchmark dataset and protocol for hand pose estimation using frame pairs of escalating difficulty. The proposed protocol allows to evaluate features or components for a hand tracker without running a full tracking pipeline. It is thus robust to error accumulation of traditional tracking protocols that might be dataset-specific. Consequently it gives better insights about the actual performance of features and methods.

As an example, we discuss a generalized Chamfer distance and evaluate four special cases. The experiments reveal that directional information is important and a signed circular distance performs better than an unsigned distance in the case of silhouettes. Interestingly, a distance using a circular threshold outperforms a smooth directional Chamfer distance both in terms of accuracy and runtime. Comparison of the results with synthetic and real image data suggests that synthetic data can be a good indicator for the performance of a single method, but might be misleading when comparing different methods. Further evaluation could include frame pairs of other sequences with more background clutter and segmentation noise.

7.2.2 Capturing Hands in Action

We then present a full tracking framework for 3d motion capture of hands in action. Although most existing works focus on gestures and single hands, we focus on the more difficult case of intense hand-hand and hand-object interactions. The presented approach is generalized to work both with monocular RGB-D videos as well as with multiple synchronized RGB videos.

To address the difficulties, we have proposed an approach that combines in a single objective function a generative model with discriminatively trained salient points, collision detection and physics simulation for contact points. Experiments suggest that each term of the objective function has a valuable contribution in tracking accuracy. Although the collision and physics term reduce the pose estimation only slightly, they increase the realism of the captured motion, especially under occlusions and missing visual data. We extensively evaluate our approach, both qualitatively and quantitatively, on 8 multicamera RGB sequences and on 21 monocular RGB-D sequences. Further experiments for both camera systems reveal that our model outperforms an approach based on particle swarm optimization [Oikonomidis et al., 2011a] in terms of tracking accuracy. For the first time, we present successful tracking results of hands interacting with highly articulated objects.

At the moment the physics component adds constraints during hand-object interaction only on the hand. However additional constraints on the manipulated object could further improve tracking stability. Moreover although the actual physics simulation is fast, the method doesn't scale up nicely for complex objects. The only speed bottleneck for this is the computation of the closest 3d points between the hand and the manipulated object. Replacing the current method with an alternative could vastly improve the runtime and its suitability for real-time systems.

Another point for runtime improvement is the resolution of the hand mesh and the input point cloud. At the moment we use a high resolution hand mesh, as well as all the points of the point cloud. However [Taylor et al., 2016] report that 192 points suffice without compromises, a fact that might suggest the same for the mesh resolution. A lower resolution mesh would speed up runtime, while sub-sampling could be adaptive according to the distance from the camera and the action performed, e.g. hand in isolation or in interaction. However experimentation for the trade-off between speed and accuracy is needed, along with special care for the fingertip vertices.

The fact that our pipeline performs nicely for our datasets but performance drops for the Dexter dataset, is to a large extent due to the use of our personal 3d hand mesh on sequences of a hand with a drastically different shape. This suggests the high importance of a personalized model in accordance to [Tan et al., 2016] and points towards adopting an approach similar to [Taylor et al., 2014] to personalize a generic template hand mesh or [Khamis et al., 2015] to personalize a lower dimensional hand shape model learned from a big population.

Finally, in this thesis we employ a structured light camera since one of our goals was 3d reconstructions of objects and its resolution was attractive. However such cameras are not optimized for thin objects like fingers, they exhibit blurring and misalignment of the input RGB and depth maps during fast motion, while missing data of non-occluded hand parts is a frequent phenomenon. In that respect our experience suggests that a Time-of-Flight (ToF) camera is better suited for hand tracking and fingertip detection.

7.2.3 3D Object Reconstruction from Hand-Object Interactions

Existing approaches for 3d reconstruction are based on visual correspondences from stable and distinctive geometric and texture features, while in the absence of them they fail. The same applies for in-hand scanning approaches, that traditionally discard the motion information of the hand that rotates the object during scanning.

Instead, we propose an in-hand scanning approach that successfully incorporates the 3d hand motion information for 3d object reconstruction. To this end, contact correspondences between the hand and the manipulated object are computed and combined with the visual correspondences. Our experiments show both quantitatively and qualitatively that our approach successfully reconstructs the 3d shape of four highly symmetric and textureless objects.

While our approach does not depend on a specific hand tracker, it only works if the hand tracker does not fail. At the moment only end-effectors are considered, therefore cases where there is only contact with the palm are not handled, but the approach can be extended to more general contact points. Moreover the case of fingers slipping over the manipulated object is not handled currently, but this could be addressed by using a hand tracker that estimates forces [Pham et al., 2015].

A further limitation of the current approach is that the object is reconstructed offline, separately from the hand tracker. A natural future extension would be to make our approach online using reconstruction techniques similar to [Newcombe et al., 2011], so that a partial model could be used online for tracking [Panteleris et al., 2015].

7.2.4 Reconstructing Object Skeletons from RGB-D Videos

Existing 3d object reconstruction approaches focus only on reconstructing the unknown 3d shape of an object. However for articulated objects also the kinematic model in the form of a 3d skeleton has to be acquired.

In this direction we present an approach to reconstruct the unknown 3d skeleton of articulated objects from RGB-D videos that contain pronounced deformations of them. For this we operate fully in 3d capitalizing on deformable tracking, spectral clustering and skeletonization based on mean curvature flow. The thorough evaluation of the parameters provides a valuable intuition about the important factors and opens up possibilities for further generalization in future work. For instance, a regularizer that is adaptive to the areas of the triangles can be used for deformable tracking to compensate seamlessly for the varying triangle sizes across different objects. The output of the system is a fully rigged model consisting of a watertight mesh, an embedded skeleton and skinning weights. This output can be used out of the box for articulated tracking or animation. Our experiments show that the generated rigged models work nicely and that our approach has prospects for future practical applications.

