University of Rome “La Sapienza”
Department of Computer, Control and Management
Engineering ‘Antonio Ruberti’

Inverse Problem Theory in Shape and Action Modeling

PhD in Engineering in Computer Science

PhD Thesis

Valsamis Ntouskos

Supervisor: Prof. Fiora Pirri
Co-advisor: Prof. Aris Anagnostopoulos

Reviewers:
Prof. Marco Fratarcangeli
Prof. George Karras

Rome, Italy, June 2016

Abstract

In this thesis we consider shape and action modeling problems under the perspective of inverse problem theory. Inverse problem theory proposes a mathematical framework for solving model parameter estimation problems. Inverse problems are typically ill-posed, which makes their solution challenging. Regularization theory and Bayesian statistical methods, which are proposed in the context of inverse problem theory, provide suitable methods for dealing with ill-posed problems.

Regarding the application of inverse problem theory in shape and action modeling, we first discuss the problem of saliency prediction, considering a model proposed by the coherence theory of attention. According to coherence theory, salience regions emerge via proto-objects which we model using harmonic functions (thin-membranes). We also discuss the modeling of the 3D scene, as it is fundamental for extracting suitable scene features, which guide the generation of proto-objects.

The next application we consider is the problem of image fusion. In this context, we propose a variational image fusion framework, based on confidence driven total variation regularization, and we consider its application to the problem of depth image fusion, which is an important step in the dense 3D scene reconstruction pipeline.

The third problem we encounter regards action modeling, and in particular the recognition of human actions based on 3D data. Here, we employ a Bayesian non-parametric model to capture the idiosyncratic motions of the different body parts. Recognition is achieved by comparing the motion behaviors of the subject to a dictionary of behaviors for each action, learned by examples collected from other subjects.

Next, we consider the 3D modeling of articulated objects from images taken from the web, with application to the 3D modeling of animals. By decomposing the full object in rigid components and by considering different aspects of these components, we model the object up this hierarchy, in order to obtain a 3D model of the entire object. Single view 3D modeling as well as model registration is performed, based on regularization methods.

The last problem we consider, is the modeling of 3D specular (non-Lambertian) surfaces from a single image. To solve this challenging problem we propose a Bayesian non-parametric model for estimating the normal field of the surface from its appearance, by identifying the material of the surface. After computing an initial model of the surface, we apply regularization of its normal field considering also a photo-consistency constraint, in order to estimate the final shape of the surface.

Finally, we conclude this thesis by summarizing the most significant results and by suggesting future directions regarding the application of inverse problem theory to challenging computer vision problems, as the ones encountered in this work.

Acknowledgments

I am grateful to many people for supporting me during my PhD studies. Trying to thank them all is certainly a challenging task but I will do my best to thank everyone.

First of all, I would like to express my gratitude to my supervisor Prof. Fiora Pirri for all her support, guidance and encouragement during my graduate studies. I will always be grateful to her for giving me the opportunity to work in the field of computer vision, for which I have always been excited about, and I will always appreciate the exciting meetings and discussions on these very challenging problems. Her great passion about research and her dedication will always be an inspiration for me.

I would like also to thank all my colleagues at ALCOR Lab for the fruitful discussions and for their collaboration which made this research possible. In particular, I would like to thank Dr. Matia Pizzoli, Dr. Arnab Sinha, Bruno Cafaro, Fabrizio Natola and Marta Sanzari, who not only I see as colleagues but also as friends.

I am also grateful to my co-advisor Prof. Aris Anagnostopoulos and my ex-colleague at ALCOR Lab Dr. Panagiotis Papadakis, not only for the interesting discussions on research, but also for the great time when going out in Rome. I am also thankful to Prof. Marco Fratarcangeli for accepting to be my reviewer and for all his help and discussions during his affiliation with ALCOR Lab.

Additionally, I am very grateful to Prof. George Karras for accepting to be my reviewer and for introducing me to the world of research back in 2006, by proposing me a challenging Diploma thesis subject on camera calibration. I always remember the long discussions with his group at his house, together with Prof. Elli Petsa, Dr. Lazaros Grammatikopoulos, Dr. Ilias Kalisperakis, Dr. Antonis Prokos – and his dog Paris. Especially, I would like to thank Christos Stentoumis for the interesting discussions regarding our common research interests, as well as for our collaboration, together with Prof. Konstantinos Karantzalos, during the last year.

I am also grateful to the anonymous reviewers of our papers who have contributed with their comments in improving the quality of our publications.

I want also to thank my dear Greek friends Panos, Christos B., Vaggelis, George, Theodore, Kyriakos, and Christos P., for their continuous support and the great times and truly relaxing moments we share during my stays in Greece.

Before closing, I would like to express my profound gratitude to my parents Giannis and Katerina, and to my brother Spyros and his fiancée Anta, for their endless support through my graduate studies and beyond.

Finally, I would like to close the acknowledgments by expressing my deep love and sincere gratitude to my wife Zoe who has been there for me during all these years with her love and her smile. Thank you, my love.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The majority of Computer Vision problems can be characterized as ill-posed [10]. This follows from the particular nature of this class of problems where a set of parameters of a model, approximating a physical process, needs to be identified by data originating from noisy measurements. Inverse problems theory provides a sound mathematical framework for dealing with these problems. The relevance of inverse problem theory has been recognized early in the research on the field of computer vision. Early examples include edge detection and motion estimation [11]. In this thesis we propose some novel approaches, based on inverse problem theory, for dealing with challenging computer vision problems related to the reconstruction and modeling of shape and motion in three dimensions. In particular, the applications considered are saliency prediction using via proto-object generation, variational image fusion, action recognition based on 3D data, 3D modeling of articulated objects, and modeling of shapes via their reflective properties. In this chapter we provide a short introduction of inverse problem theory, and then review the problems considered in this thesis and the main characteristics of the methods proposed for dealing with them.

1.1 Inverse problem theory

In this section we present an introduction to inverse problems. This introduction is not strictly formal, aiming to provide an intuition about the scope of inverse problem theory, and how it is applied. More details can be found in [12] and [13].

All physical phenomena, and subsequently all physical systems, can be described by mathematical models. The complexity of the models may vary, in relation to the required level of accuracy and the knowledge of the system's physical properties. These models are defined by a set of parameters and a series of operations that are applied on these parameters. Usually, a categorization of model parameters is made, distinguishing them to input, output, and state parameters. This parametrization most of the times comes naturally by *causality*. Considering, without loss of generality, a physical system, one considers a flow of a physical quantity (e.g. power, current, information, etc.) from the system's input to its output. The physical quantity may also get transformed, according to the processing induced by the system. The operations acting on this physical quantity, depend both on the properties of the physical quantity as well as on the state of the system. All this information is captured by the parameters of the model.

The problem of predicting the output of the system, given the values of its state parameters and its input, is called the *forward* problem. Similarly, considering this

convention, an *inverse* problem, as its name suggests, regards the estimation of the input, and possibly also the state parameters of the system, given its output. As one might expect, inverse problems are usually more difficult to treat than forward problems. Considering the discussion above, this can be attributed to the fact that the estimation does not follow the causality of the involved operations. Due to this, one usually has to make additional assumptions in order to solve these problems.

Example 1.1 (Heat distribution [13])**.** Consider as an example the phenomenon of heat diffusion. For a given initial heat distribution on a metallic rod of unit length, it is easy to find the evolution of the heat in time by solving the differential equations of the heat diffusion model. The temperature distribution $u(x,t)$ satisfies the heat equation

$$\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial t} = 0, \quad 0 < x < 1,\ t > 0, \tag{1.1}$$

with boundary conditions $u(0,t) = u(1,t) = 0$ and with initial heat distribution given by $u(x,0) = u_0(x)$.

The inverse problem though, namely the estimation of the initial heat distribution at time $t = 0$ based on measurements at time $t > 0$, poses significant challenges.

Let us write first the solution in terms of its Fourier components,

$$u(x,t) = \sum_{n=1}^{\infty} c_n \exp(-n\pi^2 t) \sin n\pi x. \tag{1.2}$$

The coefficients c_n are the Fourier sine coefficients of the initial state u_0, i.e.,

$$u_0(x) = \sum_{n=1}^{\infty} c_n \sin n\pi x. \tag{1.3}$$

Thus, to determine u_0 one has to find the coefficients c_n from the final data. Assume now that we have two initial states $u_0^{(j)}, j = 1, 2$, that differ only by a single high-frequency component, i.e.,

$$u_0^{(1)}(x) - u_0^{(2)}(x) = c_N \sin N\pi x, \tag{1.4}$$

for N large. The corresponding solutions at the final time will differ by

$$u^{(1)}(x,T) - u^{(2)}(x,T) = c_N \exp(-(N\pi)^2 T) \sin N\pi x, \tag{1.5}$$

i.e., the difference in the final data for these two initial states is exponentially small. This suggests that any information about high-frequency components will be lost in the presence of measurement errors and/or noise.

In fact, a numerical simulation of the inverse heat diffusion shows that the solutions quickly become unstable and diverge giving meaningless initial heat distributions.

The difficulties encountered when dealing with inverse problems are mathematically captured by the notion of well-posedness. Well-posedness has been defined by Hadamard in [10] as follows

Definition 1.1 (Well-posedness)**.** A problem is well posed if *all* of the following conditions hold:

- a solution exists (existence);
- the solution is unique (uniqueness);
- the solution is continuous with respect to the data (continuity).

Analogously, if any of these conditions does not hold the problem is ill-posed.

As stated in the preamble, the vast majority of inverse problems are ill-posed, and, based on the mathematical definition above, it becomes clear that the solution of these problems is challenging. Inverse problem theory constitutes a mathematical framework for solving ill-posed problems.

The methods used in inverse problem theory can be distinguished into two main categories. The first category includes regularization methods. Regularization methods introduce additional constraints to the problem in order to make it well-posed. By making the problem well-posed, classical optimization approaches are then used in order to find the solution. The constraints enforced are usually decided based on high-level knowledge regarding the problem at hand.

Example 1.2. As an example consider the problem of linear model fitting. The error with respect to the selected model is given by

$$e = Ax - d. \tag{1.6}$$

Let us now make an additional assumption that the solution x must be "smooth". The smoothness assumption can be algebraically represented by the quantity $\|Kx\|$, with K a differential operator, e.g. the discrete gradient operator. Considering a least squares minimization approach, the solution can be found by minimizing the following quantity

$$F(x;\lambda) \coloneqq \|Ax - d\|^2 + \lambda \|Kx\|^2 . \tag{1.7}$$

The degree of the solution's smoothness depends on the regularization parameter λ. The higher the value of λ, the smoother the solution will be.

The second main category of inverse problems are Bayesian statistical methods. Under this perspective, ill-posedness is modeled as uncertainty on the parameters of the system. Probability theory and Bayesian statistics provide a sound mathematical framework to model this uncertainty in the parameter space. Usually, it is necessary to make assumptions about the distribution from which the parameters are sampled. In complete lack of data, one has to assume a prior distribution for these parameters. As soon as data are collected, Bayesian statistics allow to obtain an updated distribution, called posterior distribution, which captures the update in the parameter values uncertainty under the light of the new evidence.

Example 1.3. Let $x \in X$ describe the values of a parameter and $p(x) \sim \exp(-\lambda \|Kx\|^2)$ be a prior distribution on the parameter space X. Let us now assume that the observations are affected by Gaussian noise. The likelihood of identically and independently distributed (i.i.d.) data $\mathcal{D}$ given x, is given by $p(\mathcal{D}|x) \sim \exp(-\|Ax - d\|_\Sigma^2)$. By making use of Bayes rule, we can now estimate the likelihood of x given the observed data, namely

$$p(x|\mathcal{D}) = \frac{p(\mathcal{D}|x)p(x)}{p(\mathcal{D})}. \tag{1.8}$$

The denominator in equation (1.8) does not depend on the data and acts as a normalization constant, in order to guarantee that the posterior is a well defined probability distribution. Based in this observation, the posterior distribution can be written as

$$p(x|\mathcal{D}) \sim \exp(-\|Ax - d\|_\Sigma^2) \exp(-\lambda \|Kx\|^2) = \exp(-\|Ax - d\|_\Sigma^2 - \lambda \|Kx\|^2). \tag{1.9}$$

We note that in a wide extent it is possible to unify these two categories. One might see regularization methods as a special case of Bayesian statistics, as the regularization functions can be derived from suitable prior distributions. This is immediate for example for prior distributions of the exponential function as can be seen by the previous examples. In this case, regularization functionals are equivalent to the negative log-likelihood of corresponding posterior distributions, and hence the solution of the regularization problem corresponds to the Maximum A-posteriori (MAP) solution of the Bayesian problem. Bayesian statistics generalize regularization problems as in principle they can provide richer information about the problem's solution. In practice, though, this is true only for some classes of problems. In general, posterior distributions do not have a closed form. Additionally, their numerical estimation poses significant challenges. Nevertheless, there exist methods to estimate these distributions, most notably Variational Bayes and Markov Chain Monte Carlo (MCMC) methods [14].

The previous observation seems to suggest that regularization methods are superseded by Bayesian statistical methods. However, this is not the case due to some subtle considerations. In particular, in the previous discussion we have considered that the assumed probability distributions exist. This is always possible for finite-dimensional parameter spaces. In many applications though, the parameters that need to be determined assume functions as values, which lie in an infinite dimensional space. For infinite dimensional parameter spaces, existence of the probability distributions cannot always by guaranteed.

The most important class of distributions that are well defined in infinite dimensional spaces is this of Gaussian distributions. Proving that other distributions exist over infinite-dimensional spaces is still an open problem. An important example is Laplace distributions which have recently gained popularity due to their ability to model impulsive noise (see [15] for a contribution in this direction). This is also the case for various regularization functionals, like the Total Variation which will be described in this thesis.

Hence, due to the challenges encountered when dealing with probability distributions over infinite-dimensional spaces, regularization theory is widely considered in inverse problem theory for infinite-dimensional problems. Regularization methods model the problems by integral equations, and take advantage of functional analysis and measure theory to compute solutions and study properties of these models, as for example existence, uniqueness and continuity of the solutions.

Note, that we have not treated the problem of system modeling, namely how to decide which parameters describe the system and what their interaction is. This is a very

challenging task, as it is difficult to model mathematically, and usually mainly depends on the intuition of the modeler regarding the way the system operates. Statistical methods provide tools to verify whether a particular model explains better the observations, however even in these cases, the form of the model usually has to be specified up to some extent by the modeler.

1.2 Contributions

This section provides an overview of the problems that are treated in the following chapter. Here, after stating each problem, we present the main characteristics of the method employed for its solution and highlight the main contributions. The relations of the proposed methods with inverse problem theory are also discussed.

Saliency prediction

Regarding saliency prediction, we introduce a method to computationally model the proto-object structures proposed in the coherence theory of attention. The proposed model is based on experimentally collected data in dynamic 3D environments. Proto-objects are modeled by vibrating circular membranes whose initial displacement depends on features that are extracted from the 3D scene. Saliency then depends on the energy of the vibrating membranes. Evaluation is performed considering various subjects performing search tasks.

The contribution in this field is two-fold. First it regards the localization and 3D reconstruction of the environment by the cameras of the head-mounted, gaze estimation device called the *Gaze Machine* (GM) [16]. Several prototypes of this device have been constructed in ALCOR Lab, and the device is protected by an international patent (Nr. WO2009043927). A Structure from Motion (SfM) approach is used as described in Chapter 3 to obtain a 3D reconstruction of the environment and also provide the pose of the camera at each captured frame. The GM uses an additional camera pointing to the eye that wears it in order to comptue estimates of her/his Point of Regard (POR). By means of an initial calibration procedure [16], suitable parameters of the gaze estimation model are learned which allows to recover the gaze direction in the 3D space. Combining this information we are able to estimate gaze scan paths in the 3D environment. The second contribution, as described above, regards the computational estimation of proto-objects, which are proposed in the context of the coherence theory of attention [17]. Proto-objects are modeled via thin vibrating membranes, which are excited by the scene features. Parameter estimation of the proposed models is based on the principles of inverse problem theory.

Confidence driven image fusion

In the context of image fusion we introduce a novel TGV regularization model which allows for the joint estimation of the fused values together with confidence values regarding the input data, which lead to spatially adaptive regularization effects. We show that the introduced model is biconvex in the considered variables, and provide suitable adaptations of non-smooth optimization algorithms in order to find its optimal solutions.

The contribution lies both in the formulation of the fusion model, as described above, as well as the adaptation of the non-smooth optimization algorithms for the case of

biconvex problems. Regarding the contribution in the algorithmic front, we have performed a convergence analysis of algorithms belonging to three well-known non-smooth optimization classes of algorithms, namely Alternate Convex Search (ACS) [18], Alternate Minimization Algorithms (AMA) [19], and Primal-Dual Hybrid Gradient Algorithms (PDHG) [20]. The analysis of the proposed model properties relies largely on inverse problem theory. We demonstrate the effectiveness of the proposed model by applying it to the problem of depth image fusion, namely the process of fusing together several noisy depth images to a single more accurate depth image. The results show that the proposed model is capable of recovering accurate depth images even from input data severely degraded by noise.

MoCap based action recognition

The third problem we address is that of recognizing human actions from 3D pose data. In particular, we consider sequences of human poses, represented as the positions of the joints of the subject performing a single action. The objective is to identify the action executed by the subject, considering only a limited number of poses taken from the whole sequence. This is a challenging problem as the temporal relation of the poses cannot be exploited in general. We have considered a natural decomposition of the human body in 6 groups, namely Head, Torso, Left/Right Arms, Left/Right Legs. For each of these groups we identify its principal direction by performing Principal Geodesic Analysis (PGA) on the $SE(3)$ manifold. The principal direction is then used as feature in order to obtain a clustering of the idiosyncratic motion behaviors of each body group, associated to each type of action. In this way, we associate to each action the corresponding motion behaviors of each group, as these are captured in the training data. At testing time, given a set of poses sampled from a specific action, we apply the decomposition in pose groups, compute the corresponding features using PGA and then predict the most likely cluster in which each frame corresponds. Combining the information from all samples and all groups using Maximum a-posteriori Inference, we are able to identify the most likely action performed by the subject. The results show that the proposed method provides accurate classification of the performed actions, even when using a limited number of motion samples. More importantly, the model allows for real-time inference of the performed action. This allows its use in near real-time applications, as the computation time is dominated by the calculation of the PGA-based motion features.

The main contributions in this work are the introduction of the PGA-based features for capturing the idiosyncratic behaviors of each group and the use of non-parametric Bayesian mixture models, which fall into the scope of inverse problem theory, to identify the main modes of these behaviors from a given set of data. This makes possible the near real-time classification from non sequential poses as discussed above.

Component wise articulated object modeling

We study also the challenging problem of modeling articulated objects from images taken from the Web. We consider that the input images are not of the same object, but rather from objects belonging to the same class. If the individual instances of the class are similar, hence the class shows low intra-class variation, then it is possible to obtain a 3D model of the object, generalizing individual characteristics and details. We consider animals which in general fall in this scope. First, we consider a natural

decomposition of the articulated model into components, which are considered rigid. Then, for each component we require a certain number of distinct views in order to compute the model. We call these different views *aspects* of the model components. Considering this input data, our method is divided in three main steps. First, we obtain a 3D model corresponding to each aspect by a single-view modeling approach. Then we obtain representative 3D models of each component by registering together the 3D models of all its aspects. Finally, we reassemble the model of the whole object by considering an image of the object in a reference pose.

There are three main contributions in this work, which rely on inverse problem theory. The first regards the introduction of a single image aspect modeling approach based on finite-element modeling of surfaces. The second contribution regards an approach for registering together the model aspects in order to obtain 3D models of the components and is based on the optimization of a non-smooth functional on a manifold. The third contribution is the introduction of a global optimization approach for reassembling the components into the full model. The introduced approach is based on the relation of the contour of each component in two or more reference views with the contour generator on the 3D models of the components.

Single image surface modeling based on BRDF

The last problem we consider is also in the context of surface and object modeling. We propose a method for estimating the 3D shape of objects from a single image using their reflective properties. More specifically, we consider a set of materials described by their reflective properties, which are captured by the bidirectional reflectance distribution function (BRDF) [21]. We generate 3D renderings of a large number of surfaces, considering different BRDF functions as well as ambient lighting environments. Subsequently, we divide these renderings to a set of interchangable image patches, accompanied by a map associating to each point of the patch the corresponding surface normal of the surface. Then, by employing an auto-encoder, we automatically extract appearance features of the patches, which are next provided to a non-parametric Bayesian mixture model, based on Dirichlet Processes, which provides a clustering of these features. At testing phase, we take as input a segmented image of an object made of a material among those considered previously. The input image is then divided in patches, and by using the predictive distribution we estimate first the most likely BRDF function corresponding to the surface (e.g. its material) and, at a second level, the most likely cluster in which each patch belongs. By recovering the cluster we then assign the cluster's most representative normal field to the given patch. Repeating this for all patches we estimate the normal field of the whole object with respect to the image view, and we use this information to compute an initial 3D surface of the object. The final shape of the 3D object is obtained by enforcing photoconsistency of the rendered surface with the input image, considering a TV regularization of the surface normal field. The results show that our method is able to faithfully model the surface of objects made of various highly specular materials, like steel, brass, aluminum, PVC, and plastic, as well as to correctly model concave surfaces.

The contributions lie in different directions in this work. First of all, in the process of obtaining a clustering of the patch appearance based on non-parametric Bayes mixture models and associating a property, in this case the normal map, to each cluster. Secondly, in the use of auto-encoders to capture the variance of the patch appearance

under different illumination conditions. Thirdly, in the proposed modeling of the initial surface based on constraints deriving jointly from the normal field, the curvature field and the object imaged contour, based on a finite-element approach. Finally, in the way photoconsistency is enforced, considering a TV smoothing of the normal field, using a representation of the normals, according to [22], which ensures integrability of the normal field.

1.3 Publications

This thesis is based on the contributions presented in the following publications.

- V. Ntouskos, F. Pirri, M. Pizzoli, A. Sinha, and B. Cafaro, "Saliency prediction in the coherence theory of attention," *Journal of Biologically Inspired Cognitive Architectures*, vol. 5, pp. 10–28 , 2013. [Chapter 3]

- V. Ntouskos and F. Pirri, "Confidence Driven TGV Fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Under review). [Chapter 4]

- F. Natola, V. Ntouskos, M. Sanzari and F. Pirri, "Bayesian non-parametric inference for manifold based MoCap representation," *In Proceedings of the International Conference on Computer Vision*, pp. 4606–4614 , 2015. [Chapter 5]

- V. Ntouskos, M. Sanzari, B. Cafaro, F. Nardi, F. Natola, F. Pirri and M. Ruiz, "Component-wise modeling of articulated objects," *In Proceedings of the International Conference on Computer Vision*, pp. 2327–2335 , 2015. [Chapter 6]

- F. Natola, V. Ntouskos, F. Pirri, M. Sanzari, "Single image object modeling based on BRDF and r-surflets learning," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. [Chapter 7]

1.4 Structure

In Chapter 2 we introduce the mathematical notation and the most important notions and facts which are used in this thesis. This part is written with the intention of being as short but still as comprehensive as possible in order to make this manuscript self-contained. Nevertheless, in each chapter we restate the most relative definitions to facilitate reading of individual chapters.