At the moment all inferred 3d skeleton joints have 3 Degrees of Freedom (DoF) to allow for every possible 3d rotation. An obvious extension would be to infer the actual number of DoF for each joint, as well as the direction of the corresponding rotation axes. Furthermore, an As-Rigid-As-Possible (ARAP) [Sorkine and Alexa, 2007] approach could be examined for deformable tracking. Finally, the current setup works well without hand motion information even under some occlusions, but for more challenging sequences with intense occlusions of manipulated parts hand motion could be incorporated too.

7.3 Future Work

This thesis studies aspects of 3d perception like capturing the 3d motion of hands in action, reconstructing the 3d shape of rigid objects and acquiring the kinematic skeleton of articulated objects. Although it contributes towards better understanding of the above problems, it also suggests ways for improvements in future work, as described in the following.

Holistic Approach

At the moment the aforementioned problem aspects are studied disjointly. A natural long term goal would be to tackle them jointly with a holistic approach. With such an approach a user would be able to interact in the wild with objects in the surrounding space while 3d motion would be captured transparently in the background. Interactions with known objects would not be a problem, while interaction with unknown objects would start with a very rough initial estimation of the object whose shape and kinematic models would be refined online. Interesting directions for such a holistic online approach are not only rigid or articulated objects, but also highly non-rigid de-

formable objects, or objects whose structure can change drastically, e.g. a pen whose cup is removed, a deformable paper that is torn or folded in a rigid or articulated origami, a pizza that is cut in pieces, etc.

Personalized Hand Model

As our experiments suggest there is a pressing need for a personalized shape model for tracking. This could be realized with an approach similar to [Taylor et al., 2014] to personalize a generic hand template by non-rigid fitting to a sequence of a user’s hand or [Khamis et al., 2015] to personalize a data-driven shape model which usually lies on a low dimensional manifold, so that the added complexity in the scene would not be prohibitive. These approaches are though still open questions for research. An ideal holistic system though would be able to also reconstruct the hand model without any prior knowledge, starting from a rough model and constant online refinement, so that it would even capture skin deformations due to aging or drastic shape changes, e.g. for amputations. Such an approach would also increase user’s sense of “ownership” especially in Augmented or Virtual Reality (AR/VR) applications.

Multi-layer approach with (Re)Initialization

The current method casts the tracking problem as an optimization problem. In this direction local optimization is used for pose estimation and the solution for each frame is initialized with the pose of the previous frame. However the objective function being optimized has a lot of local minima that multiply during hand object interaction due to increased ambiguities. Although discriminatively trained salient points guide the optimization away from local minima and increase the basin of convergence, additional approaches can be studied like hand-parts classification [Shotton et al., 2011, Keskin et al., 2012, Sridhar et al., 2016]. However local minima can still trap the optimization with destructive results for tracking without a reinitialization method. Moreover, the pose for the first frame is initialized manually. The above indicate a strong need for a (re)initializer based on global optimization. In this direction a multi-layer approach could be used with global optimization for a rough initial pose estimate and local optimization for pose refinement, in accordance to [Taylor et al., 2016, Gall, 2009].

Inference of Contact Points

Even such a system though would give ambiguous solutions for symmetric objects. The contact points though that were employed to drive the 3d reconstruction of symmetric objects in this thesis could be used to stabilize tracking results when a hand interacts with such objects, as also suggested by [Pham et al., 2015, Sridhar et al., 2016]. Finding contact points is thus of high importance but detecting or inferring them is very challenging. When the object is known but a manipulating finger is occluded for a very long time, the pose of the latter can be highly erroneous and trapped in a local minima. When the object is unknown, detection of contact points between the tracked hand and the object’s partial point cloud can be computed with simple

distance proximity only in case of very subtle or no occlusions. In case though of occluded manipulators the assumption of close proximity is not valid any more and the problem is highly ill-posed. This fact points to an interesting challenging problem for contact point estimation through data driven techniques from massive data.

Inference of Forces

In a similar fashion an emerging hot research topic is the inference of forces either from sequences [Pham et al., 2015] or from single images [Rogez et al., 2015b]. The largely uncharted territory is worth exploring, while fruitful results would be beneficial to model second order dynamics during manipulation and even reduce the complexity of the scene inspired from [Kyriazis and Argyros, 2013, 2014].

Priors

Data driven techniques can also be used to incorporate prior knowledge in the tracking pipeline in the form of a pose or motion prior. Such priors gain higher importance for monocular systems and in the presence of intense occlusions, when there is not enough visual data to constrain all moving parts. Although simple priors like joint limits are already commonly used, there is no systematic approach for an accurate data-driven prior for anthropometrically valid hand poses, as with full bodies [Akhter and Black, 2015]. In this direction finding pose-dependent joint angle limits is of high importance, but modeling should also take into account that interaction with objects changes drastically the subspace of physically plausible hand poses.

Inference of Correspondences

Local optimization is fast in nature, but extracting features for it is a bottleneck as our experiments show, in accordance to [Gall, 2009]. In that respect recent breakthroughs in machine learning pave the way to effectively learn a model to infer dense correspondences [Taylor et al., 2012]. However such correspondence inference is still an open problem that needs further exploration.

Event Cameras

Another way for cues for local optimization might be the use of additional modalities like the use of event cameras. Instead of capturing full images, such cameras output only a stream of asynchronous spikes of discrete changes in log intensity of pixels. Such cameras operate with unusually high dynamic range and temporal resolution, are robust to blur, occlusion boundaries and noise while they massively reduce the bandwidth requirements, characteristics highly suitable for hand tracking. Although this modality alone is fundamentally different from the traditional ones and not well explored for tracking, apart for some works on SLAM like [Kim et al., 2014, 2016], there is already work that combines traditional RGB-D sensors with event cameras

[Weikersdorfer et al., 2014]. However this field is totally unexplored for articulated tracking up to date.