Chapters 3 to 7 present the contributions in the research problems discussed above. More specifically, Chapter 3 discusses the problem of saliency prediction based on 3D data. Chapter 4 regards confidence driven image fusion with spatially adaptive regularization effects. Chapter 5 presents the application of non-parametric Bayesian mixture models for the recognition of human actions from 3D data. Chapter 6 presents our approach for component-wise 3D modeling of articulated objects based on images taken from the Web. Additionally, Chapter 7 presents our approach for 3D surface modeling from a single image, based on BRDF functions.

Chapter 8 closes the main part of this thesis, providing some general concluding remarks and indicating further research directions to be pursued.

In Appendix A, details of my PhD career are provided, starting with a list of exams and seminars taken during this period. Additionally, it contains a complete list of publications during my PhD studies. Moreover, my participation to European projects, PhD Summer Schools, and international conferences and workshops is reported.

Chapter 2

Mathematical Preliminaries

In this chapter we introduce notions and properties that will be used in the following chapters. First, we discuss variational calculus and we present a quick review of convex analysis and convex optimization, as well as some basic notions of biconvex problems. Additionally, we provide a quick introduction of Total Variation (TV) regularization as well as the Total Generalized Variation functional which is a generalization of TV allowing the reconstruction of higher order polynomial signals. Moreover, some basic notions of Lie groups and Lie algebra theory are reviewed. Finally, the chapter ends with a discussion on Non-parametric Bayes models and in particular Dirichlet Processes Mixtures and their application to clustering problems.

In the following we make use of the following notation. $\mathcal{U}$ is a Banach space, with associated norm $\|\cdot\|_{\mathcal{U}}$ and we denote its dual space as $\mathcal{U}^*$. We denote $\mathcal{H}$ a Hilbert space, equipped with the inner product $\langle\cdot,\cdot\rangle$ and associated norm $\|\cdot\| \coloneqq \langle\cdot,\cdot\rangle^{1/2}$.

2.1 Calculus of Variations

As discussed in Chapter 1, solutions to inverse problems in which we are interested in this thesis typically are functions which satisfy certain constraints. In order to estimate these functions, an optimization problem needs to be solved. The calculus of variations is the field of mathematics studying the problem of estimating functions, i.e. infinite dimensional objects which are optimal under certain constraints. A typical example is the *Brachistocrone* curve problem, raised and solved by Johann Bernouli in 1697, laying the foundations of calculus of variations. In this section we discuss some definitions and results from the calculus of variations. More details can be found in [23] or [24].

Let $\overline{\mathbb{R}} \coloneqq \mathbb{R} \cup \{-\infty, +\infty\}$ be the extended real line, a functional is then defined as a mapping from $\mathcal{U}$ to $\overline{\mathbb{R}}$. A proper functional is defined as follows.

Definition 2.1 (Proper functional)**.** A functional $F : \mathcal{U} \mapsto \overline{\mathbb{R}}$ is called proper if $F(u) \neq -\infty$ for all $u \in \mathcal{U}$ and there exists at least one $u \in \mathcal{U}$ with $F(u) \neq +\infty$. The set

$$\operatorname{dom} F \coloneqq \{u \in \mathcal{U} \mid F(u) < \infty\} \tag{2.1}$$

is called the effective domain of the functional F.

Definition 2.2 (Directional derivative)**.** Let $F : U \subseteq \mathcal{U} \mapsto \mathcal{V}$ be an operator between the Banach spaces $\mathcal{U}$, $\mathcal{V}$ and U a non empty subset of $\mathcal{U}$. Then the directional derivative at $u \in U$ in direction v is defined as

$$dF(u, v) = \lim_{t \to 0} \frac{F(u + tv) - F(u)}{t}, \tag{2.2}$$

if the limit exists. If the directional derivative exists for all $v \in U$ then F is called directionally differentiable. F is called *Gâteaux differentiable* if the directional derivative $F'(u) : \mathcal{U} \ni v \mapsto dF(u, v) \in V$ is bounded and linear, i.e. $F'(u) \in \mathcal{L}(\mathcal{U}, \mathcal{V})$. F is called *Fréchet differentiable*, if additionally the following approximation condition holds

$$\left\| F(u+v) - F(u) - F'(u)v \right\|_{\mathcal{V}} = o(\|v\|_{\mathcal{U}}), \text{ for } \|v\| \to 0. \tag{2.3}$$

Note 2.1. *If F is Gâteaux differentiable in a neighborhood of u and $F'(u)$ is continuous at u then F is also Fréchet differentiable at u.*

Fréchet differentiability generalizes the common concept of differentiability of real-valued functions for the case of extended real-valued functionals, and has a central role in the calculus of variations.

We also define the notion of semi-continuity which is a weaker concept of continuity.

Definition 2.3 (Semi-continuity)**.** A functional $F : \mathcal{U} \mapsto \overline{\mathbb{R}}$ is called *lower semi-continuous* (l.d.c.) if for all $u \in \mathcal{U}$

$$\liminf_{v \to u} F(v) \geq F(u). \tag{2.4}$$

Analogously, F is *upper semi-continuity* if $-F$ is lower semi-continuous. F is continuous at u if and only if F is both upper and lower semi-continuous at u.

2.2 Convex Optimization

As discussed above, the methods considered in this thesis often lead to optimization problems of the following form

$$\min_{u \in U \subseteq \mathcal{U}} F(u). \tag{2.5}$$

Typically, functional F will be convex. For F non convex, usually a relaxed problem will be formed leading to a sub-problem which is convex. Hence, the tools provided by convex analysis and convex optimization theory will be extensively used throughout this work. In this section we provide some of the most important concepts and theorems of convex optimization theory. For more details the reader is referred to [25], and [26].

2.2.1 Convex Analysis

Definition 2.4 (Convex sets)**.** A set $C \subseteq \mathcal{U}$ is called *convex* if

$$\alpha x + (1-\alpha)y, \quad \forall x, y \in C, \ \forall \alpha \in [0, 1]. \tag{2.6}$$

Definition 2.5 (Convex functionals)**.** Let $C \subseteq \mathcal{U}$ be a convex set. The functional $F : C \mapsto \overline{\mathbb{R}}$ is called *convex* if

$$F(\alpha x + (1-\alpha)y) \leq \alpha F(x) + (1-\alpha)F(y), \quad \forall x, y, \in C, \ \forall \alpha \in [0, 1]. \tag{2.7}$$

Definition 2.6 (Homogeneous functional)**.** A functional $F : \mathcal{U} \mapsto \overline{\mathbb{R}}$ is called homogeneous of order k, if

$$F(\alpha u) = \alpha^k F(u), \quad \forall \alpha > 0, \tag{2.8}$$

F is called *one-homogeneous* if (2.8) holds for $k = 1$.

The following propositions are useful when operations with convex functionals (the corresponding proofs are provided in [25]).

Proposition 2.1. *Let $F : X \subseteq \mathcal{H} \mapsto \overline{\mathbb{R}}$ be a proper convex functional, and an operator $A \in \mathcal{B}(X)$, with $\mathcal{B}(X)$ the space of bounded linear operators from X to X with domain defined on X. Then the functional $G : X \mapsto \overline{\mathbb{R}}$ defined as*

$$G(x) = F(Ax), \quad \forall x \in X, \tag{2.9}$$

is convex.

The previous results leads also to the next proposition.

Proposition 2.2. *Let $F_i : X \subseteq \mathcal{H} \mapsto \overline{\mathbb{R}}$, $i = 1, \dots, m$, be proper convex functionals on X, and let $\gamma_1, \dots, \gamma_m > 0$. Then the functional $G : X \mapsto \overline{\mathbb{R}}$ defined as*

$$G(x) = \gamma_1 F_1(x) + \cdots + \gamma_m F_m(x), \quad \forall x \in X, \tag{2.10}$$

is convex.

Proposition 2.3. *Let $F_i : X \subseteq \mathcal{U} \mapsto \overline{\mathbb{R}}$ be proper convex functionals for $i \in I \subset \mathbb{N}$. Then the functional $G : X \mapsto \overline{\mathbb{R}}$ defined as*

$$G(x) = \inf_{i \in I} F_i(x), \tag{2.11}$$

is convex.

2.2.2 Subgradients and optimality conditions

An important concept which can be introduced for convex functionals is the subgradient. Subgradients in practice extend the notion of differentiability for non Fréchet-differentiable, and hence non-smooth, convex functionals.

Definition 2.7 (Subgradient and subdifferential)**.** Let $F : U \subseteq \mathcal{U} \mapsto \overline{\mathbb{R}}$ be a proper convex functional. A vector $p \in \mathcal{U}^*$ is called a *subgradient* of F at a point $u \in U$ if

$$F(v) \geq F(u) + \langle p, v - u \rangle, \quad \forall v \in \mathcal{U}. \tag{2.12}$$

$\partial F(u)$ denotes the set of all subgradients at $u \in U$ and is called *subdifferential* of F at u. By convention $\partial F(u)$ is considered empty for all $u \notin \operatorname{dom} F$.

Remark 2.1. *If F is convex and one-homogeneous, the subdifferential can be written as*

$$\partial F(u) \coloneqq \{p \in \mathcal{U}^* \mid F(u) = \langle p, u \rangle, F(v) \geq \langle p, v \rangle, \ \forall v \in \mathcal{U}\}. \tag{2.13}$$

Proof. This directly follows by considering $v = 0$ and $v = 2u$ and the definition of one-homogeneity. □

In general $\partial F(u)$ forms a closed and convex set as in the following example.

Example 2.1 (Subgradient of L_1 norm)**.** The Euclidean norm $F : \mathbb{R}^N \mapsto \mathbb{R}$, $F(x) = \|x\|_{L_1}$ is not Fréchet-differentiable in $x = 0$, but it is subdifferentiable at every $x \in \mathbb{R}^N$. The subdifferential is given by

$$\partial F(x) = \begin{cases} \{p \in \mathbb{R}^n \mid \|y\|_{L_1} \geq \langle p, y \rangle,\ \forall y \in \mathbb{R}^n\} & \text{if } x = 0 \\ \frac{x}{\|x\|_{L_1}} & \text{otherwise.} \end{cases} \tag{2.14}$$

Thus in $x = 0$ the subdifferential consists of the whole Euclidean unit ball. For $N = 1$ this corresponds to the interval $[-1, 1]$.

If $F : \mathcal{U} \mapsto \overline{\mathbb{R}}$ is also Fréchet-differentiable, then $\partial F(u)$ is a singleton and in particular

$$\partial F(u) = \{F'(u)\}. \tag{2.15}$$

We now give the definition of the convex conjugate of a functional, called also Legendre-Fenchel transform, which is typically used to obtain the primal-dual form of a convex optimization problem.

Definition 2.8 (Convex conjugate)**.** Let $F : \mathcal{U} \mapsto \overline{\mathbb{R}}$ be a general extended real-valued functional (not necessarily convex). Its *convex conjugate* $F^* : \mathcal{U}^* \mapsto \overline{\mathbb{R}}$ is defined as

$$F^*(p) = \sup_{u \in \mathcal{U}} \{\langle u, p \rangle - F(u)\}. \tag{2.16}$$

The convex conjugate is always convex, as it corresponds to the point-wise supremum of a collection of affine functions (see also Proposition 2.3). The *double conjugate* functional is denoted by F^{**} and is given by

$$F^{**}(u) = \sup_{p \in \mathcal{U}^*} \{\langle u, p \rangle - F^*(p)\}. \tag{2.17}$$

Note 2.2. *In general $F^{**}(u) = (\check{\text{cl}})F(u)$ holds, where $\check{\text{cl}}\, F$ denotes the convex closure of F. If F additionally is a proper convex function then $F^{**}(u) = F(u)$.*

We provide now some basic rules of subdifferential calculus considering functionals defined in a Hilbert space. For more details see [25] and [27].

Definition 2.9 (Affine set)**.** We recall that a set $X \subseteq \mathcal{U}$ is affine if it contains all the linear combinations of pairs of points $x, y \in X$.

Definition 2.10 (Affine hull)**.** The affine hull of $X \subseteq \mathcal{H}$, denoted as $\text{aff}\, X$ is the intersection of all affine sets that contain X.

Definition 2.11 (Relative Interior)**.** Let C be a non-empty convex set. A point $x \in C$ belongs to the relative interior of C if there exists an open sphere S centered at x, such $S \cap \text{aff}\, C \subseteq C$.

Proposition 2.4 (Chain Rule)**.** *Let $F : X \subseteq \mathcal{H} \mapsto \overline{\mathbb{R}}$ be a convex extended real-valued functional, and let $A \in \mathcal{B}(X)$. Assume that the functional G given by*

$$G(x) = F(Ax) \tag{2.18}$$

is proper. Then

$$\partial G(x) \supset A^* \partial F(Ax), \quad \forall x \in X, \tag{2.19}$$

with A^* the adjoint of A. If additionally $(\operatorname{ran} A \cap \operatorname{ri}(\operatorname{dom} F)) \neq \emptyset$ holds, with $\operatorname{ran} A$ the range of the operator A, then we have

$$\partial G(x) = A^* \partial F(Ax), \quad \forall x \in X. \tag{2.20}$$

Proposition 2.5 (Subdifferential of Sum of Functionals)**.** *Let $F_i : X \subseteq \mathcal{H} \mapsto \overline{\mathbb{R}}$, $i = 1, \ldots, m$, be proper convex functionals, and assume that the functional $G = F_1 + \cdots + F_m$ is also proper. Then*

$$\partial G(x) \supset \partial F_1(x) + \cdots + F_m(x), \quad \forall x \in X. \tag{2.21}$$

If additionally, $\cap_{i=1}^m \operatorname{ri}(\operatorname{dom} F_i) \neq \emptyset$, then

$$\partial G(x) = \partial F_1(x) + \cdots + F_m(x), \quad \forall x \in X. \tag{2.22}$$

Proposition 2.6 (Optimality condition)**.** *Let $F : X \mapsto \overline{\mathbb{R}}$ be a proper convex functional and assume that $\operatorname{ri}(\operatorname{dom} F) \cap \operatorname{ri}(X) \neq \emptyset$. Then a vector $\hat{x}$ is a minimizer of F over X if and only if there exists $p \in \partial F(\hat{x})$ such that*

$$\langle p, x - \hat{x} \rangle \geq 0, \quad x \in X. \tag{2.23}$$

In particular, if $0 \ni \partial F(\hat{x})$ the previous relation is trivially satisfied.

Theorem 2.1. *Let $F : \mathcal{U} \mapsto \overline{\mathbb{R}}$ and*

$$\min_{u \in \mathcal{U}} F(u) \tag{2.24}$$

be a strictly convex optimization problem. Then there exists at most one local minimum, which is a global minimum.

Proof. ([28]) Assume that u is a local minimum of F but not a global minimum. Then $\hat{u} \in \mathcal{U}$ exists with $F(\hat{u}) < F(u)$. Let us define

$$u_\alpha := \alpha \hat{u} + (1 - \alpha), \quad u \in \mathcal{U}, \forall \alpha \in [0, 1]. \tag{2.25}$$

Due to the (strict) convexity of F we have

$$F(u_\alpha) \leq \alpha F(\hat{u}) + (1 - \alpha) F(u) < F(u) : \tag{2.26}$$

Since $u_\alpha \to u$ as $\alpha \to 0$, this is a contradiction to u being a local minimum and hence u is also a global minimum. Now let $v, w \in \mathcal{U}$ be two global minima of F. For $v \neq w$ this implies

$$F(\alpha v + (1 - \alpha) w) < \alpha F(v) + (1 - \alpha) F(w) = \inf_{u \in \mathcal{U}} F(u), \tag{2.27}$$

for $\alpha \in]0, 1[$, which is a contradiction and hence $v = w$. □

2.2.3 Special cases of non-convex functionals

Definition 2.12 (Semiconvexity and strong convexity [29])**.** A lower semicontinuous functional $F : \mathcal{U} \to \overline{\mathbb{R}}$ is called ω-*semiconvex* if $F + \frac{\omega}{2}\|\cdot\|^2$ is convex.
A lower semicontinuous functional $F : \mathcal{U} \to \overline{\mathbb{R}}$ is called *c-strongly convex* if for all $u_1, u_2 \in \mathcal{U}$, $q_1 \in \partial F(u_1)$, $q_2 \in \partial F(u_2)$, it holds that

$$\langle u_1 - u_2, q_1 - q_2 \rangle \geq c\|u_1 - u_2\|^2 .$$

Definition 2.13 (Biconvex set [18])**.** Let $U, V \subseteq \mathcal{U}$. The set $B \subseteq U \times V$ is called a biconvex set on $U \times V$ or biconvex for short, if B_u is convex for every $u \in U$ and B_v is convex for every $v \in V$.

Definition 2.14 (Biconvex functional)**.** A functional $F : B \to \overline{\mathbb{R}}$ on a biconvex set $B \subseteq \mathcal{U} \times \mathcal{V}$ is called a biconvex function on B or biconvex for short, if

$$F_u(\cdot) \coloneqq F(u, \cdot) : B_u \to \overline{\mathbb{R}}$$

is a convex function on B_u for every fixed $u \in \mathcal{U}$ and

$$F_v(\cdot) \coloneqq F(\cdot, v) : B_v \to \overline{\mathbb{R}}$$

is a convex function on B_v for every fixed $v \in \mathcal{V}$.

Definition 2.15 (Partial optimum)**.** Let $F : B \mapsto R$ be a biconvex functional. Then $(u^*, v^*) \in B$ is called a *partial optimum* of F on B, if

$$F(u^*, v^*) \leq F(u, v^*)\ \forall u \in B_{v^*} \quad \text{and} \quad F(u^*, v^*) \leq F(u^*, v)\ \forall v \in B_{u^*}. \tag{2.28}$$

The following theorem extends Theorems 4.1 and 4.2 of [18] to the case of non-smooth functions.

Theorem 2.2. *Let B be a biconvex set and let $F : B \mapsto \mathbb{R}$ be a biconvex functional. Then a point $z \coloneqq (x, y) \in \mathrm{ri}(B)$ is a stationary point of F if and only if it is a partial minimum.*

Proof. The forward direction is easily shown by using the definition of partial optimum. In particular, considering a partial minimum $\zeta \in \mathrm{ri}(B)$, then the optimality condition $0 \in \partial F$ holds. Hence, ζ is a stationary point.

The reverse direction is shown as follows. Let $\hat{z} = (\hat{x}, \hat{y}) \in \mathrm{ri}(B)$ be a stationary point of F. For $y = \hat{y}$, the functional $F_{\hat{y}} : B_{\hat{y}} \mapsto \mathbb{R}$ is convex. Since $\hat{x}$ is a stationary point of $F_{\hat{y}}$, then $0 \in \partial F_{\hat{y}}(\hat{x})$. From the definition of subgradient then we have

$$F_{\hat{y}}(x) \geq F_{\hat{y}}(\hat{x}) + \langle 0, x - \hat{x} \rangle = F_{\hat{y}}(\hat{x}),\ \forall x \in B_{\hat{y}}. \tag{2.29}$$

Analogously, for $x = \hat{x}$ we obtain

$$F_{\hat{x}}(y) \geq F_{\hat{x}}(\hat{y}),\ \forall y \in B_{\hat{x}}. \tag{2.30}$$

Hence, $\hat{z}$ is a partial minimum. □

2.3 Total Variation

Let $u(x) : \Omega \to \mathbb{R}$ a scalar function with Ω an open set of $\mathbb{R}^d$, the definition of total variation is:

$$TV(u) = \int_{\Omega} |\nabla u| \, \mathrm{d}x. \tag{2.31}$$

This definition is valid for u in the Sobolev space $W^{1,1}(\Omega)$, hence u cannot have any jump discontinuities. We remind that a Sobolev space $W^{l,p}$ is a Hilbert space with member functions that are l-weakly differentiable and L_p measurable, i.e. $(\int |u|^p)^{1/p} < +\infty$ (see [24] for a detailed definition).

The intensity values of natural images typically show jump discontinuities, thus a more general definition is required in these cases, in order to extend the definition of TV to these cases. An extended definition of total variation can be obtained by duality:

$$TV(u) \equiv \sup_{\substack{p \in C_0^{\infty}(\Omega, \mathbb{R}^d) \\ \|p\|_{\infty} \leq 1}} \int_{\Omega} u \, \mathrm{div}(p) \, \mathrm{d}x. \tag{2.32}$$

Based on this definition the space of functions of *bounded variation* (BV) can be defined:

$$BV(\Omega) \equiv \left\{ u \in L^1(\Omega) \mid TV(u) < \infty \right\}, \tag{2.33}$$

which is equipped with the norm:

$$\|u\|_{BV} = \|u\|_{L^1(\Omega)} + TV(u). \tag{2.34}$$

Using the previous definitions of TV, various image restoration tasks have been formulated as variational problems which can be summarized in the following general form:

$$\min_{u \in BV(\Omega)} \left\{ \lambda TV(u) + H(u, f) \right\}, \tag{2.35}$$

with $H(u, f)$ a convex, lower-semicontinuous (l.s.c.) functional representing the fidelity term or error measure.

The previous expression can be further generalized for other TV-like regularization terms (e.g. higher-order TV and Total Generalized Variation [30]) as

$$\min_{u \in BV(\Omega)} \left\{ \lambda J(u) + H(u, f) \right\}, \tag{2.36}$$

with $J(u)$ a general one-homogeneous, convex, l.s.c. regularization functional.

More details regarding functions of bounded variation and total variation for image restoration can be found in [31] and [32].

Intuitively, the TV functional enforces function u to be smooth almost everywhere, while it allows for a small number of jump discontinuities to occur. More specifically, it enforces u to be a piece-wise constant function. This suits very well imaging applications, as natural images tend to contain large smooth areas, but contain also a few sharp edges. Examples of TV regularization applications and the results obtained are shown in the following section.

2.3.1 Examples of TV regularization applications

Rudin-Osher-Fatemi (ROF) model

Rudin, Osher and Fatemi proposed this model in [33] for recovering images affected by noise with known variance. The noise variance was considered as a hard constraint in the original model. Subsequent works, however, usually consider this as a weak constraint resulting in the following minimization

$$\min_{u \in BV} \lambda TV(u) + \int_{\Omega} \|u - f\|_{L_2}^2 \, dx. \tag{2.37}$$

The model assumes additive Gaussian noise affecting the measurement f of the noise-free signal u. The ability of the model to remove high-frequency noise from the images while maintaining sharp edges has gained a lot of interest.

Several refinements of the original model have been proposed in order to resolve some of its shortcomings. One problem is the systematic error affecting its solutions which, in the case of images, leads to a loss of contrast. A solution to this problem was proposed in [34], by replacing the L_2 norm based error measure with the Bregman distance. Moreover, [35] has shown that TV regularization with L_1 norm based fidelity term does not suffer from contrast loss.

Another problem, which affects all models based on first-order TV, is the so-called "stair-casing" effect. The solution computed by TV models tends is a piece-wise constant function. This is caused by the inability of the first-order TV to reconstruct affine or higher polynomial order segments. A solution to this problem is to include higher-order TV terms. This has been studied by [36] and more recently in [30] which introduced Total Generalized Variation.

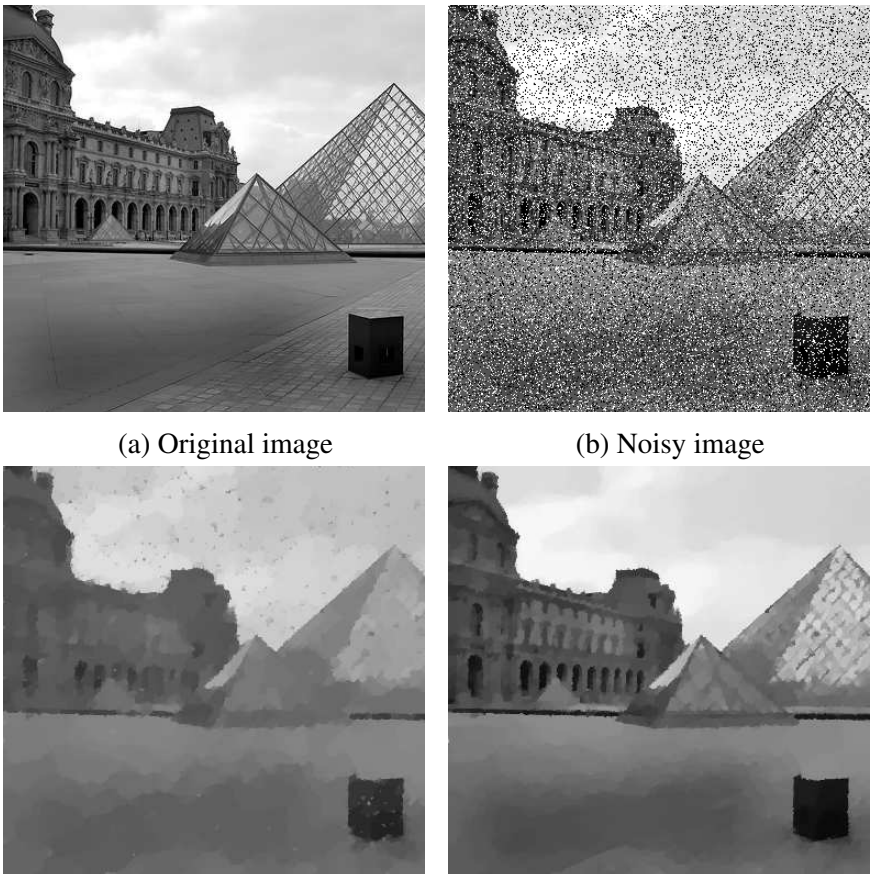
TV-L1 model

This model is an adaptation of the ROF model in the presence of impulsive noise. In this case the additive noise is considered to be sampled by a Laplace distribution, which results to an error measure based on the L_1 norm. Thus, the respective model is

$$\min_{u \in BV} \lambda TV(u) + \int_{\Omega} \|u - f\|_{L_1} \, dx. \tag{2.38}$$

One interesting property of this model is that, unlike the ROF model, its solution does not suffer from a systematic error. Moreover, it better models impulsive noise and for this reason it is also more robust against outliers.

An example of an image affected by impulsive noise and its TV-based reconstructions are presented in Figure 2.1 [32]. In Figure 2.1a the original image is displayed, while Figure 2.1b shows the image affected by noise. The solution of the ROF model is shown in Figure 2.1c while the one obtained by TV-L1 in Figure 2.1d. One can see that both models are able to remove large part of the noise while preserving image edges. In this case, the $TV - L_1$ model performs better, which can be attributed to the impulsive nature of the noise. One can also observe that the $TV - L_1$ denoised image has a higher contrast with respect to the ROF denoised image.



(a) Original image (b) Noisy image

(c) ROF denoising (d) TV-L1 denoising

Figure 2.1: Comparison of ROF and TV-L1 models for image denoising (images taken from [32]).

TV deblurring

An interesting application of TV models is image deblurring. In this case it is assumed that the original signal is convolved with a kernel which represents the *point spread function* (PSF). In practice, the PSF describes how each pixel is averaged with its neighbours producing the final image. In case PSF is known, or has been estimated, the original image can be estimated by solving the following minimization problem

$$\min_{u \in BV} \lambda TV(u) + \int_{\Omega} \|Bu - f\|_{L_p}^p \, dx, \tag{2.39}$$

where $B \in \mathbb{R}^{d \times d}$ represents the effect of applying the PSF on the original signal. An example of the application of TV in image denoising is presented in Figure 2.2. The top row illustrates an example for an image affected by Gaussian blur and the recovered image using TV regularization, while the bottom row shows an example for an image affected by motion blur and the respective TV reconstruction.

TV fusion and zooming

Another interesting application of TV regularization is TV fusion. In this case it is assumed that multiple measurements of the same image/signal are available. In some applications it might be required to estimate an image with higher signal to noise ratio

(a) Original image (b) Image with Gaussian blur (c) TV reconstructed image

(d) Original image (e) Image with motion blur (f) TV reconstructed image

Figure 2.2: TV based image deblurring (images thaken from [37]).

with respect to the original ones, by fusing together information from all the measurements (TV fusion). In other applications it may be required to estimate the image at a higher spatial density with respect to the original ones (superresolution/TV zooming). The base model considered in these cases is the following

$$\min_{u \in BV} \lambda TV(u) + \sum_k \int_\Omega \|Bu - f_k\|_{L_p}^p \, dx. \tag{2.40}$$

An example of TV zooming is presented in Figure 2.3. In Figure 2.3a an image from a 5-frame sequence is shown expanded four times by nearest neighbour interpolation. In Figure 2.3b the same image expanded by bicubic interpolation is presented. Finally, Figure 2.3c shows the result obtained by applying TV zooming.

(a) Nearest neighbour interp. (b) Bicubic interp. (c) TV-zooming

Figure 2.3: Image superresolution (zooming) results (images taken from [38]).

2.3.2 Total Generalized Variation

In [30] the authors generalize the definition of TV by means of symmetric tensors of a given order l. Let $\mathrm{Sym}^j(X)$ denote the vector space of symmetric j tensors defined on space X, and div^j the j-divergence for tensor fields in $C_0^l(\bar{\Omega}, L_p, \mathrm{Sym}^l(X))$, then the Total Generalized Variation (TGV) functional is defined as

$$\mathrm{TGV}_\alpha^l(u) = \sup_{\substack{C_0^\infty(\Omega;\mathrm{Sym}^j(\mathbb{R}^d)) \\ \|\mathrm{div}^j(q)\|_\infty \leq \alpha_j, j=0,\dots,l-1}} \int_\Omega u\, \mathrm{div}^l(q)\, dx, \tag{2.41}$$

with $\alpha = (\alpha_1, \dots, \alpha_{l-1})$. For more details regarding this definition the reader is referred to [30].

The TGV functional is convex, which is a very important property as it allows to form convex energy functionals for which global optima can be efficiently estimated. TGV regularization of order l in practice favors piecewise polynomial signals of order up to $l-1$, hence TGV^2 favors piecewise affine functions, TGV^3 piecewise quadratic and so on.

In particular the previous functional can be seen as a combination of higher order TV terms, determined by the positive weights $\alpha \in \{\alpha_1, \dots, \alpha_l\}$, with l the maximum TV order. For further details regarding the relation of TGV with other higher order TV functionals see [28].

As shown in [30] by taking the Legendre-Fenchel transform of (2.41), an alternative definition can be given:

$$\mathrm{TGV}_\alpha^l(u) = \inf_{\substack{u_j \in C_c^{l-j}(\Omega,\mathrm{Sym}^j(\mathbb{R}^d)) \\ j=1,\dots,l-1,\, u_0=u,\, u_l=0}} \sum_{j=1}^{l} \alpha_{l-j} \int_\Omega |\mathcal{T}(u_{j-1}) - u_j|\, dx, \tag{2.42}$$

with $\mathcal{T} = (\nabla u + \nabla u^\top)/2$ the symmetrized gradient operator. From this definition the convexity of the TGV^l functional is more evident. This form also allows to express the TGV functional as

$$\mathrm{TGV}_\alpha^l = \inf_{v \in V} \sum_{j=1}^{l} \int_\Omega |M_j v|\, dx, \tag{2.43}$$

for $V = \{(u_0, \dots, u_{l-1}) \mid u_j \in C_c^{l-j}(\Omega, \mathrm{Sym}^j(\mathbb{R}^d))\}$, and M_j a suitable linear operator which depends on $\mathcal{T}$ and α. This formulation is suitable for using the PDHG algorithm as will be seen in Chapter 4.

2.3.3 First-order primal-dual algorithm

A widely used algorithm for minimizing the energy-like functionals discussed above is the PDHG algorithm [32, 39, 40]. In particular, all problems considered in this section can be expressed as

$$\min_u E(u), \quad \text{with} \quad E(u) \coloneqq J(Mu) + H(u, f). \tag{2.44}$$

Assuming that J is a close convex function, it satisfies the following relation with respect to its conjugate (see Note 2.2)

$$J(Mu) = \max_{q \in Q} \{\langle q, Mu \rangle - J^*(q)\}, \tag{2.45}$$

with $u \in U$ the primal variable and $q \in Q$ the dual variable. Using this expression in the original problem, we get the following primal-dual optimization problem

$$\min_{u \in U} \max_{q \in Q} E^\star(u, q), \tag{2.46a}$$

with

$$E^\star(u, q) := \langle Mu, q \rangle + H(u, f) - J^*(q). \tag{2.46b}$$

An important advantage of solving (2.46) instead of (2.44) is that the former no longer depends on the composition of J and M operators. This *splitting* simplifies the problem and allows for an efficient computation of the saddle-point. Due to this operator splitting, these methods are also called splitting methods.

In [20], a primal-dual algorithm for solving saddle-point problems of the form of (2.46) is proposed. The algorithm makes use of the resolvent operator which for a functional $F(x)$ is defined as

$$x = (Id + \tau \partial F)^{-1}(y) = \arg\min_x \left\{ \frac{\|x - y\|^2}{2\tau} + F(x) \right\}. \tag{2.47}$$

In order to be able to efficiently solve (2.46), it is required that the resolvent operators involved have a closed-form solution or can be computed efficiently. The algorithm is summarized below.

Algorithm 2.1 (PDHG)**.** Choose an initial estimate $u_0 = \bar{u}_0 \in U$. Provided that the iterations are well defined, for every $n \geq 0$ iterate

$$q_{n+1} \in (Id + \sigma \partial J^*)^{-1}(q_n + \sigma M \bar{u}_n) \tag{2.48a}$$
$$u_{n+1} \in (Id + \sigma \partial H)^{-1}(u_n - \tau M^* q_{n+1}) \tag{2.48b}$$
$$\bar{u}_{n+1} = u_{n+1} + \theta(u_{n+1} - u_n), \ \theta \in [0, 1]. \tag{2.48c}$$

The algorithm provable converges for J and H convex, and $\sigma\tau\|M\|^2 \leq 1$.

2.4 Lie groups and Lie algebras

Some of the problems encountered in this thesis require to perform optimization on manifolds. An important case are optimal registration problems. In this context, transformations in a N-dimensional projective space $\mathbb{P}^N$ are considered, which are usually represented with transformation matrices. The registration problem consists in finding the optimal transformation that relates two frames. Usually, the optimal transformations are estimated from data which contain noise. It is required then to refine these transformations under certain considerations.

In order to find an optimal transformation, a suitable parametrization of the transformation is needed. Considering the matrix entries as parameters provides poor results, since, on one hand, this leads to an over-parametrization of the problem, and more importantly the constraints on the matrix entries have to be explicitly considered and enforced. This complicates the optimization process and makes it less efficient. Instead, a minimal parametrization is preferred, where the number of parameters is equal to the effective degrees of freedom (dof) of the transformation and constraints are implicitly enforced.

Example 2.2. A rotation of a rigid object in 3D can be expressed by a 3×3 orthonormal rotation matrix R. Due to the orthonormality constraint it is possible to parametrize the rotation matrix by a (minimal) set of three parameters. Some of the most notable parametrizations of the rigid rotation are Euler angles, normalized quaternions and axis-angle representation. It is possible to obtain the rotation matrix R from any of these parametrizations.

Lie groups and Lie algebras provide a sound mathematical framework to map smooth manifolds, and hence also specific classes of matrices, to a minimal set of parameters. We review some of the most important definitions and properties here. For more details the reader is referred to [41]. A comprehensive tutorial is also presented in [42].

Definition 2.16 (Group)**.** A group is a set G, equipped with an operator $\bullet$ that satisfies the following properties:

$$\begin{array}{ll} a \bullet b \in G, \quad \forall a, b \in G; & \text{(Closure)} \\ (a \bullet b) \bullet c = a \bullet (b \bullet c), \quad \forall a, b, c \in G; & \text{(Associativity)} \\ \exists!\, Id \in G, \text{ such that } Id \bullet b = b \bullet Id = b, \quad \forall b \in G; & \text{(Identity Element)} \\ \forall a \in G,\ \exists b \in G \text{ such that } a \bullet b = b \bullet a = Id. & \text{(Inverse)} \end{array} \tag{2.49}$$

Definition 2.17 (Lie Group)**.** A Lie group is a set G which is a group and also a finite dimensional smooth manifold. In this case the group operations of multiplication and inversion are smooth maps.

Example 2.3. Real 3×3 invertible matrices form the group $GL(3)$ under standard matrix multiplication. Additionally, all projective transformation matrices form Lie groups under matrix multiplication. In particular, 3D rotation matrices belong to the special orthogonal Lie group $SO(3)$, 3D rigid body transformation matrices belong to the special Euclidean Lie group $SE(3)$, and the 3×3 homographic transform H belongs to the special linear group $SL(3)$. The term *special* is used to express the fact that the determinant of the matrix satisfies some constraint (typically equal to 1).

It is possible to define smooth paths on the manifold corresponding to the Lie Group as well as tangent vectors. If the path exists and lies on the manifold G, the set of all tangent vectors at a point $\mathbf{y}$ spans the tangent space of the manifold at $\mathbf{y}$.

2.4.1 Tangent space and Lie algebra

Definition 2.18 (Smooth path and tangent space)**.** A differentiable function $f : [a, b] \mapsto G$ represents a smooth path on the manifold G. Without loss of generality we assume $a = 0$ and $b = 1$, and denote $f(0) = \mathbf{y}$. Then, a tangent vector on the smooth path at $\mathbf{y}$ is given by

$$\mathbf{x} = \frac{\partial}{\partial t} f(t)\big|_{t=0}. \tag{2.50}$$

If such a smooth path exists, $\mathbf{x}$ is also a tangent vector of X.
The set of all tangent vectors at $\mathbf{y}$ span the tangent space of G at $\mathbf{y}$. In particular, the set of all tangent vectors at the identity of a Lie group G span a vector space $\mathfrak{g}$, called the *vector space at the identity*.

Example 2.4. We examine the tangent space of the Lie group $SO(3)$. We consider a parametrization based on the zyx fixed-axes Euler angles, hence $R(t) = R_x(t)R_y(t)R_z(t)$. We consider first a path $f_x(t) \coloneqq R_x(t)$ and compute the corresponding tangent vector

$$\mathbf{G}_1 \coloneqq \frac{\partial}{\partial t} R_x(t)\big|_{t=0} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\sin(0) & -\cos(0) \\ 0 & \cos(0) & -\sin(0) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}. \tag{2.51}$$

Similarly we find

$$\mathbf{G}_2 \coloneqq \frac{\partial}{\partial t} R_y(t)\big|_{t=0} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad \mathbf{G}_3 \coloneqq \frac{\partial}{\partial t} R_z(t)\big|_{t=0} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{2.52}$$

We notice also that $R_x(0) = R_y(0) = R_z(0) = I_{3\times 3}$. Hence, $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$ span the tangent space to the Lie Group $SO(3)$ at the identity, and for this reason they are called also *generators* of the tangent space at the identity. Finally, we note that the tangent space at the identity of $SO(3)$, denoted $\mathfrak{so}(3)$, is equivalent to the space of skew-symmetric matrices.

Remark 2.2. *It is possible to arrive at the previous result without considering a specific parametrization, but using the orthogonality of the $SO(3)$ group instead. More specifically, differentiating the expression $R(t)R(t)^\top = I$ at the origin $t = 0$, one gets*

$$\frac{\partial}{\partial t} R(t)\big|_{t=0} R(0)^\top + R(0) \left(\frac{\partial}{\partial t} R(t)\big|_{t=0} \right)^\top = 0, \tag{2.53}$$

and since $R(0) = I$, the following holds

$$\frac{\partial}{\partial t} R(t)\big|_{t=0} = -\left(\frac{\partial}{\partial t} R(t)\big|_{t=0} \right)^\top. \tag{2.54}$$

This agrees with the result obtained in Example 2.4, showing that $\frac{\partial}{\partial t} R(t)\big|_{t=0}$ span the space of skew-matrices, for which $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$ form a basis.

Definition 2.19 (Hat and vee operators)**.** We define the *hat* operator $\hat{\cdot}_{\mathfrak{g}} : \mathbb{R}^N \mapsto \mathfrak{g}$ as

$$\hat{\mathbf{x}}_{\mathfrak{g}} = \sum_{i=1}^{N} x_i \mathbf{G}_i, \tag{2.55}$$

with N the dimensionality of the tangent space.
We also define the *vee* operator $(\cdot)^\vee : \mathfrak{g} \mapsto \mathbb{R}^N$ as the inverse of the hat operator, namely

$$(\hat{\mathbf{x}})^\vee_{\mathfrak{g}} = \mathbf{x}. \tag{2.56}$$

Example 2.5. In case of $\mathfrak{so}(3)$, for $\mathbf{x} \in \mathbb{R}^3$ we have

$$\hat{\mathbf{x}}_{\mathfrak{so}(3)} = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} = [\mathbf{x}]_\times \in \mathfrak{so}(3), \tag{2.57}$$

where the $[\mathbf{x}]_\times$ notation is based on the fact that

$$[\mathbf{x}]_\times \mathbf{v} = \mathbf{x} \times \mathbf{v}, \quad \mathbf{x}, \mathbf{v} \in \mathbb{R}^3. \tag{2.58}$$

Conversely, for $X \in \mathfrak{so}(3)$, we have

$$(X)^\vee_{\mathfrak{so}(3)} = \frac{1}{2} \begin{pmatrix} X_{3,2} - X_{2,3} \\ X_{1,3} - X_{3,1} \\ X_{2,1} - X_{1,2} \end{pmatrix} \in \mathbb{R}^3. \tag{2.59}$$

Definition 2.20 (Lie bracket)**.** The Lie bracket for a matrix group G and its corresponding tangent space $\mathfrak{g}$ is defined as

$$[U, V] := UV - VU \tag{2.60}$$

Moreover, the tangent space $\mathfrak{g}$ is closed under the Lie bracket

$$\text{for } U, V \in \mathfrak{g} \quad [U, V] \in \mathfrak{g}. \tag{2.61}$$

Definition 2.21 (Lie algebra)**.** Lie algebra is the algebraic structure obtained by associating the Lie bracket with the corresponding tangent space $\mathfrak{g}$.

2.4.2 Exponential and logarithmic map

We now discuss the role of exponential and logarithmic map.

Definition 2.22. The exponential functions for square matrices $\exp : \mathbb{R}^{N\times N} \mapsto \mathbb{R}^{N\times N}$ is defined as

$$\exp(X) = \sum_{k=1}^{\infty} \frac{X^k}{k!}, \quad \text{with: } X^0 = I. \tag{2.62}$$

The matrix exponential function shares similar properties with the standard one as seen below.

Proposition 2.7 (Exponential mapping calculus)**.** *If matrices commute, namely $XY = YX$, then the following holds*

$$\exp(X)\exp(Y) = \exp(X + Y). \tag{2.63}$$

Additionally, we have

$$\frac{\partial}{\partial X} \exp(tX) = \exp(tX)X. \tag{2.64}$$

The proof of this proposition follows closely the respective proofs for the standard exponential function.

Proposition 2.8. *The range of the exponential map is the group $GL(N)$.*

Proof. We use Proposition 2.7. Since X and $-X$ always commute we have

$$\exp(X)\exp(-X) = \exp(X - X) = \exp(0) = I. \tag{2.65}$$

Hence $\exp(X)$ is an invertible matrix with $\exp(X)^{-1} = \exp(-X)$. □

The previous results shows that the exponential map maps the set of square matrices, which belong to the tangent space $\mathfrak{gl}(N)$, to the set of invertible matrices $GL(N)$. It can be shown (see [41]) that for $X \in \mathfrak{g}(N) \subseteq \mathfrak{gl}(N)$ then $\exp(X) \in G$, where $\mathfrak{g}$ is the tangent space of the Lie group G.

If for the Lie group G the exponential map is surjective then we can define the inverse of the mapping, called *logarithmic map*.

Definition 2.23 (Logarithmic map)**.** The logarithmic map $\log : \mathbb{R}^{N\times N} \mapsto \mathbb{R}^{N\times N}$ is defined as

$$\log(\exp(X)) = X. \tag{2.66}$$

In many cases the exponential map is not injective, nevertheless the definition of the logarithmic map is usually considered also in these cases by a suitable restriction of the operator's range space. It can be shown that the for $X \in G(N) \subseteq GL(N)$, then $\log(X) \in \mathfrak{g}$.

2.5 Nonparametric Bayesian models

In many computer vision applications basic properties of the imaged objects need to be modeled. For example regarding the problem of intrinsic images, discussed in detail in Chapter 7, the shape, the albedo and the shading of a surface are modeled based on object appearance under different light conditions. Statistical inverse problem theory can be used to estimate these properties in a principled way. More specifically, in this thesis non-parametric Bayesian models are considered, which provide a powerful learning and inference framework. In this section we review some key properties and definitions regarding non-parametric Bayesian models in general, and we discuss Dirichlet Processes in detail. For additional details regarding non-parametric Bayes models we refer to [43] and [44]. A useful tutorial about Dirichlet Processes and Dirichlet Process Mixture models, which we closely follow, is provided in [45].

In this context, considering a sample space $\mathbf{X}$, a *statistical model* is a set of probability measures $M \subset PM(\mathbf{X})$, with $PM(\mathbf{X})$ the space of all probability measures on $\mathbf{X}$. Intuitively, a statistical model is used to capture the form of an underlying probability distribution from a set of observed data sampled from this distribution. Statistical models are defined based on a parameter ϕ which takes values over a parametric space Φ.