Hand and Body Mesh Fusion

Although there is a lot of research on full human body as well as human hand tracking, the problems are handled separately. Despite the similarities, the different characteristics of the two problems suggest that this disjoint approach is rather reasonable. However, there is a lack of methods to jointly capture the motion of hands and full bodies, to which the former belong.

Hand-Object Segmentation

One of the fundamental problems during hand-object interaction is the segmentation of the RGB-D input image into a hand and object region, without additional hardware like thermal cameras. In this thesis we resort to a simple approach of skin color segmentation similarly to [Romero et al., 2010, Oikonomidis et al., 2011b], however this approach performs poorly with shadows from occlusions and uncontrolled lighting conditions, while it allows the use of only non-skin colored objects. As an alternative a region growing alternative could be tried like [Panteleris et al., 2015, Taylor et al., 2016] from some initial seeds based on fingertip detections or on the pose of the previous tracked frame. Better alternatives would be a more generally applicable solution with a strong discriminative model on a single image like general hand detection [Bambach et al., 2015] or hand detection during hand-object interaction [Kang et al., 2016].

Runtime Optimization

Finally, currently the methods of this thesis are implemented with unoptimized code running on a single CPU. Optimizations for real-time performance are highly desirable for systems that show reasonable accuracy. However one should have in mind that hand tracking and object reconstruction approaches might be incorporated in bigger applications, so overwhelmingly occupying a GPU should be avoided.

7.4 Summary

This thesis studies several aspects of the challenging problems of capturing the 3d motion of hands in action, reconstructing the 3d shape of rigid objects and acquiring the 3d kinematic skeleton of articulated objects.

It presents for the first time in the literature successful tracking of hands interacting with known articulated objects, successful 3d shape reconstruction of unknown highly symmetric and textureless objects with an in-hand scanning pipeline that incorporates hand motion information, as well as an approach to acquire fully rigged models of articulated objects with unknown skeleton, all using a single RGB-D camera.

Although the studied problems are still open questions for the scientific community, the contributions of this thesis significantly contribute towards better understanding

of the aspects involved and pave the way for further improvements and generalizations in future work.

Bibliography

- Alok Aggarwal, Maria M. Klawe, Shlomo Moran, Peter Shor, and Robert Wilber. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2(1-4):195–208, 1987. (Cited on page 38.)
- Ijaz Akhter and Michael J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1446–1455, 2015. (Cited on page 108.)
- Irene Albrecht, Jörg Haber, and Hans-Peter Seidel. Construction and animation of anatomically based human hand models. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA)*, pages 98–109, 2003. (Cited on page 40.)
- Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Transactions on Graphics (TOG)*, 24(3):408–416, 2005. (Cited on page 12.)
- Vassilis Athitsos and Stan Sclaroff. Estimating 3d hand pose from a cluttered image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 432–439, 2003. (Cited on pages 9, 15 and 18.)
- Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997. (Cited on page 7.)
- Andreas Baak, Thomas Helten, Meinard Müller, Gerard Pons-Moll, Bodo Rosenhahn, and Hans-Peter Seidel. Analyzing and evaluating markerless motion tracking using inertial sensors. In *Workshop on Human Motion*, pages 137–150, 2010. (Cited on page 18.)
- Luca Ballan and Guido Maria Cortelazzo. Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, 2008. (Cited on page 42.)
- Luca Ballan, Aparna Taneja, Juergen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *European Conference on Computer Vision (ECCV)*, pages 640–653, 2012. (Cited on pages viivii, 10, 11, 12, 13, 14, 15, 18, 19, 22, 24, 28, 65, 70 and 71.)
- Sven Bambach, Stefan Lee, David Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *International Conference on Computer Vision (ICCV)*, pages 1949–1957, 2015. (Cited on pages 15 and 109.)
- Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Transactions on Graphics (TOG)*, 26(3), 2007. (Cited on pages 29, 82 and 88.)

- Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(4):509–522, 2002. (Cited on page 37.)
- Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256, 1992. (Cited on pages 64 and 69.)
- G erard Blais and Martin D. Levine. Registering multiview range data to create 3d computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(8):820–824, 1995. (Cited on page 64.)
- Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 14(1):213–230, 2008. (Cited on pages 84 and 85.)
- Matthieu Bray, Esther Koller-Meier, and Luc Van Gool. Smart particle filtering for high-dimensional tracking. *Computer Vision and Image Understanding (CVIU)*, 106(1):116–129, 2007. (Cited on pages 9 and 12.)
- Christoph Bregler, Jitendra Malik, and Katherine Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision (IJCV)*, 56(3):179–194, 2004. (Cited on pages 19 and 30.)
- Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *European Conference on Computer Vision (ECCV)*, pages 282–295, 2010. (Cited on page 86.)
- Thomas Brox, Bodo Rosenhahn, Juergen Gall, and Daniel Cremers. Combined region- and motion-based 3d tracking of rigid and articulated objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(3):402–415, 2010. (Cited on page 42.)
- John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(6):679–698, 1986. (Cited on page 33.)
- Hyung Jin Chang and Yiannis Demiris. Unsupervised learning of complex articulated kinematic structures combining motion and skeleton information. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3138–3146, 2015. (Cited on pages 82 and 83.)
- Yang Chen and G erard Medioni. Object modeling by registration of multiple range images. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2724–2729, 1991. (Cited on pages 32, 68 and 69.)
- Erwin Coumans. Bullet real-time physics simulation, 2013. URL <http://bulletphysics.org>. (Cited on page 38.)

- Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *ACM Transactions on Graphics (SIGGRAPH)*, pages 303–312, 1996. (Cited on pages 64 and 70.)
- Edilson De Aguiar, Christian Theobalt, Sebastian Thrun, and Hans-Peter Seidel. Automatic conversion of mesh animations into skeleton-based animations. *Computer Graphics Forum (CGF)*, 27(2):389–397, 2008. (Cited on page 82.)
- Teófilo Emídio de Campos and David W. Murray. Regression-based hand pose estimation from multiple cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–789, 2006. (Cited on page 9.)
- Martin de La Gorce, David J. Fleet, and Nikos Paragios. Model-based 3d hand pose estimation from monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(9):1793–1805, 2011. (Cited on pages 12 and 18.)
- Quentin Delamarre and Olivier D. Faugeras. 3d articulated models and multiview tracking with physical forces. *Computer Vision and Image Understanding (CVIU)*, 81(3):328–357, 2001. (Cited on pages 12 and 18.)
- Laura Dipietro, Angelo M Sabatini, and Paolo Dario. A survey of glove-based systems and their applications. *Transactions on Systems, Man, and Cybernetics*, 38(4):461–482, 2008. (Cited on page 8.)
- Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1538–1546, 2015. (Cited on page 11.)
- Ali Erol, George Bebis, Mircea Nicolescu, Richard D. Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding (CVIU)*, 108(1-2):52–73, 2007a. (Cited on pages 2, 7, 14, 17 and 18.)
- Ali Erol, George Bebis, Mircea Nicolescu, Richard D. Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding (CVIU)*, 108(1-2):52–73, 2007b. (Cited on pages 7 and 42.)
- Mark Everingham, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010. (Cited on page 45.)
- Joao Fayad, Chris Russell, and Lourdes Agapito. Automated articulated structure and 3d shape recovery from point correspondences. In *International Conference on Computer Vision (ICCV)*, pages 431–438, 2011. (Cited on page 83.)
- Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science, 2004. (Cited on pages 20 and 33.)

- Silvio Filipe and Luis A Alexandre. A comparative evaluation of 3d keypoint detectors. In *Conference on Telecommunications*, pages 145–148, 2013. (Cited on page 69.)
- Juergen Gall. *Filtering and optimization strategies for markerless human motion capture with skeleton-based shape models*. PhD thesis, PhD Thesis, 2009. (Cited on pages 107 and 108.)
- Juergen Gall, Carsten Stoll, Edilson De Aguiar, Christian Theobalt, Bodo Rosenhahn, and Hans-Peter Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1746–1753, 2009. (Cited on page 84.)
- Juergen Gall, Andrea Fossati, and Luc Van Gool. Functional categorization of objects using real-time markerless motion capture. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1969–1976, 2011a. (Cited on page 33.)
- Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(11):2188–2202, 2011b. (Cited on pages 36 and 74.)
- Bernd Gärtner and Sven Schönherr. An efficient, exact, and generic quadratic programming solver for geometric optimization. In *Symposium on Computational Geometry (SCG)*, pages 110–118, 2000. (Cited on page 38.)
- Dariu M. Gavrilă. Multi-feature hierarchical template matching using distance transforms. In *International Conference on Pattern Recognition (ICPR)*, pages 439–444, 1998. (Cited on pages 18 and 21.)
- Dariu M. Gavrilă. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding (CVIU)*, 73(1):82 – 98, 1999. (Cited on page 8.)
- Henning Hamer, Konrad Schindler, Esther Koller-Meier, and Luc Van Gool. Tracking a hand manipulating an object. In *International Conference on Computer Vision (ICCV)*, pages 1475–1482, 2009. (Cited on pages 9, 12, 13, 19 and 65.)
- Henning Hamer, Juergen Gall, Thibaut Weise, and Luc Van Gool. An object-dependent hand pose prior from sparse training data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 671–678, 2010. (Cited on pages 9, 13, 19 and 65.)
- Dong Han, Bodo Rosenhahn, Joachim Weickert, and Hans-Peter Seidel. Combined registration methods for pose estimation. In *International Symposium on Visual Computing (ISVC)*, pages 913–924, 2008. (Cited on page 19.)
- Tony Heap and David Hogg. Towards 3d hand tracking using a deformable model. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 140–145, 1996. (Cited on pages 7, 9, 12 and 18.)

- Thomas Helten, Andreas Baak, Meinard Müller, and Christian Theobalt. *Full-Body Human Motion Capture from Monocular Depth Images*, pages 188–206. Springer, 2013. (Cited on page 8.)
- Stefan Holzer, Radu Bogdan Rusu, Michael Dixon, Suat Gedikli, and Nassir Navab. Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2684–2689, 2012. (Cited on pages 32 and 84.)
- iSense. isense. <http://cubify.com/products/isense>, 2016. Accessed: 17/08/2016. (Cited on page 64.)
- Hiroshi Ishii and Brygg Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *ACM Special Interest Group on Computer-Human Interaction (SIGCHI)*, pages 234–241, 1997. (Cited on page 1.)
- Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *ACM Symposium on User Interface Software and Technology (UIST)*, 2011. (Cited on page 2.)
- Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision (IJCV)*, 46(1):81–96, 2002. (Cited on pages 32, 66 and 84.)
- Byeongkeun Kang, Kar-Han Tan, Hung-Shuo Tai, Daniel Tretter, and Truong Q. Nguyen. Hand segmentation for hand-object interaction from depth map. *arXiv:1603.02345*, 2016. (Cited on page 109.)
- Dov Katz, Mostafa Kazemi, Andrew J. Bagnell, and Anthony Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5003–5010, 2013. (Cited on page 83.)
- Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Eurographics Symposium on Geometry Processing (SGP)*, 2006. (Cited on page 70.)
- Cem Keskin, Furkan Kırış, YunusEmre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *European Conference on Computer Vision (ECCV)*, pages 852–863, 2012. (Cited on pages 9 and 107.)
- Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth

- images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2540–2548, 2015. (Cited on pages 9, 11, 12, 105 and 107.)
- David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *ACM Symposium on User Interface Software and Technology (UIST)*, pages 167–176, 2012. (Cited on pages 8 and 9.)
- Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew J. Davison. Simultaneous mosaicing and tracking with an event camera. In *British Machine Vision Conference (BMVC)*, 2014. (Cited on page 108.)
- Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew J. Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *European Conference on Computer Vision (ECCV)*, 2016. (Cited on page 108.)
- KINFU. Kinfu. http://pointclouds.org/documentation/tutorials/using_kinfu_large_scale.php, 2016. Accessed: 17/08/2016. (Cited on page 73.)
- Oscar Koller, Hermann Ney, and Richard Bowden. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3793–3802, 2016. (Cited on page 15.)
- Dirk Kraft, Nicolas Pugeault, Emre Başeski, Mila popović, Danica Kragić, Sinan Kalkan, Florentin Wörgötter, and Norbert Krüger. Birth of the object: detection of objectness and extraction of object shape through object-action complexes. *International Journal of Humanoid Robotics*, 5(2):247–265, 2008. (Cited on page 65.)
- Michael Krainin, Peter Henry, Xiaofeng Ren, and Dieter Fox. Manipulator and object tracking for in-hand 3d object modeling. *International Journal of Robotics Research (IJRR)*, 30(11):1311–1327, 2011. (Cited on page 65.)
- KSCAN3D. Kscan3d. <http://www.kscan3d.com>, 2016. Accessed: 17/08/2016. (Cited on page 64.)
- Nikolaos Kyriazis and Antonis Argyros. Physically plausible 3d scene tracking: The single actor hypothesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9–16, 2013. (Cited on pages 13, 14, 65 and 108.)
- Nikolaos Kyriazis and Antonis Argyros. Scalable 3d tracking of multiple interacting objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3430–3437, 2014. (Cited on pages 13 and 108.)
- LEAP MOTION. Leap motion. <https://www.leapmotion.com>, 2016. Accessed: 17/08/2016. (Cited on page 2.)

- John P. Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *ACM Transactions on Graphics (SIGGRAPH)*, pages 165–172, 2000. (Cited on pages 19, 29 and 67.)
- Hao Li, Etienne Vouga, Anton Gudym, Linjie Luo, Jonathan T. Barron, and Gleb Gusev. 3d self-portraits. *ACM Transactions on Graphics (SIGGRAPH Asia)*, pages 187:1–187:9, 2013. (Cited on page 64.)
- Ming-Yu Liu, Oncel Tuzel, Ashok Veeraraghavan, and Rama Chellappa. Fast directional chamfer matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1696–1703, 2010. (Cited on pages 18, 19, 21 and 22.)
- Yebin Liu, Juergen Gall, Carsten Stoll, Qionghai Dai, Hans-Peter Seidel, and Christian Theobalt. Markerless motion capture of multiple characters using multiview image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(11):2720–2735, 2013. (Cited on pages 82 and 84.)
- Charles Loop. *Smooth Subdivision Surfaces Based on Triangles*. Department of Mathematics, University of Utah, 1987. (Cited on pages 10 and 12.)
- William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM Transactions on Graphics (SIGGRAPH)*, 21(4):163–169, 1987. (Cited on page 70.)
- David G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, 1999. (Cited on page 69.)
- Shan Lu, D. Metaxas, D. Samaras, and J. Oliensis. Using multiple cues for hand tracking and model refinement. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 443–450, 2003. (Cited on pages 12 and 18.)
- Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981. (Cited on page 64.)
- John MacCormick and Michael Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *European Conference on Computer Vision (ECCV)*, pages 3–19, 2000. (Cited on page 8.)
- Roberto Martín-Martín, Sebastian Höfer, and Oliver Brock. An integrated approach to visual perception of articulated objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5091–5097, 2016. (Cited on page 83.)
- Stan Melax, Leonid Keselman, and Sterling Orsten. Dynamics based 3d skeletal hand tracking. In *Graphics Interface*, pages 63–70, 2013. (Cited on page 9.)

- MESHLAB. Meshlab. <http://meshlab.sourceforge.net>, 2016. Accessed: 2016-05-13. (Cited on page 84.)
- Damien Michel, Xenophon Zabulis, and Antonis A Argyros. Shape from interaction. *Machine Vision and Applications*, 25(4):1077–1087, 2014. (Cited on page 65.)
- Razvan-George Mihalyi, Kaustubh Pathak, Narunas Vaskevicius, and Andreas Birk. Uncertainty estimation of ar-marker poses for graph-slam optimization in 3d object model generation with rgbd data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1807–1813, 2013. (Cited on page 65.)
- Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding (CVIU)*, 81(3):231–268, 2001. (Cited on page 8.)
- Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding (CVIU)*, 104(2):90–126, 2006. (Cited on page 8.)
- Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4207–4215, 2016. (Cited on page 15.)
- Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994. (Cited on pages 19 and 30.)
- Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011. (Cited on pages 64, 70, 73, 81 and 105.)
- Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems NIPS*, volume 2, pages 849–856, 2002. (Cited on page 87.)
- NIMBLE VR. Nimble vr. <http://nimblevr.com>, 2016. Accessed: 17/08/2016. (Cited on page 2.)
- Kenichi Nirei, Hideo Saito, Masaaki Mochimaru, and Shinji Ozawa. Human hand tracking from binocular image sequences. In *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 297–302, 1996. (Cited on pages 12, 17 and 18.)
- Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *International Conference on Computer Vision (ICCV)*, pages 3316–3324, 2015. (Cited on pages 10, 11 and 15.)