A parametric model is a model based on a parameter ϕ of finite dimensions. Consider as an example the normal distribution $\mathcal{N}(\mu, \sigma)$. A model based on normal distribution has a two dimensional parameter space, spanned by μ and σ. Contrary, considering a non-parametric model, the dimensionality of its parametric space is not fixed, but depends on the number of the observations instead. Hence the term non-parametric is used in the sense that the number of parameters, or equivalently the dimensionality of its parametric space, is not fixed a-priori. Formally, this is modeled by considering a parametric space of infinite dimensions.

Bayesian non-parametric models, are non-parametric models for which the parameter is considered a random variable X with values on the parametric space Φ. This is based on the fact that the value of the parameter is unknown, and according to Bayesian statistics any uncertainty is modeled as randomness. To model this uncertainty we consider that the parameter X is distributed according to some specific distribution Q, called

the *prior distribution*. Combining the previous observations, a Bayesian non-paramatric model is defined as

$$X \sim Q \tag{2.67}$$
$$Y_1, Y_2, \ldots | X \sim_{iid} P_X \tag{2.68}$$

The symbol $\sim_{iid}$ means that the data are *conditionally* identical and independently distributed (i.i.d.) according to P_X. The objective of Bayesian nonparametric modeling is then to determine the posterior distribution

$$Q(X|Y_1, \ldots, Y_N). \tag{2.69}$$

Conversely to classical statistical approaches, the values of the parameters remain uncertain given the (finite) observations, and the posterior distribution (2.69) captures this uncertainty.

In the following sections we discuss Dirichlet Processes (DP) which belong to the family of non-parametric Bayesian models.

2.5.1 Dirichlet Processes

To motivate the use of Dirichlet Processes let us consider the problem of clustering. In particular given a set of data samples we make the assumption that the samples form groups. The clustering problem deals with finding the optimal grouping of the samples. Clustering problems can be modeled using mixture models which in their general from are defined as

$$p(x) = \sum_{k \in \mathbb{N}} \pi_k p(x|\theta_k). \tag{2.70}$$

If the number of groups is defined a-priori, then classical statistical approaches can be used to solve the assignment problem, namely in which group each sample corresponds.

Example 2.6. Gaussian Mixture Model (GMM) is a popular mixture model. According to this model, the data can be described by a mixture of Gaussian density functions. The overall distribution is defined as

$$M(\mathbf{x}) = \sum_{k=1}^{N} \pi_k \mathcal{N}_k(\mathbf{x}|\mu_k, \Sigma_k), \text{ with } \sum_{k=1}^{N} \pi_k = 1. \tag{2.71}$$

The assignment problem can then be solved using the Expectation Maximization algorithm, see [14] for more details.

In practice, many times the number of groups is not known a priori. More importantly, even if there exists some natural grouping of the data, it may not agree with the underlying assumption made about the distribution of the data in each cluster. Dirichlet processes provide a Bayesian framework which allows to infer from the data the number of components. This is achieved by considering an infinite dimensional parameter space.

Dirichlet processes are distributions over probability measures defined as follows [46].

Definition 2.24. Let H be a distribution over Θ and $\alpha > 0$. Then for any finite measurable partition $A_1, \ldots, A_r$ of Θ, we say that G is distributed according to a Dirichlet process if its marginal distributions are Dirichlet distributed, namely

$$(G(A_1), \ldots, G(A_r)) \sim \mathrm{Dir}(\alpha H(A_1), \ldots, \alpha H(A_r)) \tag{2.72}$$

The expected value of the DP is $\mathbb{E}[G(A)] = H(A)$ and its variance $\mathrm{Var}[G(A)] = H(A)(1 - H(A))/(\alpha + 1)$. This allows for an interpretation of the H distribution as the mean of DP and the concentration parameter α as its inverse variance.

An alternative (constructive) definition is provided by [47].

Definition 2.25. Let $\alpha > 0$ and H a probability measure on Θ, and let

$$\beta_1, \beta_2, \ldots \sim_{iid} \mathrm{Beta}(1, \alpha) \text{ and } \pi_k \coloneqq \beta_k \prod_{l=1}^{n-1} (1 - \beta_l) \tag{2.73a}$$

$$\theta_k^* \sim_{iid} H, \tag{2.73b}$$

The random probability measure $G \coloneqq \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*}$, with $\delta_{\theta_k^*}$ point masses centered at θ_k^*, is generated according to a Dirichlet process with base measure H and concentration α, namely $G \sim \mathrm{DP}(\alpha, H)$.

The construction of π_k is called *stick-breaking* construction as it corresponds to the following metaphor. Considering a stick of initial length 1, we break it at β_1, and hence π_1 is the length of the first stick. Recursively breaking the remaining portion we obtain π_2, π_3 and so on. The stick-breaking construction provides an intuitive definition of DPs and has been also used to introduce various extensions and inference techniques for DPs.

This definition justifies why DPs are well defined probability distributions. In general, for an infinite number of partitions, normalization of the i.i.d. variables fails as their infinite sum is equal to $+\infty$ almost surely. Stick-breaking construction of π_k, though, makes clear that $G \coloneqq \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$ has a finite sum, and since $G \sim \mathrm{DP}(\alpha, H)$, DP is a valid probability distribution. Moreover, the stick-breaking construction highlights the clustering properties of the distributions drawn from DP.

Posterior distribution

We now examine the posterior distribution of a DP. Since G, sampled from $DP(\alpha, H)$, is also a (random) distribution, it is possible to draw samples from it. We denote $\theta_1, \ldots, \theta_n$ a sequence of independently drawn samples from G, with each sample θ_i taking values on Θ. In various cases and in particular for the clustering problem, we are interested in the posterior distribution of G given the observations $\theta_1, \ldots, \theta_n$. Let $A_1, \ldots, A_r$ be a finite measurable partition of Θ, and let n_k be the number of θ_i observed in partition A_k. Using the fact that G are Dirichlet distributions, and using the conjugacy between the Multinomial and the Dirichlet distribution the following holds

$$(G(A_1), \ldots, G(A_r)) | \theta_1, \ldots, \theta_n \sim \mathrm{Dir}(\alpha H(A_1) + n_1, \ldots, \alpha H(A_r) + n_r). \tag{2.74}$$

This leads to the following values for the updated distribution

$$\alpha' = \alpha + n, \tag{2.75a}$$

$$H' = \frac{\alpha H + \sum_{i=1}^{n} \delta_{\theta_i}}{\alpha + n}. \tag{2.75b}$$

Hence, the posterior can be expressed as

$$G|\theta_1, \ldots, \theta_n \sim \mathrm{DP}\left(\alpha + n, \frac{\alpha}{\alpha + n} H + \frac{n}{\alpha + n} \frac{\sum_{i=1}^{n} \delta_{\theta_i}}{n}\right). \tag{2.76}$$

From (2.76) we observe that the posterior of G given the observations is also distributed according to a DP distribution, hence the DP prior is closed under posterior updates given observations. We also observe that the posterior base distribution corresponds to a weighted average of the prior base distribution H and an empirical distribution given by $\sum_{i=1}^{n} \delta_{\theta_i}/n$.

Predictive distribution

Let us now derive the predictive distribution for θ_{n+1}. From the previous result follows that $\theta_{n+1}|G, \theta_1, \ldots, \theta_n \sim G$. For a measurable partition A then we have

$$P(\theta_{n+1} \in A|\theta_1, \ldots, \theta_n) = \mathbb{E}[G(A)|\theta_1, \ldots, \theta_n], \tag{2.77}$$

and by using the posterior distribution of G given the observations we obtain

$$P(\theta_{n+1} \in A|\theta_1, \ldots, \theta_n) = \frac{1}{\alpha + n}\left(\alpha H(A) + \sum_{i=1}^{n} \delta_{\theta_i}(A)\right). \tag{2.78}$$

Marginalizing out G we get

$$\theta_{n+1}|\theta_1, \ldots, \theta_n \sim \frac{1}{\alpha + n}\left(\alpha H + \sum_{i=1}^{n} \delta_{\theta_i}\right), \tag{2.79}$$

which is equivalent to the posterior distribution of G given the observations.

An important observation is that the predictive distributions has point masses located on the previous observations, hence with positive probability the new samples will take values equal to previous observations. This shows that even for a smooth base distribution H, the distributions sampled from $\mathrm{DP}(\alpha, H)$ are discrete. Additionally, this shows that samples of a DP tend to be clustered together. In particular, if we consider the unique values among θ_i, denoted as θ_k^*, we obtain

$$\theta_{n+1}|\theta_1, \ldots, \theta_n \sim \frac{1}{\alpha + n}\left(\alpha H + \sum_{i=k}^{m} n_k \delta_{\theta_k^*}\right), \tag{2.80}$$

which shows that draws from a DP are composed by weighted sums of point masses. This agrees with the stick-breaking definition we provided above (Definition 2.25).

Dirichlet Process Mixture Models

Returning to the problem of data clustering, mixture models based on Dirichlet processes have been proven very powerful. The main advantage of using Dirichlet process is based on the fact that they can model a potentially infinite number of components. In

this context, the samples $\theta_1, \ldots, \theta_n$ of the distribution G are considered as latent variables, which define the number and the location of the clusters in Θ. The observed data x_i are then considered to be sampled from a distribution $F(\theta_i)$, which depends on the value of θ_i and corresponds to the distribution of the data in the i-th cluster. Dirichlet Process Mixture models (DPM) are defined as follows.

Definition 2.26. Let z_i be a cluster assignment variable which assigns the data x_i to the k-th cluster with probability π_k. Dirichlet Process Mixture model is defined as

$$\beta_1, \beta_2, \ldots | \alpha \sim_{iid} \mathrm{Beta}(1, \alpha) \text{ and } \pi_k \coloneqq \beta_k \prod_{l=1}^{n-1} (1 - \beta_l) \tag{2.81a}$$

$$\theta_k^* | H \sim H, \tag{2.81b}$$

$$z_i | \pi \sim \mathrm{Mult}(\pi) \tag{2.81c}$$

$$x_i | z_i, \{\theta_k^*\} \sim F(\theta_{z_i}^*) \tag{2.81d}$$

with $G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*}$.

We note that although DPM are infinite mixture models, due to the fact that π_k decrease exponentially, only a small number of clusters will be used to model the data. Nevertheless, the number does not need to be defined a-priori like in finite mixture models. Standard Bayesian inference techniques can be applied on DPMs, as the ones discussed in the previous section (e.g. Variational Bayes and MCMC approaches). In [48] a survey of MCMC inference techniques applied for DPM is provided.

Chapter 3

Saliency Prediction

In the coherence theory of attention, introduced by Rensink, O'Regan, and Clark, a coherence field is defined by a hierarchy of structures supporting the activities taking place across the different stages of visual attention. At the interface between low level and mid-level attention processing stages are the proto-objects; these are generated in parallel and collect features of the scene at specific location and time. These structures fade away if the region is no further attended. In this Chapter, we discuss a method to computationally model these structures. Our model is based experimentally on data collected in dynamic 3D environments via the Gaze Machine, a gaze measurement framework. This framework allows to record pupil motion at the required speed and projects the point of regard in the 3D space. To generate proto-objects the model is extended to vibrating circular membranes whose initial displacement is generated by the features that have been selected by classification. The energy of the vibrating membranes is used to predict saliency in visual search tasks.

3.1 Introduction

Saliency prediction in visual search requires to understand which features of the scene are processed and how, and in which way this processing delivers a structure that is overtaken by attention, which then induces focusing on a selected region of the scene.

In artificial systems this is a crucial concept. There are two main reasons for that. On the one hand the complexity of searching the visual field is too high to be managed by processing the whole visual input at the resolution of the fovea, as indicated by [49]. On the other hand feature detectors and orientation filters handle pre-attentive processing by partially discarding the visual input, but they cannot handle the further integration processing required to lift up the low-level structures to focused attention.

We should note that artificial systems suffer of several limitations due to the mechanic, electronic and software components. Yet artificial systems need to learn to predict saliency to find targets in crowded scenes, without overloading teir resources. This is a necessary step in the design of efficient cognitive systems, to avoid memory or reasoning being clogged and paralyzed by the huge amount of visual information acquired at possibly high frame rate. A tacit assumption is that artificial computational models rely on psychophysical, neurophysiological and psychological studies (PNP) on pre-attentional and attentional processing, and then add further constraints to these models to cope with the above mentioned limitations.

This is the line of research mainly taken so far, though following two main directions, namely predicting saccade directions and predicting saliency from the features standpoint. Predicting saccades directions has been analyzed in [50], [49], [51], [52], [53]. Predictions of saccade targets with a number of features, via bottom-up models, have been tested in [54].

In general, approaches have exploited the simulation of saccades either by active cameras, as in [55], [56], or via biologically founded prior models of saliency as in [57], [58], [59], [60], [61], [62], to cite some of the works from the wide literature on saliency prediction.

In this work we focus on the steps between features analysis and collection, and their integration into a coherent structure that is then passed to attention, basing our approach purely on collected data and the concept of proto-object developped within the coherence theory of attention by [17].

Indeed, since the fundamental work of [63] on feature integration, it has become clear that in the pre-attentive, early vision phase, primitive visual features can be rapidly accessed in searching tasks. For example, colors, motion, and orientation can be processed in parallel and effortlessly, and the underlying operations occur within hundreds of milliseconds. So the pre-attentive level of vision is based on a small set of primitive visual features organized in maps, that are extracted in parallel while the attentive phase serves to group these features into coherent descriptions of the surrounding scene. When attention takes control, processing passes from parallel to serial.

Since Treisman's feature integration theory, several models have been further provided in the literature for feature integration. Among those that have led to a concept of representation we consider [64] who have observed that there is a large differentiation *in search difficulty, observed across different stimulus material*. On this basis Duncan introduces the theory of visual selection as distinguished into three stages: the *parallel one*, that produces an internal structured representation, a *selective one* matching the internal representation, and the *transduction one* providing the input of selected information to the visual short term memory. This theory relies on the evidence of low efficiency of basic features parallel processing, in the presence of heterogeneous distractors. On the basis of this observation Duncan introduces the concept of *structural unit* as an internal representation given to the visual input (close to 3-D model of [65]). Further, [66] has shouldered the concept of structural units, by noting that visual search might need grouping and categorization. Indeed, [67] suggest that categorization is a strategy that is invoked when it is useful and that it could affect different features of the visual input. [68] makes clear that attentional deployment is guided by the output of earlier parallel processes, but its control can be exogenous, *based on the properties of the visual stimulus*, or endogenous, based on the subject task, and he introduces the notion of feature maps (see also [69]) as independent parallel representations for a set of basic limited visual features. Finally, activation maps, both bottom-up and top-down, serve in [68] model to guide attention toward distinctive items in the field of view. In summary, Wolfe suggests that information extracted in parallel, with loss of details, serves to create a representation for the purpose of guiding attention.

The huge amount of literature that has studied how, from parallel processing, across large areas of the visual field, focused attention emerges (see also [70] and [71]) has led to the quest for a virtual representation that could explain the way input is discarded and selected features are integrated in a coherent representation.

According to these principles, in this work we propose a methodology, suitable

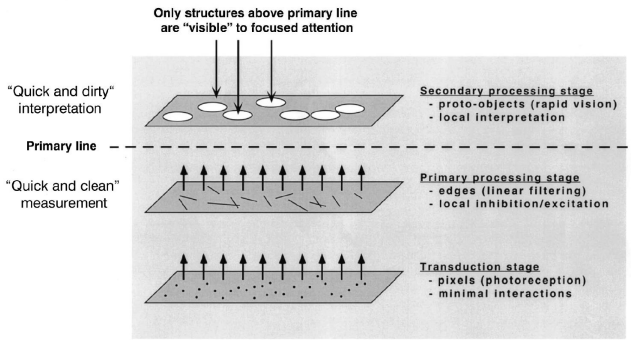


Figure 3.1: The image above, taken from [17], illustrates Rensink low-level vision architecture whose output are proto-objects *that become the operands for attentional objects*.

for computational artificial-attention, to study saliency for visual search, in dynamic complex scenes, motivated by the concept of virtual representation developed in the coherence theory of attention of [17] [72], [73]. Rensink introduces the concept of *proto-object* as a volatile support for focused attention, which is actually needed to see changes, see [74]. [17] assumes that proto-objects are formed in parallel across the visual field and form a continuously renovating flux that is accessed by focused attention. Proto-objects are collected by focused attention to form a stable object temporally and spatially coherent, which provides a structure for perceiving changes.

In Figure 3.1 Rensink's triadic architecture is illustrated. In this architecture the lower level corresponds to the retinotopic mapping and, going up, proto-objects are structures for more complex feature configurations formed in parallel across the visual field and lying at the interface between low-level vision and higher attentional operations. These structures are said to be volatile, and fading away as new stimuli occur, within "few hundreds of milliseconds", as detailed in [72]. Focused attention, in Rensink's triadic architecture, accesses some of the generated proto-objects to stabilize them and form individual objects "with both temporal and spatial coherence", [17]. Proto-objects are linked within a coherence field to the *nexus*, a structure coarsely summarizing the properties of the stabilized ones. Proto-objects have been explored in computational attention for modeling how object recognition can use their representation and generation, thus at the high-level interface, in [75], and in [76]. Here, instead, we are interested in the other side of the interface, namely we model their generation and study their spatial and temporal persistence across the visual fields in visual search tasks. Note that we take into account real dynamic environments. Furthermore we show that these structures can be used to learn the parameters of the underlying process and predict saliency distribution across the scene.

The chapter is organized around the problem of modeling the data acquisition, for a freely moving subject, the recovery of the point of regard in the scene and the proto-object generation, as follows. In the next section we illustrate how to obtain the scanpath of a subject searching for some objects in the scene. Namely how to obtain the position of the head and the direction of the gaze in the scene, using a wearable device, the Gaze Machine (GM). In the next section, we illustrate how features are learned from the data acquired by the GM, specifically for a set of search tasks. Then, in Section 3.4, we introduce a model for the generation of proto-objects based on vibrating membranes to account for their volatility, according to the learned features. Finally we provide some experimental validation.

Figure 3.2: The Gaze Machine (GM) worn by the subject collecting PORs in an outdoor search task.

3.2 Acquisition model for search strategy estimation

To model saliency prediction, computational studies have quite limited resources available, as data acquisition is based on uncertain measurements and ground truth is available only if experiments are rather constrained. The realization of a wearable device that allows to register the Point of Regard of a subject in an unconstrained condition has made possible to collect a great amount of data, see Figure 3.2.

We aim at exploiting these data for modeling the features that are selected during a search task, whether these specify general properties that are preserved across tasks or local properties closely related to the target. These properties characterize the spatial and temporal relations inducing the stimulus to be triggered. As highlighted in [77] the V4 area displays neural activity with features similar to the target, and this is the area involved in the formation of a coherence field, according to the coherence theory of attention. Indeed, the interaction between stimuli-driven and voluntary factors becomes further and further relevant in the later stages of attentional processing, where more complex coherent fields of features configurations are formed. From the stand point of computational attention a *proto-object* can be described as a *configuration of features having relative time and spatial coherence, directly affected by attention, and generating a motion field pulling the gaze toward the target*.

Proto-objects in this sense are dynamic and relatively volatile feature structures related both to fast eye movements, namely saccades, and to saliency. These feature structures are precursors of attention and further used by attention to drive recognition – this is the double face of proto-objects between pre-attentive and selective attention, as highlighted in [64] and [17] – and can be localized in time and space: proto-objects may last few milliseconds up to hundreds of milliseconds.

We recall that the POR, namely the Point of Regard, is *the point on the retina at which the rays coming from an object regarded directly are focused*. In particular, we assume that PORs are the point on the fovea, sub-tending a visual angle of about 1.7 degrees.

Saccades are fast eye movements that can reach peak velocities of $1000°/s$. While a subject is moving, like in our framework, saccades do not exceed $30°$, but the velocity follows an exponential function. According to [78], the range in the duration of $30°$ saccades can be up to $100\ ms$. Saccades models rarely explain the role of saliency, being mainly motivated by the need to model the motion control (see [79], [78], [80], and for a review see [81] and the references therein). It follows that saccade models

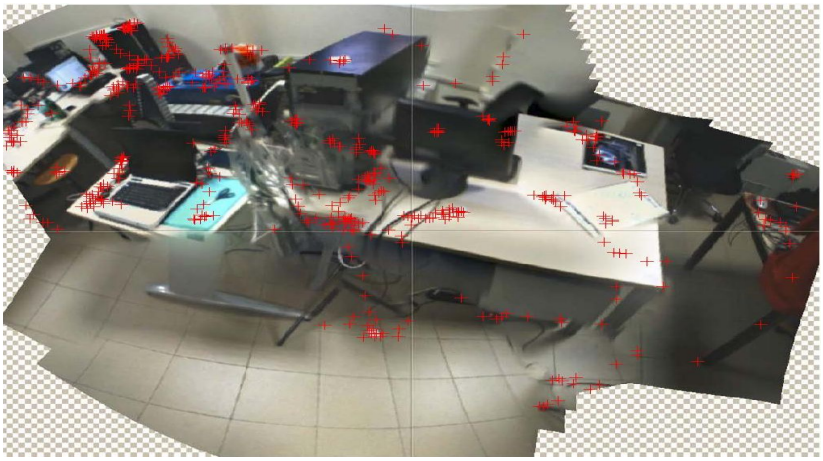


Figure 3.3: A panoramic stitching showing the PORs collected in $20s$; the stitching has been realized with 30 images over a collection of 600 images of the scene. The acquisition of the scene is at 30 Hz while the acquisition of the eye is at 120 Hz. The PORs are measured on the scene via dense structure from motion and further reprojected on the retinal plane (image plane).

do not contribute to the interpretation of proto-objects, although saccades direction and speed are substantial to explain the motion field a proto-object generates and how it fades away.

Similarly, saliency models not grounded in the 3D visual scene fail to explain the coherence of proto-objects, their motion field, hence their dynamics. To measure the volatility of proto-objects we rely on two models: a model of the scan path, and a model of the surface response to the POR. To obtain meaningful data from which parameters can be estimated, we use an acquisition device, the Gaze Machine specified in [16], here denoted GM. In particular we present below a novel method to recover the scan path of the the head and eyes of a subject wearing the device.

Scan path estimation. The formal model for scene acquisition, PORs projections into the retinal plane (image plane) and their registration into the scene structure, while the subject explores the environment, is the Gaze Machine (GM) model, described in [82] and [16]. Here we are mainly concerned with the scan path of the head; namely of the subject's head, while she/he is moving across the environment to perform a search task. The task implies possible return to previously focused regions, in so inducing relations among the PORs at different time periods. In other words the scan path model has to establish whether a set of PORs belongs to the same saliency region, according to the process deployed during search. Some results of scanpath estimation, namely of the projection of the gaze on the visual field, are illustrated in Figure 3.3.