- Markus Oberweger, Gernot Riegler, Paul Wohlhart, and Vincent Lepetit. Efficiently creating 3d training data for fine hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4957–4965, 2016. (Cited on pages 9 and 15.)
- OCULUS. Oculus. <https://www.oculus.com>, 2016. Accessed: 17/08/2016. (Cited on page 2.)
- Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *British Machine Vision Conference (BMVC)*, pages 101.1–101.11, 2011a. (Cited on pages 9, 10, 11, 12, 15, 43, 47, 48, 55, 57 and 104.)
- Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *International Conference on Computer Vision (ICCV)*, pages 2088–2095, 2011b. (Cited on pages 10, 12, 13, 14, 19, 47, 48, 65 and 109.)
- Iason Oikonomidis, Manolis I.A. Lourakis, and Antonis A. Argyros. Evolutionary quasi-random search for hand articulations tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3422–3429, 2014. (Cited on pages 9 and 10.)
- Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. Tracking the articulated motion of two strongly interacting hands. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1862–1869, 2012. (Cited on pages 10, 12, 13, 15, 18, 19, 47 and 48.)
- Pantelis Panteleris, Nikolaos Kyriazis, and Antonis A. Argyros. 3d tracking of human hands in interaction with unknown objects. In *British Machine Vision Conference (BMVC)*, pages 123.1–123.12, 2015. (Cited on pages 13, 14, 66, 105 and 109.)
- Sylvain Paris and Frédo Durand. A fast approximation of the bilateral filter using a signal processing approach. *International Journal of Computer Vision (IJCV)*, 81(1):24–52, 2009. (Cited on pages 32 and 84.)
- Tu-Hoa Pham, Abderrahmane Kheddar, Ammar Qammaz, and Antonis A. Argyros. Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2810–2819, 2015. (Cited on pages 13, 14, 105, 107 and 108.)
- Sudeep Pillai, Matthew R. Walter, and Seth Teller. Learning articulated motions from visual demonstration. In *Robotics: Science and Systems (RSS)*, 2014. (Cited on pages 82 and 83.)
- Gerard Pons-Moll and Bodo Rosenhahn. *Model-Based Pose Estimation*, chapter 9, pages 139–170. Springer, 2011. (Cited on pages 30, 33 and 85.)

- Gerard Pons-Moll, Laura Leal-Taixé, Tri Truong, and Bodo Rosenhahn. Efficient and robust shape matching for model based human motion capture. In *Annual Symposium of the German Association for Pattern Recognition (DAGM)*, 2011. (Cited on page 19.)
- Ronald Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding (CVIU)*, 108(1-2):4–18, 2007. (Cited on page 8.)
- Kari Pulli. Multiview registration for large data sets. In *3-D Digital Imaging and Modeling*, pages 160–168, 1999. (Cited on pages 64 and 65.)
- Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1106–1113, 2014. (Cited on pages 9, 10, 12, 15 and 43.)
- Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244 – 256, 1972. (Cited on pages 21 and 22.)
- James M. Rehg and Takeo Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *European Conference on Computer Vision (ECCV)*, pages 35–46, 1994. (Cited on pages 8, 18 and 28.)
- Grégory Rogez, Maryam Khademi, James S. Supančič III, Jose Maria Martinez Montiel, and Deva Ramanan. 3d hand pose detection in egocentric rgb-d images. In *ECCV Workshop on Consumer Depth Cameras for Computer Vision*, pages 356–371, 2014. (Cited on pages 9, 13 and 15.)
- Gregory Rogez, James S. Supančič III, and Deva Ramanan. First-person pose recognition using egocentric workspaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4325–4333, 2015a. (Cited on pages 9 and 13.)
- Grégory Rogez, James S. Supančič III, and Deva Ramanan. Understanding everyday hands in action from rgb-d images. In *International Conference on Computer Vision (ICCV)*, pages 3889–3897, 2015b. (Cited on pages 9, 13, 15 and 108.)
- Javier Romero, Hedvig Kjellström, and Danica Kragic. Monocular real-time 3d articulated hand pose estimation. In *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, pages 87–92, 2009. (Cited on page 9.)
- Javier Romero, Hedvig Kjellström, and Danica Kragic. Hands in action: real-time 3d reconstruction of hands in interaction with objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 458–463, 2010. (Cited on pages 9, 13, 19, 65 and 109.)
- Rómer Rosales, Vassilis Athitsos, Leonid Sigal, and Stan Sclaroff. 3d hand pose reconstruction using specialized mappings. In *International Conference on Computer Vision (ICCV)*, pages 378–387, 2001. (Cited on pages 9, 15 and 17.)

- Bodo Rosenhahn, Thomas Brox, and Joachim Weickert. Three-dimensional shape knowledge for joint image segmentation and pose tracking. *International Journal of Computer Vision (IJCV)*, 73(3):243–262, 2007. (Cited on pages 20 and 33.)
- David A. Ross, Daniel Tarlow, and Richard S. Zemel. Learning articulated structure and motion. *International Journal of Computer Vision (IJCV)*, 88(2):214–237, 2010. (Cited on pages 82 and 83.)
- Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling (3DIM)*, pages 145–152, 2001. (Cited on page 32.)
- Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-time 3d model acquisition. *ACM Transactions on Graphics (TOG)*, 21(3):438–446, 2002. (Cited on pages 14, 33, 64 and 65.)
- Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H.J. Kelly, and Andrew J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1352–1359, 2013. (Cited on page 65.)
- Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5):578–596, 2007. (Cited on pages 2 and 64.)
- Tanner Schmidt, Richard Newcombe, and Dieter Fox. Dart: Dense articulated real-time tracking. In *Robotics: Science and Systems (RSS)*, 2014. (Cited on pages 9, 10 and 11.)
- Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. Accurate, robust, and flexible real-time hand tracking. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 3633–3642, 2015. (Cited on pages 10, 12 and 15.)
- Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 116–124, 2011. (Cited on pages 8 and 107.)
- Leonid Sigal, Alexandru O. Balan, and Michael J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision (IJCV)*, 87:4–27, 2010. (Cited on page 18.)
- SKANECT. Skanect. <http://skanect.occipital.com>, 2016. Accessed: 17/08/2016. (Cited on pages 64, 73, 83 and 89.)

- Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Eurographics Symposium on Geometry Processing (SGP)*, pages 109–116, 2007. (Cited on pages 11 and 106.)
- Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *International Conference on Computer Vision (ICCV)*, pages 2456–2463, 2013. (Cited on pages 9, 10, 11, 15, 42, 45, 48 and 49.)
- Srinath Sridhar, Helge Rhodin, Hans-Peter Seidel, Antti Oulasvirta, and Christian Theobalt. Real-time hand tracking using a sum of anisotropic gaussians model. In *3DV*, pages 319–326, 2014. (Cited on pages 9 and 10.)
- Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and robust hand tracking using detection-guided optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3221, 2015. (Cited on page 10.)
- Srinath Sridhar, Franziska Mueller, Michael Zollhoefer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *European Conference on Computer Vision (ECCV)*, 2016. (Cited on pages 10, 13, 14, 15 and 107.)
- Björn Stenger, Paulo R.S. Mendonça, and Roberto Cipolla. Model-based 3d tracking of an articulated hand. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 310–315, 2001. (Cited on page 8.)
- Björn Stenger, Arasanathan Thayananthan, Philip H.S. Torr, and Roberto Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(9):1372–1384, 2006. (Cited on pages 18 and 20.)
- Jorge Stolfi. *Oriented Projective Geometry: A Framework for Geometric Computation*. Boston: Academic Press, 1991. (Cited on pages 20 and 33.)
- Carsten Stoll, Juergen Gall, Edilson de Aguiar, Sebastian Thrun, and Christian Theobalt. Video-based reconstruction of animatable human characters. *ACM Transactions on Graphics (TOG)*, 29(6):139:1–139:10, 2010. (Cited on page 82.)
- Jürgen Sturm, Vijay Pradeep, Cyrill Stachniss, Christian Plagemann, Kurt Konolige, and Wolfram Burgard. Learning kinematic models for articulated objects. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1851–1856, 2009. (Cited on page 83.)
- Jürgen Sturm, Cyrill Stachniss, and Wolfram Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research (JAIR)*, 41(2):477–626, 2011. (Cited on page 83.)

- Jürgen Sturm, Erik Bylow, Fredrik Kahl, and Daniel Cremers. Copyme3d: Scanning and printing persons in 3d. In *German Conference on Pattern Recognition (GCPR)*, pages 405–414, 2013. (Cited on pages 64 and 82.)
- Erik B. Sudderth, Michael I. Mandel, William T. Freeman, and Alan S. Willsky. Visual hand tracking using nonparametric belief propagation. In *Workshop on Generative Model Based Vision*, pages 189–189, 2004. (Cited on pages 9, 18 and 19.)
- James S. Supančič III, Grégory Rogez, Yi Yang, Jamie Shotton, and Deva Ramanan. Depth-based hand pose estimation: data, methods, and challenges. In *International Conference on Computer Vision (ICCV)*, pages 1868–1876, 2015. (Cited on pages 2, 4, 7 and 12.)
- Andrea Tagliasacchi, Ibraheem Alhashim, Matt Olson, and Hao Zhang. Mean curvature skeletons. *Computer Graphics Forum (CGF)*, 31(5):1735–1744, 2012. (Cited on pages 87 and 88.)
- Andrea Tagliasacchi, Matthias Schröder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. Robust articulated-icp for real-time hand tracking. *Eurographics Symposium on Geometry Processing (SGP)*, 34(5):101–114, 2015. (Cited on page 10.)
- David Joseph Tan, Thomas Cashman, Jonathan Taylor, Andrew Fitzgibbon, Daniel Tarlow, Sameh Khamis, Shahram Izadi, and Jamie Shotton. Fits like a glove: Rapid and reliable hand shape personalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5610–5619, 2016. (Cited on pages 12 and 105.)
- Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *International Conference on Computer Vision (ICCV)*, pages 3224–3231, 2013. (Cited on pages 10, 15 and 42.)
- Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3786–3793, 2014. (Cited on pages 15 and 42.)
- Danhang Tang, Jonathan Taylor, Pushmeet Kohli, Cem Keskin, Tae-Kyun Kim, and Jamie Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *International Conference on Computer Vision (ICCV)*, pages 3325–3333, 2015. (Cited on page 9.)
- Jonathan Taylor, Jamie Shotton, Toby Sharp, and Andrew Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 103–110, 2012. (Cited on page 108.)

- Jonathan Taylor, Richard Stebbing, Varun Ramakrishna, Cem Keskin, Jamie Shotton, Shahram Izadi, Aaron Hertzmann, and Andrew Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 644–651, 2014. (Cited on pages 9, 11, 12, 54, 105 and 107.)
- Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, Arran Topalian, Erroll Wood, Sameh Khamis, Pushmeet Kohli, Toby Sharp, Shahram Izadi, Richard Banks, Andrew Fitzgibbon, and Jamie Shotton. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (SIGGRAPH)*, 35(4), 2016. (Cited on pages 2, 10, 12, 105, 107 and 109.)
- Moritz Tenorth, Jan Bandouch, and Michael Beetz. The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition. In *International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences*, pages 1089–1096, 2009. (Cited on page 18.)
- Matthias Teschner, Stefan Kimmerle, Bruno Heidelberger, Gabriel Zachmann, Laks Raghupathi, Arnulph Fuhrmann, Marie-Paule Cani, François Faure, Nadia Magnenat-Thalmann, Wolfgang Strasser, and Pascal Volino. Collision detection for deformable objects. In *Eurographics*, pages 119–139, 2004. (Cited on page 33.)
- Arasanathan Thayananthan, Björn Stenger, Philip H.S. Torr, and Roberto Cipolla. Shape context and chamfer matching in cluttered scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 127–133, 2003. (Cited on pages 9 and 18.)
- Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European Conference on Computer Vision (ECCV)*, pages 356–369, 2010. (Cited on page 69.)
- Federico Tombari, Samuele Salti, and Luigi Di Stefano. A combined texture-shape descriptor for enhanced 3d feature matching. In *International Conference on Image Processing (ICIP)*, pages 809–812, 2011. (Cited on page 69.)
- Federico Tombari, Samuele Salti, and Luigi Di Stefano. Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision (IJCV)*, 102(1-3): 198–220, 2013. (Cited on page 69.)
- Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (TOG)*, 33(5):169:1–169:10, 2014. (Cited on pages 9, 10, 12, 15 and 43.)
- Phil Tresadern and Ian Reid. Articulated structure from motion by factorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1110–1115, 2005. (Cited on page 83.)