First note that the GM enables good controlled experiments, as the device can be well fitted on the head, the pupil rate acquisition can reach 180 Hz, ensuring to get good saccades approximation, while the visual field can be acquired at a rate up to 30 Hz, the association with the much faster acquisition of gaze is maintained by time-stamping. The GM calibrated stereo rig records the experimental stimuli, allowing for dense 3D reconstruction from multiple views. Moreover, the localization of the subject in the 3D experimental scenario is based on the visual data acquired by the GM scene cameras.

The above statement assesses that the model we propose is quite general and allows a calibration procedure that is efficient and easy to perform *on field*, with little intervention

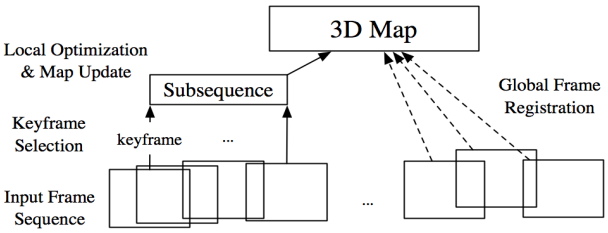


Figure 3.4: Visual Localization of the subject. Local consistency is enforced by optimization on frame subsequences limited by keyframes. Frame registration with the 3D map ensures global consistency.

from the subject. After the calibration, the parameters for the model of eye positions are recovered and the gaze direction $\hat{\rho}(t)$ is computed, on the basis of the imaged pupil at time t, and the geometry of the multi-camera system. The estimated POR is relative to the acquisition device and a localization step is needed in order to measure gaze behaviors in the 3D world taking into account the changes in the pose of the subject's head.

To build a map of gazed 3D points requires the following steps:

1. estimating the 3D POR π^c in the reference frame of the GM left scene camera.

2. estimating the 3D pose (6 degrees of freedom) of the GM left scene camera in the reference frame of the experiment at hand, in terms of translation $\mathbf{t}$ and orientation $\mathbf{R}$;

3. computing the 3D POR in the world reference frame as $\pi^w = \mathbf{R}\pi^c + \mathbf{t}$.

Note that the 3D PORs are naturally attached to 3D points that are imaged in the retinal plane, and the 3D points generate the 3D global map. For an abstract structure of the hierarchical construction see Figure 3.4.

Subject localization Most of the issues affecting the localization of a camera system, see [83, 84], also apply to the GM, with some notable differences. Indeed, the main concern of the GM localization is high precision in the estimation of the whole trajectory, needed to correctly estimate the 3D POR, see Figure 3.5, to see the head poses of a subject performing a search task.

We follow an efficient hierarchical approach subdividing the whole trajectory into sets of frames, that we specify as *coherent subsequences*. Indeed, subsequences are characterized by a high level of coherence in terms of what the subject is attending in the course of the experiment. More specifically, the pose estimation is performed sequentially, adding a new frame to the last acquired set, denoted *subpath*, as long as the estimation is sufficiently accurate, performing sparse bundle adjustment to enforce consistency and to avoid drifting, see [85, 83].

Subsequences are induced by the selection of a *keyframe* to delimit the coherence of head poses. Namely, the set of keyframes constitutes a subset of the whole frame sequence and a new keyframe, eliciting a new subsequence, is created upon the event of a change in the visual scene.

The sequence of images collected by the GM scene cameras is used to localize the subject in the experimental environment. The estimation of the subject's pose relies on

matching descriptors from visual features corresponding to the current view with those recorded in the map built so far. The overall process is summarized as follows:

1. Take the first frame of the sequence as the first keyframe. A map of 3D feature points is initialized by triangulating matched image features in the first pair of stereo frames.

2. For each new pair of stereo frames, compute matched feature points and descriptors among left and right views; triangulate to get a new set of unoptimized 3D points. Match the computed descriptors with the current map. Estimate the pose w.r.t. the current map and compute the POR in 3D. Check if a new keyframe has to be selected, if not repeat 2.

3. Upon the selection of a new keyframe, add the current frame to the keyframe list. Optimize by a local bundle adjustment w.r.t. unoptimized 3D points and cameras from the subsequence. Add the optimized points to the map and empty the set of unoptimized points.

Let us call $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{X}}_i)$, $i = 1, \ldots N$, the N pairs of matched retinal plane and map points, $\tilde{\mathbf{x}}_i \in \mathbb{R}^2$ and $\tilde{\mathbf{X}}_i \in \mathbb{R}^3$ respectively. The pairs $(\mathbf{x}_i, \mathbf{X}_i)$ represent the same points in homogeneous coordinates: $\mathbf{x}_i \in \mathbb{R}^3$ and $\mathbf{X}_i \in \mathbb{R}^4$. The goal is to compute the pose, expressed by the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$, of the camera that is projecting the 3D points $\mathbf{X}_i$ into the retinal points $\mathbf{x}_i$. We refer in general to cameras specified by a translation $\mathbf{t}$, a rotation $\mathbf{R}$ and a calibration matrix $\mathbf{K}$ as $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$. The rotation, translation and calibration might be decorated by superscripts specifying whether they involve the left (l), the right (r), or the scene (s) cameras. According to [83], let us define the matrix $\mathbf{K}$ expressing the intrinsic camera parameters, namely the focal lengths f_x and f_y and the position of the principal point in image coordinates (p_x, p_y), as

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix} . \tag{3.1}$$

Fiore's linear algorithm for exterior orientation [86] has been used to generate multiple hypotheses in a RANSAC-based, robust estimation process ([87]). The core routine estimates the camera pose by solving

$$Z_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = s\mathbf{K}\mathbf{R}(\tilde{\mathbf{X}}_i + \mathbf{t}) \qquad i = 1, \ldots, N. \tag{3.2}$$

Here Z_i, $i = 1, \ldots, N$ are the depth parameters and s is the scale parameter. Note that these last parameters can be recovered up to an arbitrary common scale factor, and that the calibration matrices (likewise those of the eye cameras) are pre-estimated. The algorithm first estimates Z_i in order to subsequently solve the problem of absolute orientation with scale. The model selection process makes use of an error function that takes into account re-projection errors in both the left and right retinal planes of the stereo pair. Using the l and r superscripts to identify quantities related to the left and right scene cameras, respectively, and assuming the relative pose $\mathbf{R}^s$ and $\mathbf{t}^s$ of the scene cameras fixed to the GM stereo rig known from calibration, the error function is:

$$\epsilon_i = d\left(s\mathbf{K}^l\mathbf{R}(\tilde{\mathbf{X}}_i + \mathbf{t}), \mathbf{x}_i^l\right)^2 + d\left(s\mathbf{K}^r\mathbf{R}^s[\mathbf{R}(\tilde{\mathbf{X}}_i + \mathbf{t}) - \mathbf{t}^s], \mathbf{x}_i^r\right)^2, \tag{3.3}$$

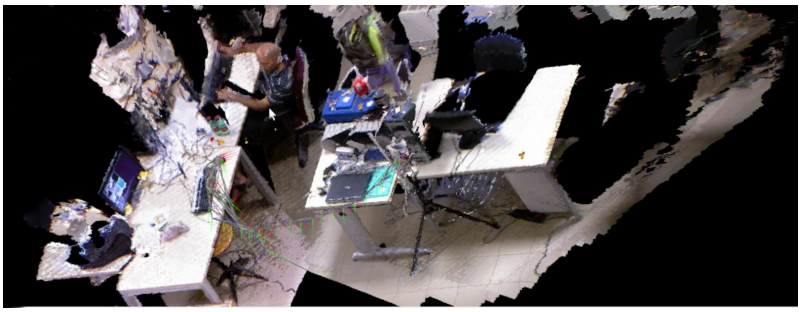


Figure 3.5: The figure illustrates the reconstruction of the scene where the subject is performing the experiment *searching for J*, wearing the GM. The head poses, which are projected on the scene, are computed with the described localization algorithm.

where d is the Euclidean distance and $\mathbf{K}^l, \mathbf{K}^r$ are the calibration matrices of the left and right scene cameras, see [16]. The two distance terms in equation (3.3) account for reprojection errors in the left and right scene camera planes. The largest consensus set is selected by RANSAC according to equation (3.3) and used to estimate a model. A final Levenberg-Marquardt optimization is carried out to refine the linearly estimated pose by iteratively minimizing ϵ_i with respect to $\mathbf{R}$ and $\mathbf{t}$:

$$\mathbf{R}, \mathbf{t} = \arg\min_{\mathbf{R},\mathbf{t}} \sum_i \epsilon_i. \tag{3.4}$$

Details of the suggested minimization can be found, for example, in [83].

Keyframe selection Upon the acquisition of a new pair of scene frames, the pose of the subject is estimated from matched features among the current frames and the 3D map. This method guarantees a global consistency across the whole experiment and it is accurate as long as the global map is accurate.
At this point the goal is to detect the change in space of the focus of overt attention in order to identify sequences of PORs that exhibit a coherence in space and time.

The collected scene frames are clustered into subsequences according to the subject's POR and *keyframes* are used to delimit coherent subsequences. Roughly speaking, keyframes consist of scene frames corresponding to time steps in which the focus of overt attention changes and a new sequence of PORs starts. Therefore, a strategy is required to select keyframes when no knowledge of the pose and, thus, of the 3D point of regard of the subject is retained.

We introduce a keyframe selection method that evaluates the *novelty* of a view in the experiment by measuring how different it is from the last selected keyframe. The quantities involved in the keyframe selection are the n matched pairs of visual features $\{(\mathbf{x}, \mathbf{x}')_i, i = 1 \dots n\}$, between the current scene frame and the last keyframe, and the pair (γ, γ') of gaze positions as projected into to the current frame and into the last keyframe. Note that in this phase the correspondences $(\mathbf{x}, \mathbf{x}')_i$ are drawn among frames collected by one of the scene cameras at different timesteps and the pair (γ, γ') refers to coordinates on the image plane.

A change in the subject's vantage point induces a motion of the camera acquiring the scene and a variation of the POR in space. Suppose that the subject, during a search task, is focusing on a particular object in the scene and that her pose, in the experiment frame, can be described by a certain motion model. This will induce a sequence of PORs that

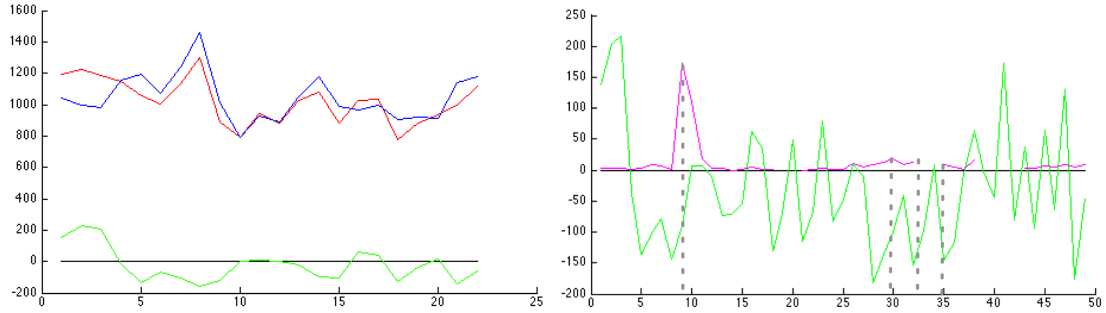


Figure 3.6: Keyframe selection criterion. Left: $\Gamma(\mathbf{F})$ (red), $\Gamma(\mathbf{H})$ (blue) and $\Gamma(\mathbf{F}) - \Gamma(\mathbf{H})$ (green). Right: $\Gamma(\mathbf{F}) - \Gamma(\mathbf{H})$ (green) and δ (magenta). Keyframes are selected in correspondence of dashed lines.

is consistent with the given motion model. Therefore, we evaluate the opportunity to instantiate a new keyframe by checking the consistency of the current POR with a motion model estimated on the basis of frame to keyframe correspondences. We characterize the subject's change in head pose by means of two types of motion models that can be estimated from the scene frames: a planar homography, represented by the $\mathbf{H}$ matrix, and the *fundamental matrix* $\mathbf{F}$ (see [83] for a comprehensive treatment). A motion characterized by a small baseline between the current frame and the last keyframe is best described by a plane homography $\mathbf{H}$. In contrast, when the subject's head undergoes a translational motion, the fundamental matrix $\mathbf{F}$ is more suitable to describe a general camera motion.

Building on the Geometric Robust Information Criterion (GRIC, [88]), a score function is evaluated for both the $\mathbf{F}$ and $\mathbf{H}$ motion models at every frame in order to quantitatively measure the fitness of each model to the data. The score function takes into account the n matched features with the last keyframe, the residuals e_i, the number k of model parameters, the error standard deviation σ, the dimensions r of the data and q of the model:

$$\Gamma = \sum_{i=1}^{n} \rho(e_i^2) + [nq \ln(r) + k \ln(rn)], \tag{3.5}$$

where

$$\rho(e_i^2) = \min\left(\frac{e_i^2}{\sigma^2}, 2(r-q)\right). \tag{3.6}$$

Equation (3.5) returns the lowest score for the model that best fits the data. Once the motion model has been selected, it is used to evaluate the gaze variation, see Figure 3.6. According to the selected motion model, changes in the subject's vantage point involving the gaze projections γ and γ' can be detected and new keyframes are instantiated on the basis of the following criterion, balancing between the choice of an homography $\mathbf{H}$ and of the fundamental matrix $\mathbf{F}$:

$$(\Gamma(\mathbf{F}) - \Gamma(\mathbf{H})) \cdot \delta < 0, \qquad \delta = \begin{cases} \gamma'^{\top}\mathbf{F}\gamma & \text{if } \Gamma(\mathbf{F}) < \Gamma(\mathbf{H}) \\ ||\mathbf{H}\gamma - \gamma'|| & \text{otherwise.} \end{cases} \tag{3.7}$$

Upon the instantiation of a new keyframe at time t, the following steps are performed:

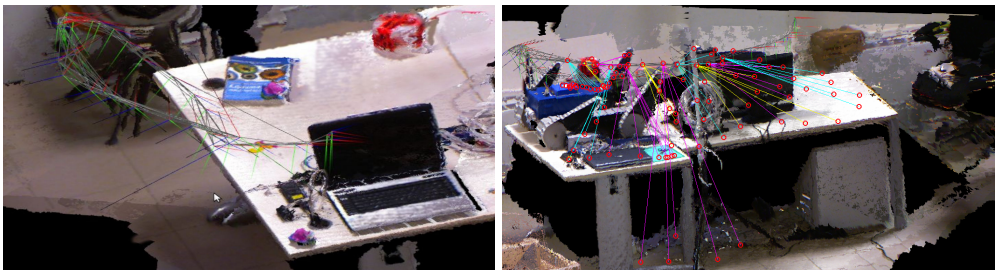


Figure 3.7: Head poses of the subject during the experiment *searching for J*, computed with the described localization algorithm, and the rays joining the head pose with the PORs (the red circles) projected on the scene point cloud. The lines represent, ideally, the intersection of the visual axes.

- *Subsequence Optimization*. Let $\mathcal{X}$ be the set of unoptimized points, then this set is optimized by Sparse Bundle Adjustment (SBA) ([89]) on the sequence of the last k camera poses, using a reprojection error ϵ_{ij} as objective function

$$\min_{\mathbf{R}_i,\mathbf{t}_i,\mathbf{X}_j} \sum_{ij} \epsilon_{ij}, \tag{3.8}$$

  with

$$\epsilon_{ij} = d\left(s\mathbf{K}^l\mathbf{R}_i(\tilde{\mathbf{X}}_j + \mathbf{t}_i), \mathbf{x}_{ij}^l\right)^2 + d\left(s\mathbf{K}^r\mathbf{R}^s[\mathbf{R}_i(\tilde{\mathbf{X}}_j + \mathbf{t}_i) - \mathbf{t}^s], \mathbf{x}_{ij}^r\right)^2. \tag{3.9}$$

  Here $i = t-1, \ldots, t-k$, $\tilde{\mathbf{X}}_j \in \mathcal{X}$ and $\mathbf{x}_{ij}^c$, $c \in \{l, r\}$ is the point $\tilde{\mathbf{X}}_j$ imaged by the i-th left or right camera respectively.

- *Map Upgrade*. Let $\mathcal{M}$ be the global 3D map, built so far, then $\mathcal{M}$ is updated with the new set of optimized points $\mathcal{X}$: $\mathcal{M} = \mathcal{M} \cup \mathcal{X}$.

- *Subsequence Initialization*. The set of optimized points is emptied and the number k of camera poses is set to 0.

When a new keyframe is selected, the previous subsequence is terminated, the correspondent points and cameras are optimized and the resultant structure is added to the global map. Each subsequence as defined above is a *coherent subsequence* as it collects a coherent set of PORs, on a specific region in space.

Figure 3.7 illustrates the head pose and the PORs related to the scanpath elicited during the search task *looking for J* (see Section 3.5).

3.3 Coherent features for point saliency

In the previous section we illustrated how to compute the head scanpath, leading to coherent subsequences of head poses and gaze directions. Once the head poses are retrieved, retrieving the scene structure can be done using the computed camera poses. The scene structure, even if partial, is needed to collect the features of the attended regions. For example, a crucial feature is the space range of PORs, and this is available only if the scene structure is available. Note that by estimating the scene depth, using the computed cameras, a point cloud of the scene structure is obtained.

Figure 3.8: The sequence of images illustrates the notion of *coherent region*. Here the coherent regions induced by a subsequence of PORs are highlighted in red. They are identified among the frames collected during a search experiment with the GM on the street. In this case the experiment was "looking for a parking fine". The PORs are shown as white circles, while the current POR is shown as a white cross.

In this section we illustrate how the coherent subsequence of frames, the point of regard in space and the fixations on the retinal plane can contribute to the definition of the set of features that best specify the visual search task. Though we remark that each search task experiment cleaves the feature set into some unknown prior component; this prior component cannot be recovered experimentally from the PORs data, as it is

embedded into some prior knowledge the subject has about the shape, dimension and color of both the environment and the object, while she is performing the search.

Now, in our experimental approach, we build an inverse problem, namely given the PORs, the head scan path and the points in the image, we want to determine the properties that are common to all of the experiments. Once these properties are identified then, as described in the following section, we can use them to attempt to define a forward model.

Here we want to recover the features that elicited the PORs, from the scene structure, as computed from different experiments. Features are specific for both the space geometry, such as position on a surface and orientation, and the image such as color and intensity variation. Slightly changing the notation adopted in the previous section, in the following we shall denote a non-homogeneous point in space or on the retinal plane as $\mathbf{X}$ and $\mathbf{x}$, respectively, while in the previous section they were denoted by $\tilde{\mathbf{X}}$, and $\tilde{\mathbf{x}}$. On the other hand, when a homogeneous point is needed we shall denote it $\hat{\mathbf{X}}$ or $\hat{\mathbf{x}}$.

Let us consider a coherent subsequence of frames in terms of the set of collected PORs $\mathcal{X} = \{(\mathbf{X}_1, t_0), \ldots, (\mathbf{X}_m, t_q)\}$, $\mathbf{X}_j \in \mathbb{R}^3$, labeled with the time stamp of their acquisition. It is easy to show that two PORs, even if the same region has been observed at time t and t', cannot coincide, as none is able to observe exactly the same point in space twice. Therefore given the camera $\mathbf{P}_j = \mathbf{K}[\mathbf{R}_j \mid \mathbf{t}_j]$, there is only one retinal plane $\mathcal{I}_h$ where the POR $\mathbf{X}_h$ is imaged. However if we consider the region around the POR then the points in the region can be imaged into different retinal planes.

Now, for each coherent subsequence, define a monotonic grid of about 12×10^3 nodal points $n_{\mathbf{X}} = (X, Y, Z)^\top$; then we approximate the point cloud with a thin plate surface $S : V \mapsto \mathbb{R}^3, V \subset \mathbb{R}^2$ minimizing the energy functional:

$$\mathcal{M}_\alpha(S) = \sum_{i=1}^{n} (S(X_i, Y_i) - \hat{Z}_i)^2 + \eta \int_\Omega S_{XX}(X,Y)^2 + 2S_{XY}(X,Y)^2 + S_{YY}(X,Y)^2 dX dY. \tag{3.10}$$

Here $S(X,Y) = Z$, and $\hat{Z}_i$ is the depth of the i^{th} point in the point cloud, $S_{XX}(\mathbf{v})$, $S_{YY}(\mathbf{v})$, $S_{XY}(\mathbf{v})$ are the second order derivatives of S, η is a stabilization parameter, and $\Omega \subset \mathbb{R}^2$ is the surface domain; the first term in the rhs of (3.10) is the penalty term and the second one is the stabilizing functional, for the energy functional, see [90].

A ray $\mathbf{X}(\lambda) = \mathbf{P}^+\mathbf{x} + \mathbf{C}\lambda$ backprojecting a point $\mathbf{x} = (x, y, 1)^\top$, where $\mathbf{P}^+$ is the pseudo inverse of the current camera matrix, and $\mathbf{C}$ its center, shall intersect the surface S into a point $\mathbf{p} = (X, Y, S(X,Y))^\top$, when this point is a POR, it is denoted $\mathbf{p}^\star$. The surface patch around such a point $\mathbf{p}^\star$, is defined according to a distance threshold a; this surface patch is reprojected on the retinal planes of the subsequence, and forms a patch on the retinal planes which is defined the *coherent region*. Therefore a coherent region is the foveated area in the image surrounding a gaze direction. Coherent regions in images are illustrated in Figure 3.8.

Given the surface approximating the point cloud, we can sample from the whole data set, retrieved from an experiment, two different set of points: the points on the surface patches centered at $\mathbf{p}^\star$, the pixels on the coherent regions on the retinal planes, and those points, on S and on the retinal planes, who have never been observed, according to the current subsequence. Once these points have been transformed into a feature space, we can obtain a training set $(\mathbf{W}, h)$ such that $h = 1$ if the back transformed item comes from a POR region and $h = -1$ otherwise.

Given a coherent subsequence $\mathcal{I}_1, \ldots, \mathcal{I}_q$ in a time interval $(t_0, t_0 + \Delta t)$, and its associated collection of PORs $\mathcal{X} = \{(\mathbf{X}_1, t_0), \ldots, (\mathbf{X}_m, (t_o + \Delta t)\}$, $\mathbf{X}_j \in \mathbb{R}^3$, labeled

with their time stamp, a surface S, and a region $\mathbf{s}_P = \{\mathbf{p} \in S \,|\|\mathbf{X} - \mathbf{p}\| \leq a\}$, with a the distance threshold indicated above, then for each point in $\mathbf{s}_P$ there is a pixel $\mathbf{x}$ and a retinal plane $\mathcal{I}_s$, $1 \leq s \leq q$ imaging it. Therefore the set of data, obtained from the POR regions, given a coherent subsequence, in a time interval $(t_0, t_0 + \Delta t)$ and the surface S, is:

$$\{(\mathbf{p}, (\mathbf{x}_1, \ldots, \mathbf{x}_m))|\mathbf{p} \in S,\ \|\mathbf{X}^P - \mathbf{p}\| < a,\ \hat{\mathbf{x}}_j = \mathbf{P}_j\hat{\mathbf{X}}(\lambda), 1 \leq j \leq m,\ \text{with } \mathbf{x}_j \text{ on some retinal plane } \mathcal{I}_j \text{ in the subsequence}\}. \tag{3.11}$$

Here $\hat{\mathbf{x}}$ and $\hat{\mathbf{X}}$ are the homogenized version of $\mathbf{x}$ and $\mathbf{p}$, respectively. Points not in this set are the non-observed ones, and are sampled uniformly on the surface and projected on to the corresponding retinal planes points.

Given the above sample set, it is possible to introduce a set of functions mapping points $\mathbf{p} \in S$ and points $\mathbf{x} \in \mathbb{R}^2$ to a suitable feature space. In feature space it is then possible to learn the function f separating points belonging to salient regions from all the other ones. More precisely, we introduce a set of transformations $\mathcal{F}$ mapping $\mathbf{p} \in \mathbb{R}^3$ and $\mathbf{x}_j \in \mathbb{R}^2$, $j = 1, \ldots, m$, into a feature space, then the learned function f is such that $f\left(\{\mathcal{F}\} \cdot (\mathbf{p}, (\mathbf{x}_1, \ldots, \mathbf{x}_m))\right) = h$, $h \in \{1, -1\}$. Here the $\cdot$ indicates that a transformation in $\mathcal{F}$ is applied to the specific set of points, as specified below. We aim at: (1) identify the optimal set of features characterizing a search task and (2) define the function f that separates regions that can/should be attended, according to the search task, from the not attended ones.

A large amount of literature on feature selection (see for example [91] and references therein) uses a discriminative model, based on the well known family of Support Vector Machines (SVMs) [92], to select the most significant features among a starting base set. Given the set of all possible separating hyperplanes, there are two main optimality criteria for identifying the best one: ℓ_1 and ℓ_2-norm. In the former case, the 1-norm SVM ([93]) with the ℓ_1-norm, known as lasso penalty is obtained. In the latter case, standard SVM ([94], [95]) is obtained and the ℓ_2-norm is indicated as ridge penalty. In [96] it is argued that 1-norm SVM have advantages over the standard 2-norm, when there are redundant features. The simplest method for achieving feature selection is recursive feature elimination [91], assigning a relative importance to a feature, according to its weight vector within the SVM classifier (see below eq. (3.17)). This method allows to remove more than a single feature at a time, once a threshold has been identified.

A first observation for feature selection is that the data collected by the Gaze Machine are available only for training and feature selection, while in general data are taken with a freely moving camera, maybe mounted on a robot pan-tilt head. In general we expect that visual search is performed by a single moving camera, the camera localization and the camera parameters are available during search, a surface patch S for each coherent subsequence is available, though obviously the PORs are available only for the training dataset. Therefore no data specific of the GM can be selected.

Given the surface S, a point $\mathbf{p} = (X, Y, S(X, Y))^\top$ on it and its projection $\mathbf{x}$, we consider different surface parameters that can be obtained from the first and second derivatives of S, in space, and of the image intensity L. The surface $S(X, Y) = Z$ is parametric; let S_X, S_Y be the first order partial derivatives and S_{XX}, S_{YY}, S_{XY} be the second order ones. In the following we identify the surface S with its parametrization.

Let $\mathbf{p}$ be a point on S, the normal N at $\mathbf{p}$ is:

$$N = \frac{S_x \times S_y}{|S_x \times S_y|}. \tag{3.12}$$

Let $\mathbf{v}$ be a vector on the tangent plane at $\mathbf{p}$, the matrices of first and second form for S are:

$$g = \begin{bmatrix} S_x^\top \cdot S_x & S_x^\top \cdot S_y \\ S_x^\top \cdot S_y & S_y^\top \cdot S_y \end{bmatrix}, \quad H = \begin{bmatrix} S_{xx}^\top & S_{xy}^\top \\ S_{xy}^\top & S_{yy}^\top \end{bmatrix} \cdot \begin{bmatrix} N & \mathbf{0} \\ \mathbf{0} & N \end{bmatrix}. \tag{3.13}$$

The above matrices are both symmetric and $det(g) > 0$. Then we consider the Gaussian curvature $K_G = det(H)/det(g)$, namely:

$$K_G = \frac{H_{11}H_{22} - H_{12}^2}{g_{11}g_{22} - g_{12}^2}. \tag{3.14}$$

Actually we considered also the mean Gaussian curvature. Namely, let the best values for $H(\mathbf{v})$ be obtained by $\|\mathbf{v}\| = 1$ and by maximizing the quadratic form $\mathbf{v}^\top H\mathbf{v}$, under the constraint that $\mathbf{v}^\top g\mathbf{v} = 1$. Call these maximal values κ_1 and κ_2. Then the mean Gaussian curvature is:

$$K_M = \frac{\kappa_1 + \kappa_2}{2}. \tag{3.15}$$

We have verified that K_G is more influential than K_M, we indicate the Gaussian curvature of the surface S as σ_S.

Similarly, consider the patches with points $\mathbf{x} = (x, y)^\top$, corresponding to the surface patch with each $\mathbf{x}$ the projection of $\mathbf{p}$ according to the current camera. The Gaussian curvature for the RGB surface is specified as:

$$\sigma_L = \eta_1\eta_2. \tag{3.16}$$

Here η_1 and η_2 are obtained as κ_1 and κ_2 considering the RGB surface. Therefore also for the intensity surface we have considered the principal curvatures. Both σ_S and σ_L are invariant to rotation.

The last feature that turned out to be important is the task domain, namely the range of the values $\mathbf{p}$ corresponding to PORs. Their importance, as gathered above, is quite intuitive, since we do not search in general an item in the sky unless we know in advance that it can challenge gravity. Clearly the constraints on the range can be given only on S. We define $\mathcal{R}_\tau$ to be the plausibility interval $((X_{min}, X_{max}), (Y_{min}, Y_{max}), (Z_{min}, Z_{max}))$ for a search task τ.

We can now list the features we have inferred. For the scene structure:

- $\mathcal{F}_1$: the surfaces points on S_i, given in global coordinates, whose center $\mathbf{0}$ is the search task starting point; the surfaces are matrices $n \times 3$;
- $\mathcal{F}_2$: σ_S for each patch corresponding to nodal points $\mathbf{p}$ on the surface;
- $\mathcal{F}_3$: the plausible interval $\mathcal{R}_\tau$ on the surface domain;
- $\mathcal{F}_4$: the timestamp.

For the image structure, for each point $\mathbf{x}$, image of $\mathbf{p}$ in frame $\mathcal{I}$, the features are defined as follows:

- $\mathcal{F}_5$: $L(x, y) = L(\mathbf{x})$ the RGB values of the pixels;
- $\mathcal{F}_6$: σ_L;
- $\mathcal{F}_7$: a patch size consistent with a meaningful distance Z of the projected point $\mathbf{p}$, namely we fix the maximum depth to 3m. and the acute vision angle to about 15 degrees.

This concludes the set of feature operators. We consider a feature point $\mathbf{W} = \{\mathcal{F}\} \cdot \{(\mathbf{p}, (\mathbf{x}_1, \ldots, \mathbf{x}_m))\}$. Following the approach of [97], we map this set into the vector space defined by a kernel function and set a maximum margin classification problem to separate the data from the origin. Let $\Phi : \mathbb{D}^n \to \mathcal{V}_k$ represent a mapping to the vector space $\mathcal{V}_k$ corresponding to the kernel function $\mathcal{K}$. The separating hyperplane in $\mathcal{V}_k$ space is computed by solving the quadratic program

$$\min_{w \in \mathcal{V}_k, \xi \in \mathbb{R}^+, \rho \in \mathbb{R}} \frac{1}{2}||w||^2 + \frac{1}{\upsilon l} \sum_i \xi_i - \rho \tag{3.17}$$

$$\text{s.t. } (w\Phi(\mathbf{W})) \geq \rho - \xi_i \quad , \quad \xi_i \geq 0, \tag{3.18}$$

Here ξ_i are slack variables, while υ is a regularization parameter controlling the trade-off between the goals of maximizing the width of the margin and minimizing the training error at the points $\{\mathcal{F}\} \cdot \big(\mathbf{p}, (\mathbf{x}_1, \ldots, \mathbf{x}_m)\big)\}$, which takes value 1. So for a new point $\mathbf{W}$ the side of the hyperplane it falls on in $\mathcal{V}_k$ can be determined by evaluating

$$f(\mathbf{W}) = \text{sgn}((w\ \Phi(\mathbf{W})) - \rho). \tag{3.19}$$

The learned function, in principle, separates salient regions from non salient ones. More precisely, given a set of corresponding points $\{(\mathbf{p}, (\mathbf{x}_1, \ldots, \mathbf{x}_m))\}$, according to some cameras $\mathbf{P_1}, \ldots \mathbf{P}_m$ mapping $\hat{\mathbf{p}}$ into a point $\hat{\mathbf{x}}$ in different scene images of the same bundle; given that $(X, Y, S(X, Y))^\top$ is the point on the surface corresponding to $\mathbf{X}(\lambda)$, and given the feature transformations set $\mathcal{F}$, then $f(\mathcal{F} \cdot (\mathbf{p}, (\mathbf{x}_1, \ldots, \mathbf{x}_m))) = 1$ if this is a point in a possible salient region and -1 otherwise.

Results on the classification performed on the above devised feature set are illustrated in Section 3.5. We can note that for a 50 sec. search experiment we collect about 1500 frames, since each image has dimension 480×640, then we have a number of points of the order of $10^{8.5}$. On the other hand as at most 7 PORs are gathered in a single frame and for each POR we collect a surface of about 31×31 pixels then we have positive examples of the order of 10^7, since PORs are often in the same region. Therefore we have rather sparse matrices. The outcome of these experiments is to validate the feature set across different search tasks and to understand what is missing, what is actually part of a prior ability of the searcher and cannot be recovered from the data.

3.4 Generating Proto-Objects

In the previous sections we have illustrated a model for head and point of regard localization in space for a gaze machine that can be worn by a subject looking for specific objects in the environment. Using the model we have identified several features, among which we sorted out the most relevant ones for learning a function that can separate the attended regions from the unattended ones, given a specific search task. Note that the

function needs to be learned for each task, to cope with the PORs elicited during the specific visual search experiment, though the set of features remain fixed: it is like a continuous recalibration process.

This lack of generalization is to be expected, human visual-search relies on an inner model able to generalize search abstracting from the context and the specific task. We argued in the introduction that this might be a consequence of the way features are aggregated into a coherent structure, that is, a proto-object.

If the unknown function to be learned has to be one generalizing all the learned functions for all the search tasks, then it should be a function minimizing a distance from all the learned functions, for all the experimented tasks. This function u should be one minimizing the following functional:

$$E(u) = \int_{\mathcal{L}} \int_{\Omega} w(\mathbf{X}) \|u_X(W) - f(W)\|^2 d\mathbf{X} df. \tag{3.20}$$

Here f is any function learned for the task of visual search, with $\mathcal{L}$ its domain, w is a weight given to the features selected within classification, and $\mathbf{X}$ the observations. In other words, given a search task, the observations, the models specified by the features and the learned function space, $E(u)$ returns the function u which is as close as possible to the value of any possible function selected by the learning process, where the distance is weighted by the features

Here, however, rather than deriving the function u we propose a forward model, based on the previously selected features, which generalizes the learning results. The model is based on wave motion, more specifically it is governed by the equations of a vibrating membrane, with the membranes distributed on the surface S and having an initial displacement induced by the selected features at the specific location.

The main idea of the model is to mimic the stimulus activation, during search, by integrating the features into a vibrational energy. Indeed, due to the initial displacement, the vibration model returns a vibrational energy that is higher where proto-object are expected to be generated and lower or null elsewhere.

In the following, after recalling the model of the finite circular membrane we show how its motion is determined by its initial displacement, induced by the features integration strength. Note that here we do not consider possible interferences between two or more membranes. This will be considered in future works. In Figure 3.9 we illustrate the underlying structure of the proposed model.

The general equation for a vibrating circular membrane, occupying a finite region, is the following,

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}\right) \quad 0 \leq r < a, \theta \leq 2\pi, t > 0. \tag{3.21}$$

This admits a solution by separating variables, and using the positive roots of the Bessel functions of first and second kind. In particular, if the membrane is finite, as in our case, the Bessel functions of the second kind, of any order, are excluded from the solution. Indeed, the general solution of (3.21), for a membrane that is held fixed at the boundary, $r = a$, and it is finite, is obtained using the Bessel function of the first kind of any order as follows:

$$u(r,\theta,t) = \sum_{m,n} \{\alpha_{mn} \sin(j_{mn}t) + \beta_{mn} \cos(j_{mn}t)\}\{\alpha^{\star}_{mn} \sin(m\theta) + \beta^{\star}_{mn} \cos(m\theta)\} J_m(j_{mn}r). \tag{3.22}$$

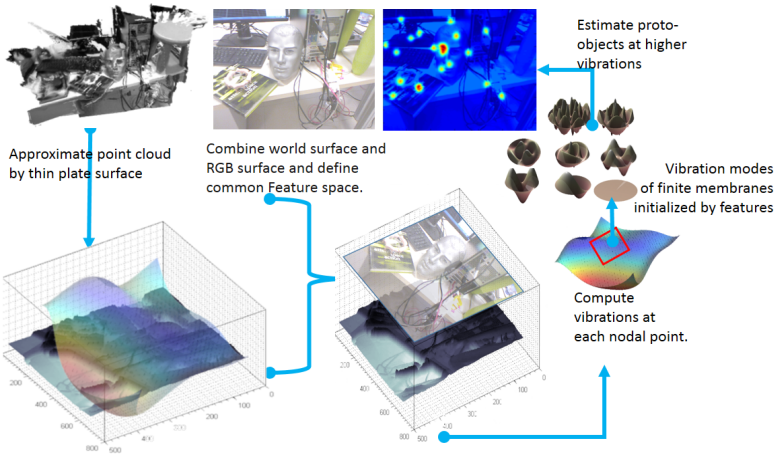


Figure 3.9: The figure above illustrates the model for generating proto-object based on wave motion. The model generates vibration at nodal points where, according to the integrated features a stimulus should occur.

Here J_m is the Bessel function of the first kind of order m, j_{mn} is the n-th root of J_m and $\alpha, \beta, \alpha^\star$ and $\beta^\star$ are constants that can be determined by the initial conditions of the membrane. We recall that the Bessel functions are the solutions of the second order differential equation

$$z^2 \frac{d^2y}{dz} + z\frac{dy}{dz} + \left(z^2 - m^2\right) y = 0. \tag{3.23}$$

With two classes of solution, the J_m of the first kind and the Y_m of the second kind. Though, as observed above, here the Bessel of the second kind is disregarded.

The interest of the membrane is in its vibration modes, they provide a plausible model for integrating features and, accordingly, they release energy via their displacement, and because of the Bessel function the energy vanishes in time.

The main aspect of the model is to provide the right initial displacement so that a solution is found in closed form, for up to a certain order, and the energy induced pulls attention or it fades away, as suggested in the coherence theory.

Let (r, θ, Z) be the cylindrical coordinates of a nodal point $\mathbf{X}$ on the surface. Let $\sigma = \sigma_{\mathbf{S}} + \sigma_L + \epsilon$ be the surface variations introduced in the previous section (see eq. (3.14, 3.16)). We assume that the initial velocity is zero, namely $\partial u/\partial t|_{t=0} = 0$ therefore the general solution becomes:

$$u(r,\theta,t) = \sum_{m=0,1}^{\infty} \left(\sum_{n=1,2,}^{\infty} \alpha_{mn} J_m(j_{mn} r) \sin(m\theta) + \sum_{n=1,2,}^{\infty} \beta_{mn} J_m(j_{mn} r) \cos(m\theta) \right) \cos(j_{mn} t). \tag{3.24}$$

Using the initial condition $\gamma(r, \theta, 0)$, we can separate the inner summations of the above equation (3.24), for $t = 0$ as follows:

$$\begin{array}{ll} C_m = \sum_{n=1,2,}^{\infty} & \alpha_{mn} J_m(j_{mn} r) \\ D_m = \sum_{n=1,2,}^{\infty} & \beta_{mn} J_m(j_{mn} r) \end{array} \tag{3.25}$$

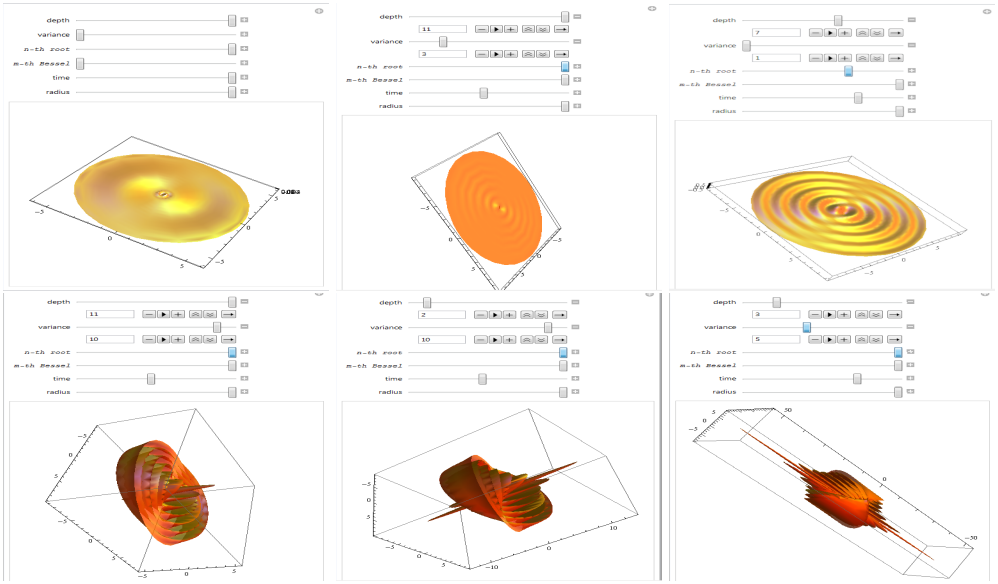


Figure 3.10: Vibrations generated by different initial displacements, according to the initial feature values. The interface made in Mathematica, allows to understand the influence of the Gaussian Curvature σ_S and σ_L, for S and L, specified in the GUI as *variance*, and the distance Z, on the vibration frequency.

and by Fourier series obtain:

$$C_m = \begin{cases} \frac{1}{\pi}\int_0^{2\pi} \gamma(r,\theta,0)\cos(m\theta)d\theta, & \text{for } m \geq 1, \\ \frac{1}{2\pi}\int_0^{2\pi} \gamma(r,\theta,0)d\theta, & \text{for } m = 0 \end{cases} \tag{3.26}$$

and

$$D_m = \frac{1}{\pi}\int_0^{2\pi} \gamma(r,\theta,0)\sin(m\theta)d\theta, m \geq 1. \tag{3.27}$$

Now, we let the initial displacement be given by the following equation:

$$\gamma(r,\theta,0) = 4r\sigma \exp\left(\frac{-z^2}{2\sigma^2}\right)\sin\left(\frac{1}{z}\theta\right). \tag{3.28}$$

This initial displacement ensures that where the surfaces variations σ increase the energy increases too, while the frequency at which the energy is released depends on the radius and the θ values, in such a way that distant points, namely for increasing values of Z, on the surface are penalized. Using equation (3.26) we obtain:

$$\begin{aligned} C_m &= \frac{4zr\sigma \exp\left(\frac{-z^2}{2\sigma^2}\right)\left(-1+\cos(2m\pi)\cos(\frac{2\pi}{z})+mz\sin(2m\pi)\sin(\frac{2\pi}{z})\right)}{\pi(m^2z^2-1)}, \ m>0 \\ C_0 &= \frac{8zr\sigma \exp\left(\frac{-z^2}{2\sigma^2}\right)\sin(\frac{\pi}{z})^2}{\pi} \end{aligned} \tag{3.29}$$

and

$$D_m = \frac{4zr\sigma \exp\left(\frac{-z^2}{2\sigma^2}\right)\left(\cos(\frac{2\pi}{z})\sin(2m\pi)-mz\cos(2m\pi)\sin(\frac{2\pi}{z})\right)}{\pi(m^2z^2-1)}, \ m \geq 1 \tag{3.30}$$

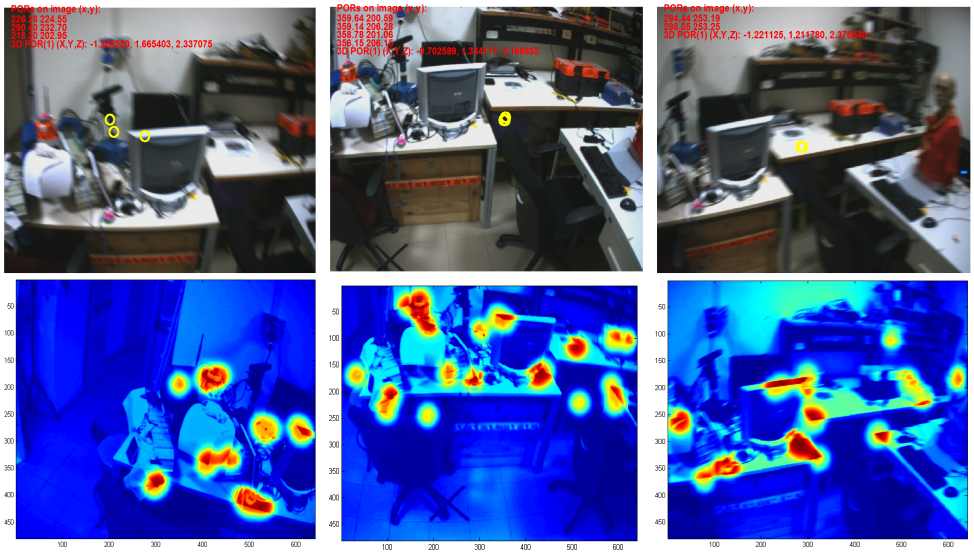


Figure 3.11: Comparison between PORs taken from a coherent subsequence and the inferred proto-objects. We see that in general the generated proto-objects are plausible.