- Dimitrios Tzionas and Juergen Gall. A comparison of directional distances for hand pose estimation. In *German Conference on Pattern Recognition (GCPR)*, pages 131–141, 2013. (Cited on pages viivii, 4 and 15.)
- Dimitrios Tzionas and Juergen Gall. 3d object reconstruction from hand-object interactions. In *International Conference on Computer Vision (ICCV)*, pages 729–737, 2015. (Cited on pages viivii, 5, 13, 14 and 15.)
- Dimitrios Tzionas and Juergen Gall. Reconstructing articulated rigged models from rgb-d videos. In *Computer Vision – ECCV 2016 Workshops*, pages 620–633, 2016. (Cited on pages viivii and 5.)
- Dimitrios Tzionas, Abhilash Srikantha, Pablo Aponte, and Juergen Gall. Capturing hand motion with an rgb-d sensor, fusing a generative model with salient points. In *German Conference on Pattern Recognition (GCPR)*, pages 277–289, 2014. (Cited on pages viivii, 5 and 15.)
- Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision (IJCV)*, 118(2): 172–193, 2016. (Cited on pages viivii, 5, 10, 13, 14 and 15.)
- N.P. Van der Aa, Xinghan Luo, Geert-Jan Giezeman, Robby T. Tan, and Remco C. Veltkamp. Umpm benchmark: A multi-person dataset with synchronized video and motion capture data for evaluation of articulated human motion and interaction. In *Workshop on Human Interaction in Computer Vision*, pages 1264–1269, 2011. (Cited on page 18.)
- Jörg Vollmer, Robert Mencl, and Heinrich Mueller. Improved laplacian smoothing of noisy surface meshes. In *Computer Graphics Forum (CGF)*, pages 131–138, 1999. (Cited on page 70.)
- Robert Y. Wang and Jovan Popović. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics (TOG)*, 28(3):63:1–63:8, 2009. (Cited on pages 8 and 9.)
- Yangang Wang, Jianyuan Min, Jianjie Zhang, Yebin Liu, Feng Xu, Qionghai Dai, and Jinxiang Chai. Video-based hand manipulation capture through composite motion control. *ACM Transactions on Graphics (TOG)*, 32(4):43:1–43:14, 2013. (Cited on page 13.)
- David Weikersdorfer, David B. Adrian, Daniel Cremers, and Jörg Conradt. Event-based 3d slam with a depth-augmented dynamic vision sensor. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 359–364, 2014. (Cited on page 109.)
- Thibaut Weise, Bastian Leibe, and Luc Van Gool. Accurate and robust registration for in-hand modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. (Cited on pages 14, 64 and 65.)

- Thibaut Weise, Thomas Wismer, Bastian Leibe, and Luc Van Gool. Online loop closure for real-time interactive 3d scanning. *Computer Vision and Image Understanding (CVIU)*, 115(5):635–648, 2011. (Cited on pages 14, 64 and 65.)
- Aaron Wetzler, Ron Slossberg, and Ron Kimmel. Rule of thumb: Deep derotation for improved fingertip detection. In *British Machine Vision Conference (BMVC)*, 2015. (Cited on page 15.)
- Ying Wu, John Y. Lin, and Thomas S. Huang. Capturing natural hand articulation. In *International Conference on Computer Vision (ICCV)*, pages 426–432, 2001. (Cited on pages 8 and 9.)
- Chi Xu and Li Cheng. Efficient hand pose estimation from a single depth image. In *International Conference on Computer Vision (ICCV)*, pages 3456–3462, 2013. (Cited on page 15.)
- Chi Xu, Ashwin Nanjappa, Xiaowei Zhang, and Li Cheng. Estimate hand poses efficiently from single depth images. *International Journal of Computer Vision (IJCV)*, 116(1):21–45, 2016. (Cited on page 15.)
- Jingyu Yan and Marc Pollefeys. Automatic kinematic chain building from feature trajectories of articulated objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 712–719, 2006. (Cited on pages 82 and 83.)
- Jingyu Yan and Marc Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(5):865–877, 2008. (Cited on pages 82 and 83.)
- Mao Ye, Qing Zhang, Liang Wang, Jiejie Zhu, Ruigang Yang, and Juergen Gall. A survey on human motion analysis from depth data. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 149–187. Springer, 2013. (Cited on page 7.)
- Kaan Yücer, Oliver Wang, Alexander Sorkine-Hornung, and Olga Sorkine-Hornung. Reconstruction of articulated objects from a moving camera. In *ICCVW*, pages 823–831, 2015. (Cited on page 83.)
- Carl Yuheng Ren, Victor Prisacariu, David Murray, and Ian Reid. Star3d: Simultaneous tracking and reconstruction of 3d objects using rgb-d data. In *International Conference on Computer Vision (ICCV)*, pages 1561–1568, 2013. (Cited on pages 14 and 65.)
- Wenping Zhao, Jinxiang Chai, and Ying-Qing Xu. Combining marker-based mocap and rgb-d camera for acquiring high-fidelity hand motion data. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 33–42, 2012. (Cited on page 8.)

Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *ICCV Workshop on 3D Representation for Recognition (3dRR)*, pages 689–696, 2009. (Cited on page 69.)

Hanning Zhou and Thomas Huang. Okapi-chamfer matching for articulate object recognition. In *International Conference on Computer Vision (ICCV)*, pages 1026–1033, 2005. (Cited on pages 15 and 18.)