Finally the coefficients α_{mn} and β_{mn} are obtained as follows:

$$\alpha_{mn} = \frac{2}{\pi a^2 J_{m+1}(j_{mn}a)^2}\int_0^a rJ_m(j_{mn}r)C_m = \frac{2^{2-m}\sigma z\Gamma\left(\frac{m+3}{2}\right)\exp\left(-\frac{z^2}{2\sigma^2}\right)\left(j_{m,n}\right)^m\left(mz\sin(2\pi m)\sin(\frac{2\pi}{z})+\cos(\frac{2\pi}{z})-1\right)K}{\pi\left(m^2z^2-1\right)J_{m+1}\left(j_{m,n}\right)^2} \tag{3.31}$$

Here Γ is the Gamma function, $K = {}_1\tilde{F}_2\left(\frac{m+3}{2};\frac{m+5}{2},m+1;-\frac{1}{4}\left(j_{m,n}\right)^2\right)$, denoting ${}_pF_q(a;b;z)$ the regularized generalized hypergeometric function. And the second parameter β_{mn} is given below:

$$\beta_{mn} = \frac{2}{\pi a^2 J_{m+1}(j_{mn}a)^2}\int_0^a rJ_m(j_{mn}r)D_m = \frac{2^{2-m}\sigma z\Gamma\left(\frac{m+3}{2}\right)\exp\left(-\frac{z^2}{2\sigma^2}\right)\left(j_{m,n}\right)^m\left(mz\cos(2\pi m)\sin(\frac{2\pi}{z})-\sin(2\pi m)\cos(\frac{2\pi}{z})\right)K}{\pi\left(m^2z^2-1\right)J_{m+1}\left(j_{m,n}\right)^2} \tag{3.32}$$

Analogously, here Γ is the Gamma function, $K = {}_1\tilde{F}_2\left(\frac{m+3}{2};\frac{m+5}{2},m+1;-\frac{1}{4}\left(j_{m,n}\right)^2\right)$, where ${}_pF_q(a;b;z)$ is the regularized generalized hypergeometric function. Noting that the roots of the Bessel J_m are easily computed with Mathematica, Matlab or Maple, it follows that up to a given order and to a given root, the vibrating membrane takes a solution for varying features values in closed form. Some of the computed membranes with vibrations varying according to the features, inducing the initial displacement $\gamma(r,\theta,0)$ are illustrated in Figure 3.10 showing some of the vibration modes.

The full algorithm to compute the energy elicited by the features structured by the vibrating membrane and to generate proto-object is as follows.

First of all let us define $\mathbb{D} = \bigcup S_{\setminus\mathcal{R}}$ be the domain of all the experiments, in terms of the plausible regions $\mathcal{R}$. Let Q be a coherent subsequence of frames, and $\{\hat{Z}\}_{i=1,\ldots,n}$ the

point cloud for Q, note that a coherent subsequence includes no more than 15 frames, hence it is labeled by a time interval $(t_0, t_0 + \Delta t)$ of less than half second. Let $\mathbf{K}[\mathbf{I}\,|\mathbf{0}]$ be the reference camera and $\mathbf{R}|\mathbf{t}]_1, \ldots, [\mathbf{R}|\mathbf{t}]_m$ the poses of the other views with respect to the reference one.

1. For each nodal point $\mathbf{p}$ of S, such that $\mathbf{p} \in \mathbb{D}$, and for each projected pixel, according to the camera poses, select the regions generated by the points $(\mathbf{p}, (\mathbf{x_1}, \ldots, \mathbf{x}_m))$ restricted to the domain $\mathbb{D}$.

2. Compute the feature set W for the sampled set.

3. Using the above equations, and the obtained features $\mathbf{W}$ at each nodal point, compute the vibrating membrane, allowing the radius r to vary about the membrane distribution on S, between 1 and 5. Here we exploit the pre computation of the Bessel roots in a lookup table.

4. Compute equation (3.24) for each $0 \leq m \leq 12$ and for $1 \leq n \leq 9$. Define the membrane surface as:

$$\left(rm\cos(\theta), rm\sin(\theta), u(r, \theta, t)\right), \tag{3.33}$$

   with t varying from zero to the maximum time lapse of the subsequence interval. Some examples with varying σ, z, and r are illustrated in Figure 3.10. Sum the membrane surface absolute values for each time $t \in (t_0, t_0 + \Delta t)$ and using gradient descent, find the membranes that have maximal energy at $t_0 + \Delta t$.

5. The nodal points with maximal energy are generators of proto-objects.

6. Consider the energy of all the neighbor these selected nodal points, according to the maximal radius a, and identify these patches in S and their projection on the retinal planes of the subsequence as the proto-objects predicting saliency.

Results of this algorithm, for the indoor experiments *looking for J* and *looking for the pink elephant* are illustrated in Figure 3.11.

3.5 Experimental validation

Experiments are at the basis of our experimental model of saliency, whose main stages are shown in the left panel of Figure 3.12.

An experiment, begins with a calibration phase, in which the subject moves her/his eyes, head and body while fixating a specified target. This phase is needed to calibrate the wearable device with the subject eye motion manifold and scene cameras, as illustrated in [16]. Thereafter, according to the search task, the search experiment lasts a certain amount of time T, $120s \leq T \leq 180s$ and it collects the frame sequence F, of the left and right images, at a frequency of $f_T \in [15, 30]Hz$; frames are gathered in bundles specifying the local coherence of the gaze motion. Further it collects the pupil sequence P at a frequency $f_t \in [120,\ 180]Hz$ and the head motion H via a compact inertial device part of the acquisition device. Data are processed off-line and the following set of data is returned together with a synchronization of images, visual axes and head poses: the head pose in global coordinates $\mathcal{H}$ via the localization, [82], the point

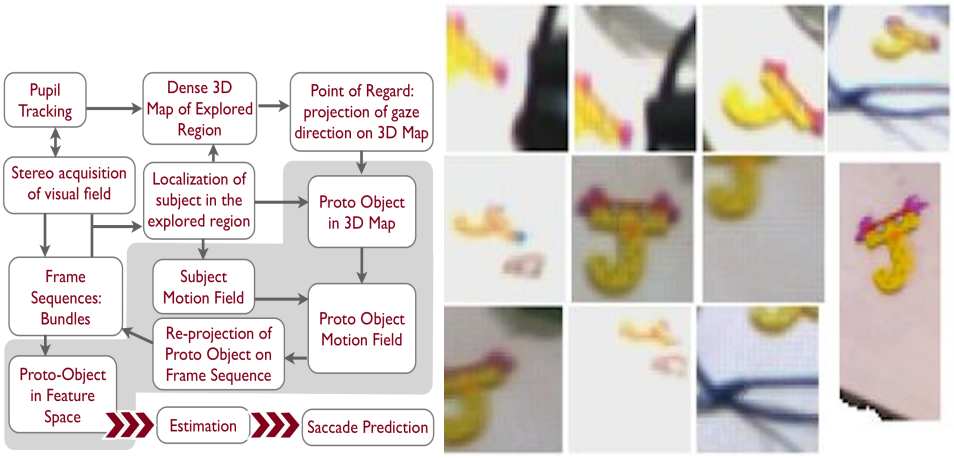


Figure 3.12: The left panel shows the stages of saliency prediction according to our *experimental saliency model*. We use the term *experimental* as it is based on 3D measurements of the gaze in natural scenes and of its motion field. The model copes with the coherence theory of attention with respect to the interpretation of Proto-Objects in early attention stages. On the right the backprojection of proto-objects during the task *looking for J*, the last image in the right panel is a proto-object in the 3D dense map.

cloud $\mathcal{M}$ in global coordinates, the visual axes of the eye manifolds, namely the PORs directions, projected as point in the global coordinates of the scene $\mathcal{P}$, the reprojection of the PORs in the images $\mathcal{R}_{POR}$, synchronized, so that in each image a certain amount of PORs, between 7 up to 15 is reprojected. Finally, $\mathcal{B}$ are the relative positions of the observer with respect to the scene.

An experiment, therefore, comes with the following formal structure:

$$E = \langle \mathcal{H}, \mathcal{M}, (\mathcal{B}, \Delta T), (\mathcal{P}, \Delta t), \mathcal{R}_{POR} \rangle. \tag{3.34}$$

Here ΔT is the time lapse between two measurements of the scene, $\Delta T \approx 60ms$; Δt is the time lapse between two measurements of the PORs direction in the scene, $\Delta t \approx 8ms$ exploiting the scene constancy – namely, the speed of the eyes is faster than any meaningful motion in the scene and of the head and body motion. To these data we add the membrane structures to support the proto-objects. The principal outcomes of an experiment E are the PORs and their localization in the 3D space together with the localization of the head pose in the dense map reconstruction of the scene. These are illustrated in Figure 3.5, Figure 3.7 showing the dense map, the path of the head poses, together with PORs as located in the natural scenes, and in Figure 3.3, showing a meaningful part of an experiment, via a stitched panorama, with the PORs reprojected on the images. A typical dataset with the tracked head poses, a dense point cloud with the projected PORs is illustrated in Figure 3.13.

Experimental validation of the acquisition model Investigating the accuracy of the proposed acquisition model involves different aspects. Localization and mapping of the POR in the 3D scene rely on the estimation of the POR relative position and the localization of the subject in the reference frame of the experiment. In addition, the identification of coherent regions depends on the effectiveness of the keyframe-based mechanism to detect changes in the POR sequence.

A first evaluation focuses on investigating the accuracy of the proposed method in localizing and mapping the PORs. The ground truth has been produced as follows: 5

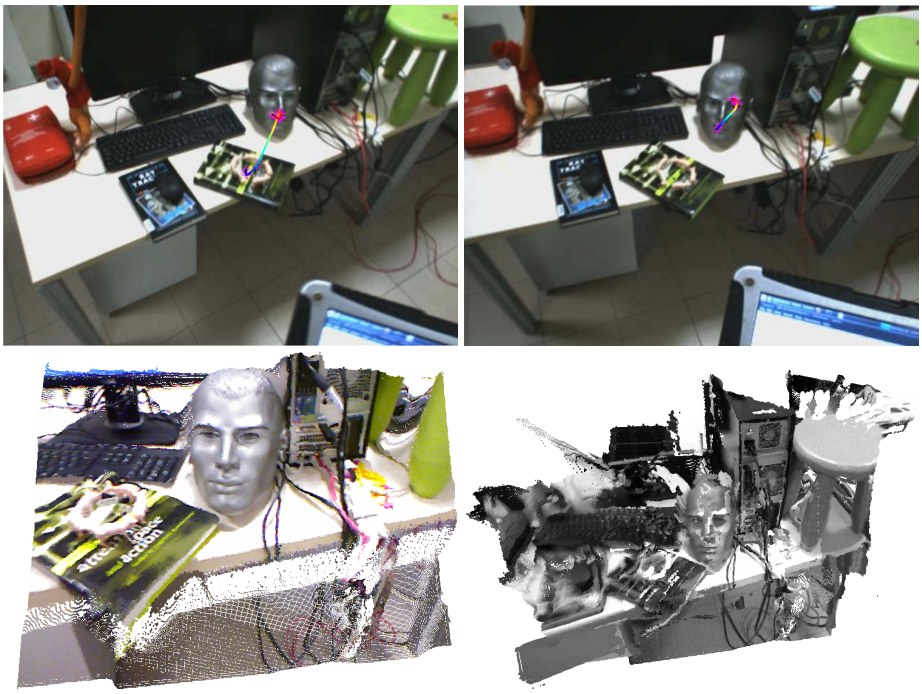


Figure 3.13: Dataset of a visual search experiment with the GM; the dataset includes: point cloud, head scan-path, projection of PORs in space and on the retinal planes.

visual landmarks have been placed in the experimental scenario and their position has been measured with respect to a fixed reference frame; 6 subjects have been instructed to fixate the visual landmarks while freely moving in the scenario, annotating (by voice) the starting and ending of the landmark observations. In each sequence, an average of 60 PORs were produced for each landmark. The validation sequences comprise about 6000 frames each. After registration of the subject initial pose with the fixed reference system, the PORs in the annotated frames were computed and compared with the ground truth, producing a Root Mean Square (RMS) value of 0.094 meters.

For a quantitative analysis of the keyframe selection strategy we relied on a manual coding to produce ground truth data: after the acquisition, subjects were shown the scene sequence overlapped with the POR projection on the image plane and used their innate human pattern recognition skill to select coherent subsequences, annotating for each one the starting keyframe. The performance measure is the *agreement*, defined as the ratio between the number of keyframes recognized by the system over the number of keyframes identified by the subject. Experiments on sequences characterized by a number of frames in the range 4000-6000, yielding a number of keyframes in the range 120-200 produced an average agreement of 85%.

Validation of the coherent subsequence Coherent regions constitute the support for the attended proto-objects during an experiment. Each coherent region also selects, in the related sequence of frames, the appearance of the attended structure that is used to train the saliency model. To validate the method introduced in Section 3.3, we quantified the extent of the coherent region projections in each of the related bundle images. The result for an experiment producing 16 regions, with centroid distances ranging from 1.8 and 8 meters from the observer, is shown in Figure 3.14. For each region, the extent of its projection to the frames of the sequence is evaluated as percentage of the total number of pixels in a frame. Scene frames have size 640×480 pixels in the

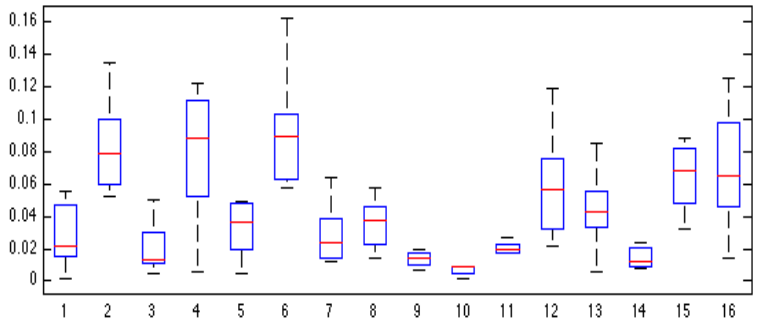


Figure 3.14: Box plot for the extent of 16 coherent regions identified in a GM experiment on the street. The extent of the coherent regions is in percentage with respect to the frame dimension in pixels.

experiments. Figure 3.14 shows the median values, the boxes representing the 25th and 75th percentiles, the minimum and maximum values. The validation confirms that the extent of the projections is mostly confined between 1% and 10% of the image area, and is thus suitable for the proposed feature model.

Validation of the features model Given a visual search task, we have implemented both a slight varied version of [98] and the easier selection addressed in [91]. Focusing on sets of features we obtain the *balanced error rate* as follows:

$$ber = \frac{1}{2}\left(\frac{wp_+}{|D|_+} + \frac{wp_-}{|D|_-}\right). \tag{3.35}$$

Here $|D|_+$ are the positive instances and $|D|_-$ are the negative ones, while wp_+ and wp_- are, respectively, the false negatives and false positives. In the case of the approach of [98], to keep trace of the decrease of the objective function on feature groups, we generate $k!/(k-m)!m!$ m-tuples of even features, up to $k = 5$, so as to assign a *ber* value to each feature group.

A model trained on the complete set of features selected as described in Section 3.3, is able to predict if a new sample point is likely to be attended, i.e., if it belongs to a coherent region, when the experiment is fixed. To validate this assumption, we ran maximum margin classification experiments. A *K-fold cross-validation* strategy has been followed: we divided the available data comprising more than 6 million points in 3 subsets; in turn, 2 of the three subsets have been used to train the classifier and the remaining one for validation.

The process is iterated until every subset is used for validation. As expected, classification accuracy is very high, as reported in Table 3.1.

Table 3.1: Results from the *k-fold cross validation of the maximum margin classification using the complete image+bundle feature set. Here* wp^+ *and* wp^- *are, respectively, the false negatives and false positives.*

iteration	**number of positives**	$wp_+/\|D\|_+$	$wp_-/\|D\|_-$	**accuracy**
1	44707	0.0127	0.0318	95.334%
2	46881	0.01883	0.0206	93.591%
3	420034	0.0093	0.0157	93.019%

Figure 3.15: Results of features and classification validation for the outdoor experiment *looking for parking fines*. In red the PORs, and the coherent patches, in green the estimated point saliency, for the specific task.

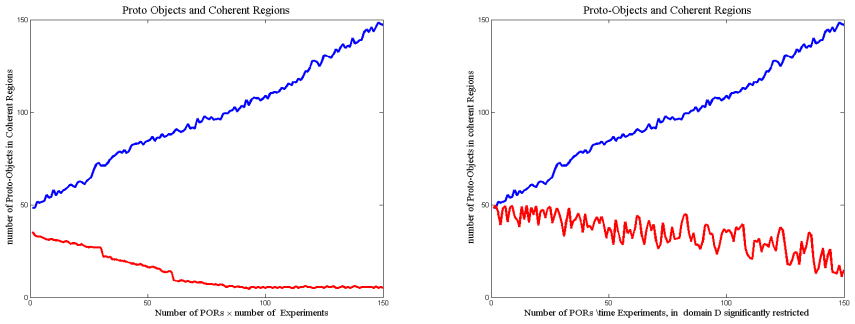


Figure 3.16: Results for computed POR as functions of energy vibration at time $t_0 + \Delta t$, given the domain of the specified experiments, and given the limited domain of selected experiments.

The accuracy is, in particular illustrated in the tables of Figure 3.15, where the outcome of the classification and the measured PORs is highlighted, the first in green and the second in red.

Validation of the vibration model To validate the vibration model we have tested the algorithm described in Section 3.4. The implementation of the membrane has been done in both Mathematica, where a GUI is implemented to study the variations according to the initial displacement conditions, see Figure 3.10, and in Matlab, exploiting a look up table of the Bessel roots computed in Mathematica. We used also the implementation of *gridfit* by [99] for surface approximation. After classification, we have collected the domain elicited by the learned function. And we have generated two sets. The first set with free domains, namely the range of the $\mathbf{p}$ values was given by the domains of all experiments. In the second set we have limited the range to similar domains. The results are illustrated in Figure 3.16. Here the number of PORs per experiments, indicates the $\mathbf{p}^\star$ collected by the GM, with varying experiments, both indoor and outdoor. The number of proto-objects in coherent regions indicates the regions of maximal energy at $t_0 + \Delta t$, computed at the time steps given for the end of a coherent subsequence.

3.6 Conclusions

The computational theory of visual attention aims at mimicking the human capability to select, among stimuli acquired in parallel, those that are relevant for the task at hand. Similar to the biological counterpart, artificial systems can accomplish this by orienting the vision sensors toward regions of space that are more promising. 3D saliency prediction resides in defining a quantitative measure of how attention should be deployed in the three-dimensional scene. Current state-of the art does not model the integration of features in space and time, which is required when dealing with a three-dimensional, dynamic scene. In the coherence theory of attention, as introduced in [72], the concept of proto-object emerged to explain how focused attention collects features to form a stable object that is temporally and spatially coherent. In this work we address the problem of modeling the process of formation of proto-objects and their relative spatial and temporal coherence according to a double process. At first a pure experimental setting allows as to identify the best features, which are stable across different experiments and different contexts. We show their stability using a classifier that has been exploited also to select the best features. Further we define a forward model based on the selected features. The forward model define a vibrational energy capturing coherent proto-objects. These encapsulate the information about the search task and we show that some good approximation results are possible. We have thus shown a whole process which, starting from three-dimensional gaze tracking experiments, extract features that are relevant to predict saliency and introduce a novel energy based model to indicates the salient regions in space.

A drawback of the proposed method is the lack of motion features. We intend to address these aspect in future research, note that for an experimental method as the one proposed here it is required to deal with the reconstruction of motion, which is still a hard problem.

Chapter 4

Confidence driven TGV fusion

In this Chapter, we discuss a novel model for spatially varying variational data fusion, driven by point-wise confidence values. The proposed model allows for the joint estimation of the data and the confidence values based on the spatial coherence of the data. We discuss the main properties of the introduced model as well as suitable algorithms for estimating the solution of the corresponding biconvex minimization problem. Additionally, an extension of the primal-dual hybrid gradient algorithm is proposed and we discuss its convergence. The performance of the proposed model is evaluated considering the problem of depth image fusion by using both synthetic and real data from publicly available datasets.

4.1 Introduction

Variational methods have gained a large popularity advantage over other methods when dealing with ill-posed problems in computational vision. The reason is that they have shown good computational properties and high flexibility in large scale regularization problems, typically those arising in computational image processing applications, as for example image denoising, inpainting and super-resolution. In these contexts the original problem is transformed into an energy minimization problem, by introducing a suitable energy functional which favors some desired characteristics of the optimal solution. More specifically, given a domain $\mathcal{U}$, which is a Banach space, and the extended real line $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$, the energy minimization problem is driven by an energy functional $E : \mathcal{U} \to \overline{\mathbb{R}}$ of the following general form:

$$E(u) = F(Ku) + H(u; d, \lambda). \tag{4.1}$$

Functional H enforces fidelity to the given data d, namely the observations. On the other hand, functional F acts as a regularization on a linear transformation of u, specified by the linear operator K, which usually represents a differential operator. In case both F and H are convex, lower-semicontinuous functions, efficient algorithms for the minimization of E have been proposed, even when F and possibly also H are not differentiable everywhere. First-order proximal splitting algorithms are amongst the most relevant. A well-known method belonging to this class of algorithms is the primal-dual hybrid gradient method (PDHG) [32, 39, 40].

The parameter λ in (4.1) balances the relative importance of the two terms F and H, and it is usually assigned *a-priori* and applied uniformly on the effective domain of E. We consider here λ as a multiplicative parameter applied to the fidelity term H.

When λ is applied as a multiplicative parameter in H, the effect of the parameter is to act as confidence value of the data fidelity term, which is crucial when data come in a multiplicity, and varying in space. Actually, a spatially varying regularization parameter λ has been examined in the past as for example in [100] for the well-known ROF model [33]. However the idea of introducing a spatial prior on the fidelity term to asses confidence on the data accuracy is new, to our knowledge.

Given this background, the main contribution of this work is a new model, which extends (4.1) to govern the fusion of multiple data observations, with occurring spatial overlaps. The fusion problem amounts to integrating redundant and complementary information from several data sources, each bringing different degree of accuracy, which can highly vary especially in the case the source data are depth images. A key aspect of the proposed model is to generalize the energy minimization problem to jointly accommodate estimation of the data and their confidence values, in the following form:

$$E(u) = F(Ku) + H(u, \lambda; d) + G(\lambda). \tag{4.2}$$

This model induces spatially adaptive regularization effects, letting the coherence of the available data guide the regularization process. The corresponding minimization problem is no longer convex, though we show that it is biconvex if G is a convex, lower-semi-continuous functional. On this basis, we extend biconvex optimization algorithms for dealing with non-smooth functionals and examine their convergence. In summary, this work contributes to the data fusion problem with a new model which we present in its discrete version so as to focus on the algorithms and the experiments on different datasets, showing the performance of the model.

Furthermore we present the algorithms Alternative Convex Search (ACS) and Alternate Minimization (AMA), adapted to our model, showing that they converge for the biconvex joint estimation problem and settle the conditions that have to be satisfied to guarantee convergence.

We consider also the PDHG method for our model, contributing with a convergence analysis for the case of *a-priori* assigned spatially varying confidence values, and provide suitable bounds for the PDHG step parameters. Moreover, we extend the analysis of the PDHG algorithm for the joint estimation problem and discuss its convergence.

The remaining of the work is organized as follows. Section 4.2 discusses related work. Section 4.3 introduces the confidence driven data fusion model and its properties. In Section 4.4 we examine the convergence of the alternate minimization (AMA) and the alternate convex search (ACS) algorithms. We also discuss convergence of the PDHG algorithm for spatially varying confidence values. In Section 4.5 we examine numerical results and the performance of the model with respect to state of the art methods for the problem of depth image fusion on real and synthetic data. Finally, in Section 4.6 we provide some conclusions and future work directions.

4.2 Related Work

The idea of spatially altering the effects of regularization, to the best of our knowledge, has been first introduced by Strong and Chan [100] who provided analytical solutions for the minimizers of specific classes of signals. They also considered spatially varying regularization parameters, to locally control the image scale space. Calvetti and Sommersalo [101] use a weighting scheme based on the statistics of the edges in natural

images, proposing the gamma and the inverse gamma distributions as hyper-priors of the regularization term. Their Bayesian regularization model includes the Perona-Malik [102] and ROF [33] models as special cases.

We recall that Total Variation (TV) for image denoising has been introduced by Rudin, Osher and Fatemi (ROF) in [33]. Several generalizations of total variation regularization have been proposed to allow for exact reconstruction of higher-order piecewise polynomial signals, e.g. piece-wise affine or quadratic signals. Some well known such generalizations are the Infimal Convolution Total Variation (ICTV) proposed by Chambolle [36] and Total Generalized Variation (TGV), introduced by Bredies and colleagues [30]. We consider the latter, which further generalizes ICTV. See [28] for further details and comparisons between the ICTV and TGV methods.

Going back to spatially varying regularization effects, Newcombe and colleagues [103] apply weighting parameters in order to ensure lower regularization near image edges, so as to enforce sharp edges of the computed depth image. In a similar way, [104] proposes anisotropic regularization by considering the Nahel-Enkelmann operator applied to the regularization term. In the mentioned works spatially varying weighting is applied to the regularization term. Under this respect, the model we propose shows some important novelties. First of all, the weighting scheme is applied to the fidelity term. This brings a new interpretation for the data fusion problem, in which the different contributions of the data sources are gauged by a map of confidence values. More importantly, the proposed model estimates these confidence values directly from the available data, by solving a biconvex minimization problem. Additionally, the model resorts to a fidelity term based on the L_1 norm, which is quite robust to outliers.

As a result, the proposed method combines the advantages of L_1 regularization, namely robustness against impulsive noise and contrast invariance, which corresponds to purely geometrical effects in the scale space, with the ability to locally control the image scale space, by varying confidence values at each image region. As will be shown in the following, the model entails a biconvex minimization problem, which poses some challenges in finding the optimal solution, with respect to convex TGV models.

As a matter of fact, many interesting problems in image processing can be better modeled with non-convex regularization models. Recently a number of non-convex models have been proposed in order to attack the problems of image inpainting [105], depth smoothing [106], and TV regularization on manifolds [107, 108, 109]. Algorithms for optimizing non-convex functionals have been recently proposed focusing on distinctive properties of the terms involved, we recall here some of them.

The *Alternating minimization* methods transform a constrained optimization problem into an unconstrained optimization one, by adding a quadratic penalization on the constraint violation. Typically the weight on the penalization term increases as the iterations proceed. Examples for this class of algorithms can be found in [110], and convergence properties are discussed in [19].

Splitting methods are used when the problem can be separated in a smooth nonconvex term and a possibly non-smooth part. A recently proposed forward-backward splitting method for dealing with this class of problems, called *iPiano*, was introduced in [111].

Semi-convex regularization is considered when the nonconvex term can be made convex, for example by adding an additional L_2 norm (see Section 2.2.3), [29] proposed a method based on the augmented Lagrangian and proved that it converges to critical points. More recently [112] proposed a PDHG method for problems with a semicon-

vex regularization term. The authors prove convergence of the algorithm to critical points when the convexity of the fidelity term compensates the nonconvexity of the regularization term, and they show various examples where the algorithm converges, even when this assumption is violated, indicating the (possibly local) robustness of the PDHG methods applied to nonconvex problems. Finally, Valkonen in [113] provides a proof of local convergence of the PDHG method in the case of *non-linear regularization operators* (NL-PDHG), when the non-linear operator satisfies certain smoothness assumptions and the operator of the update steps satisfies the Aubin property [114].

The method we propose touches, in some sense, all the problems mentioned above. Indeed, we discuss two algorithms for solving the biconvex optimization problem which gives the optimal solution of our model. First, we consider the alternate convex search algotiyhm [18] and then we examine the application of alternate minimization methods [19], discussing their convergence to critical points. We consider also the application of the PDHG algorithm on biconvex optimization problems and discuss its convergence.

As an application domain we consider the problem of variational fusion of depth images, which is recognized to be a crucial aspect in many surface reconstruction approaches. Campbell *et al.* [115] employ a Markov Random Field to find a solution for multiple depth hypotheses. Merrell *et al.* in [116] adopted a depth image fusion scheme, based on visibility, considering appropriate confidence measures to assess the stability of each depth estimate. In [117] the authors use a reduced dictionary of depth patches to regularize and fuse depth images of mostly planar structures.

Total generalized variation models for the fusion of depth images has been introduced in [118]. In [119], the authors fuse low-resolution high-fidelity depth images, from Time-of-Flight sensors, with high-resolution and low-fidelity depth images, generated from stereo matching, using a primal-dual optimization algorithm on a model based on anisotropic diffusion. As mentioned above, in [106] the authors consider non-convex regularizers and propose an iterative algorithm for the optimization of the corresponding problems, evaluating their method with a number of image processing applications, including depth image fusion.

In a different line of work, [120] proposes a volumetric fusion of the depth images based on Total Variation, to regularize the resulting signed distance function (SDF). In [121] the authors propose a hierarchical SDF, which allows the fusion of depth images with very different scales. [122] proposes a method to both estimate the pose of the RGB-D camera and to integrate new depth images with the reconstructed 3D model. Fusion is performed by taking the weighted average of individual truncated SDFs. Recently, [123] has proposed a surface reconstruction approach from depth images by globally optimizing a signed distance function, defined on an octree grid, which scales very well with the number of input data.

Finally, we mention that image fusion is also treated in other application domains, like medical [124] and hyper-spectral imaging [125], which we do not treat in this work.

4.3 Fusion Model

In this section we introduce the confidence driven fusion model and state some of its main properties. Let $\mathcal{X} \subseteq \mathbb{R}^N$ be a finite-dimensional Hilbert space equipped with inner-product $\langle \cdot, \cdot \rangle$ and norm $\|\cdot\|_2 = \sqrt{\langle \cdot, \cdot \rangle}$, and let $\mathcal{L} \subseteq \mathbb{D}^n_{++}$ be a finite-dimensional Hilbert space equipped with the Frobenious inner product and the associated norm.

The proposed fusion model, making precise the general model (4.2), is the following:

$$E(x,\Lambda) \coloneqq \mathrm{TGV}^l_\alpha(x) + \sum_{k=1}^{K} \big\|\Lambda(x-d_k)\big\|_1 + \frac{1}{2}\,\mathrm{Tr}(W^{-1}\Lambda) - b\log\det\Lambda, \tag{4.3}$$

with $(x,\Lambda) \in \mathcal{X}\times\mathcal{L}$, $W\in\mathcal{L}$, $b>0$.

For $N\to\infty$ an infinite dimensional version of (4.3) is obtained. We focus though on the finite dimensional case and show the main properties of the proposed model for confidence driven fusion, which demonstrate the regularization behavior of the model and are essential for the convergence analysis of the algorithms considered in Section 4.4.

4.3.1 Convexity

Proposition 4.1. *The model (4.3) is biconvex on* $\mathbb{R}^N\times\mathbb{D}^n_{++}$.

Proof. Given $B\coloneqq\mathbb{R}^N\times\mathbb{D}^n_{++}$, which is a convex set, we show that (4.3) is biconvex. Indeed, for fixed $\bar\Lambda\in\mathcal{L}$, both the L_1 norm and the TGV functional are convex in x, hence (4.3) is convex on the convex set $B_{\bar\Lambda}\coloneqq\{x\mid(x,\bar\Lambda)\in B\}$. On the other hand, for fixed $\bar x\in\mathbb{R}^N$, the L_1 norm, the trace and the $-\log\det$ operators are convex in Λ [26], hence (4.3) is convex on the convex set $B_{\bar x}\coloneqq\{\Lambda\mid(\bar x,\Lambda)\in B\}$. It follows that (4.3) is biconvex on B. □

The following example shows that (4.3) is in general not convex in (x,Λ).

Example 4.1. Let $K=1$, $W^{-1}=2I$, $b=(e-1)(e+2)^{-1}$, $d_1=0$ and consider the values $z_0=(\vec 0,I)$, and $z_1=(2\cdot\vec 1,e^{-1}I)$, for the joint variable $z\coloneqq(x,\Lambda)$. We have

$$E(z_0)=N,\ E(z_1)=N(1+3e^{-1}). \tag{4.4}$$

It follows that

$$E_{1/2}\coloneqq\frac{E(z_0)+E(z_1)}{2}=N(1+\frac{3}{2}e^{-1}), \tag{4.5}$$

and

$$E\left(\frac{z_0+z_1}{2}\right)=E_{1/2}+N\left(\frac{e-1}{e+2}\log 2\right)>E_{1/2}. \tag{4.6}$$

Hence, (4.3) is in general not convex in z.

Note 4.1. *The previous result shows that the model (4.3) is not convex with respect to the joint variable* (x,Λ). *Hence, in general its minima do not form a compact connected set.*

Proposition 4.2. *The model (4.3) is* $\sqrt{N}$*-semiconvex.*

Proof. First, we show that the fidelity term $\big\|\Lambda(x-d)\big\|_1$ is semiconvex, namely that $D(z)\coloneqq\big\|\Lambda(x-d)\big\|_1+\frac{\omega}{2}\|\Lambda\|_2^2+\frac{\omega}{2}\|x\|_2^2$ is convex, for $\omega\geq\sqrt{N}>0$. That is, for $\gamma\in[0,1]$, and $z_1=(x_1,\Lambda_1)$ and $z_2=(x_2,\Lambda_2)$ we have $D(\gamma z_1+(1-\gamma)z_2)\leq$

$(\gamma D(z_1) + (1-\gamma)D(z_2))$. Indeed, denoting $y_i = x_i - d$ and $\gamma^c = (1-\gamma)$, we have

$$\begin{aligned}
&\left\|(\gamma\Lambda_1 + \gamma^c\Lambda_2)(\gamma y_1 + \gamma^c y_2)\right\|_1 - \gamma\|\Lambda_1 y_1\|_1 - \gamma^c\|\Lambda_2 y_2\|_1 && \text{(4.7a)}\\
\leq &\left\|(\gamma\Lambda_1 + \gamma^c\Lambda_2)(\gamma y_1 + \gamma^c y_2)\right\|_1 - \|\gamma\Lambda_1 y_1 + \gamma^c\Lambda_2 y_2\|_1 && \text{(4.7b)}\\
\leq &\left\|(\gamma\Lambda_1 + \gamma^c\Lambda_2)(\gamma y_1 + \gamma^c y_2) - \gamma\Lambda_1 y_1 - \gamma^c\Lambda_2 y_2\right\|_1 && \text{(4.7c)}\\
= &\left\|\gamma\gamma^c(\Lambda_1 y_2 + \Lambda_2 y_2) - \gamma\gamma^c\Lambda_1 y_1 - \gamma\gamma^c\Lambda_2 y_2\right\|_1 && \text{(4.7d)}\\
= &\,\gamma\gamma^c\left\|(\Lambda_1 - \Lambda_2)(y_1 - y_2)\right\|_1 && \text{(4.7e)}\\
\leq &\,\gamma\gamma^c\sqrt{N}\left\|(\Lambda_1 - \Lambda_2)(y_1 - y_2)\right\|_2 && \text{(4.7f)}\\
\leq &\,\gamma\gamma^c\sqrt{N}\|\Lambda_1 - \Lambda_2\|_2\,\|x_1 - x_2\|_2\,, && \text{(4.7g)}
\end{aligned}$$

where convexity of the $\|\cdot\|_p$ operator for $p \geq 1$ is used in (4.7b), triangle inequality in (4.7c), and Cauchy-Schwarz inequality in (4.7f) and (4.7g). On the other hand for the quadratic terms we have

$$\|\gamma u_1 + \gamma^c u_2\|_2^2 - \gamma\|u_1\|_2^2 - \gamma^c\|u_2\|_2^2 = -\gamma\gamma^c\|u_1 - u_2\|_2^2 \tag{4.8}$$

Adding (4.7a-g) and (4.8) for x and Λ, we get

$$\begin{aligned}
&D(\gamma z_1 + \gamma^c z_2) - (\gamma D(z_1) + \gamma^c D(z_2)) && \text{(4.9a)}\\
&\leq \gamma\gamma^c\left(\sqrt{N}\|\Lambda_1 - \Lambda_2\|_2\,\|x_1 - x_2\|_2 - \frac{\omega}{2}\|x_1 - x_2\|_2^2 - \frac{\omega}{2}\|\Lambda_1 - \Lambda_2\|_2^2\right). && \text{(4.9b)}
\end{aligned}$$

From (4.9b) it is immediate that for $D(\cdot)$ to be convex $\omega \geq \sqrt{N}$ must hold. Since all other terms of (4.3) are convex, the statement holds. □

4.3.2 Boundedness

Theorem 4.1. *The model (4.3) is bounded from below.*

Proof. We use the fact

$$\inf_u \sum_v f(u,v) \geq \sum_v \inf_u f(u,v). \tag{4.10}$$

Hence

$$\begin{aligned}
\inf_{x,\Lambda} E(x,\Lambda) \geq &\inf_x \mathrm{TGV}(x) + \sum_k \inf_{x,\Lambda}\left\|\Lambda(x-d)\right\| && \text{(4.11a)}\\
&+ \inf_\Lambda\left\{\frac{1}{2}\operatorname{Tr}(W^{-1}\Lambda) - b\log\det\Lambda\right\} && \text{(4.11b)}\\
\geq &\inf_\Lambda\left\{\frac{1}{2}\operatorname{Tr}(W^{-1}\Lambda) - b\log\det\Lambda\right\}. && \text{(4.11c)}
\end{aligned}$$

The term in (4.11c) has a finite infimum for every $W \in \mathbb{D}^n_{++}, b \geq 0$. To see this for $b > 0$, we differentiate with respect to Λ obtaining

$$\frac{1}{2}\operatorname{Tr}(W^{-1}) - b\operatorname{Tr}(\Lambda^{-1}) = \operatorname{Tr}\left(\frac{1}{2}W^{-1} - b\Lambda^{-1}\right), \tag{4.12}$$

which vanishes for $\hat{\Lambda} = 2bW$.

Substituting back to (4.11c) we get

$$\inf_{x,\Lambda} E(x,\Lambda) \geq Nb\left(1 - \log(\det 2bW)^{1/N}\right) > -\infty. \tag{4.13}$$

For $b = 0$ the infimum is trivially zero. □

Note 4.2. *The previous proofs do not use the fact that $\mathcal{L} \subseteq \mathbb{D}^n_{++}$. In fact they are also valid for $\mathcal{L} \subseteq \mathbb{S}^N_{++}$. We consider here $\mathcal{L} \subseteq \mathbb{D}^n_{++}$ as it simplifies the convergence analysis of the minimization algorithms and is also computationally feasible. Indeed, taking $\mathcal{L} \subseteq \mathbb{S}^N_{++}$, then solutions are computationally feasible only for toy problems.*

Model (4.3) offers a new perspective to the general problem (4.1), focusing on the pair (x, Λ). In fact, a prominent problem in applying models such as (4.1) is in the choice of the regularization parameter, especially in the case of non-smooth models.

In principle, the choice of the regularization parameter is determined by the data coherence with respect to the solution of x represented by the fidelity term. Namely, the formulation using the regularization parameter on the penalty term tries to establish a compatibility of this parameter with the noise in the data. Heuristic rules have been established in this sense, such as for example the well known Hanke-Rause [126, 127] rule explicitly linking the regularization parameter to the fidelity term. This perspective requires some evaluation of the noise level, which turns out to be quite complex when the data comes in a multiplicity, such as in fusion applications.

The approach we propose here does not require a prior knowledge on the noise level since this is implicitly coded in the scalar field represented by Λ, which is estimated by the given data. Here Λ effectively balances the noise level, given by the fidelity term, by spatially adapting the penalization term to the estimated value of x. Since Λ is bounded from above, thanks to the hyperparameters $b > 0$ and $W \in \mathbb{D}^n_{++}$, and given that x is bounded too, we can see that, in principle, the estimation of Λ cannot add any new information where no information is available from the source data. On the other hand, its values depend on the data coherence, adapting to the noise pointwise. These considerations are also illustrated in the optimality conditions of Λ discussed in the next section.

4.4 Algorithms

In this section we examine three different algorithms for finding the critical points of the biconvex model (4.3). We present first an adaptation of the ACS algorithm for the case of non-smooth functionals, and then AMA, which is also commonly used for the solution of non-convex optimization problems. We discuss its application for minimizing (4.3) and its relation with ACS. Both these algorithms introduce convex minimization subproblems. We present the PDHG algorithm for spatially varying confidence values which can be used to solve these convex subproblems. Finally, we discuss the applicability of the PDHG algorithm on the biconvex problem.

4.4.1 Alternative Convex Search

The ACS algorithm [128, 18] is an algorithm commonly used for solving biconvex problems. We discuss here its convergence for minimizing (4.3). ACS is based on a

relaxation of the original problem, by minimizing at each iteration a set of variables which lead to a convex subproblem.

Algorithm 4.1 (ACS)**.** Choose an initial estimate $(x_0, \Lambda_0) \in \mathcal{X} \times \mathcal{L}$. For every $n \geq 0$ iterate

Iter 1 $\Lambda_{n+1} \in \arg\min \left\{ E(x_n, \Lambda) : \Lambda \in B_{x_n} \right\}$,

Iter 2 $x_{n+1} \in \arg\min \left\{ E(x, \Lambda_{n+1}) : x \in B_{\Lambda_{n+1}} \right\}$.

Considering the optimality condition of the optimization problem in Iter 1, the updates for the elements $i = 1, \ldots, N$ of the diagonal of Λ are given by

$$(\Lambda_{n+1})_{i,i} = \frac{b}{\sum_{k=1}^{K} |(x_n)_i - (d_k)_i| + \frac{1}{2}(W)_{i,i}^{-1}}. \tag{4.14}$$

As discussed in Section 4.3, W and b correspond to hyper-parameters of the model (4.3), which result in a regularization of Λ as will be discussed below. Examples regarding the values that can be assigned to W and b and their effect on the solution are discussed in Section 4.5.

Before discussing convergence of ACS for the minimization of (4.3), we review two theorems given in [18].

Theorem 4.2. *Let $B \subseteq \mathcal{U} \times \mathcal{V}$, $F : B \mapsto \mathbb{R}$ be bounded from below, and let the optimization problems at each iteration of ACS be solvable. Then the sequence $\{F(u_n, v_n)\}_{n \in \mathbb{N}}$ generated by ACS converges monotonically.*

Theorem 4.3. *Let $U \subseteq \mathcal{U}$ and $V \subseteq \mathcal{V}$ be closed sets, $F : U \times V \mapsto \mathbb{R}$ be continuous, and let the optimization problems at each iteration of ACS be solvable.*

1. *If the sequence $\{z_n\}_{n \in \mathbb{N}}$ generated by the ACS algorithm is contained in a compact set, then the sequence has at least one accumulation point.*

2. *In addition suppose that for each accumulation point $z^* = (u^*, v^*)$ of the sequence $\{z_n\}_{n \in \mathbb{N}}$ the optimal solution of ACS for $v = v^*$ or the optimal solution for $u = u^*$ is unique, then all accumulation points are partial optima and have the same functional value.*

3. *If for each accumulation point $z^* = (u^*, v^*)$ of the sequence $\{z_n\}_{n \in \mathbb{N}}$ the solution of both iterations are unique then $\|z_{n+1} - z_n\| \to 0$, and the accumulation points form a connected, compact set.*

The following lemma justifies the roles of the terms $\mathrm{Tr}(W^{-1}\Lambda)$ and $b \log\det \Lambda$.

Lemma 4.1. *The sequence $\{\Lambda_n\}_{n \in \mathbb{N}}$, produced by ACS for the model (4.3), is well defined and bounded from above for $b > 0$ and $W \in \mathcal{L}$.*

Proof. We note first that for $b > 0$, (4.3) has a unique attainable optimum with respect to $\Lambda \in \mathcal{L}$ for every x_n, given by (4.14). Additionally, the denominator of (4.14) is always greater than zero for $W \in \mathcal{L}$, thus the sequence $\{\Lambda_n\}_{n \in \mathbb{N}}$ is bounded from above by $2b \max_i \{(W)_{i,i}\}$. □

In the following, we use Theorems 4.2 and 4.3, and Lemma 4.1 to prove weak convergence of the ACS algorithm to the critical points of (4.3).

Proposition 4.3. *The sequence $\{(x_n, \Lambda_n)\}_{n\in\mathbb{N}}$ obtained by applying Algorithm 4.1 for minimizing (4.3), converges weakly across subsequences to critical points of (4.3).*

Proof. The sequence $\{\Lambda_n\}_{n\in\mathbb{N}}$ is bounded by Lemma 4.1. Consequently, the sequence $\{(x_n, \Lambda_n)\}_{n\in\mathbb{N}}$ is bounded due to the boundedness of $\{x_n\}_{n\in\mathbb{N}}$ and $\{\Lambda_n\}_{n\in\mathbb{N}}$. By Bolzano-Weirstrass theorem $\{(x_n, \Lambda_n)\}_{n\in\mathbb{N}}$ has at least one accumulation point.

By Theorem 4.1 and Theorem 4.2, the sequence $\{E(x_n, \Lambda_n)\}_{n\in\mathbb{N}}$, generated by Algorithm 4.1, converges monotonically. Then, by Theorem 4.3 all accumulation points have the same functional value and hence correspond to partial optima of (4.3). Finally, by Theorem 2.2 all partial optima correspond to critical points of (4.3), which proves the statement. □

We note that the optimal solution of Λ at each iteration depends on the current value of x_n and, more specifically, on the coherence of x_n with the data. Following the proof above, the same holds for the optimal solution $(\hat{x}, \hat{\Lambda})$.

The solution of Iter 2 can be estimated using the PDHG algorithm, as discussed in Section 4.4.3.

4.4.2 Alternate minimization method

AMA is another algorithm which can be used to solve biconvex problems (see [19]). Here we briefly review AMA and discuss its convergence for finding the stationary points of (4.3).

Algorithm 4.2 (AMA)**.** Choose initial estimate $(x_0, \Lambda_0) \in \mathcal{X} \times \mathcal{L}$. For every $n \geq 0$ iterate

Iter 1 $\Lambda_{n+1} \in \underset{\Lambda\in B_{x_n}}{\arg\min} \left\{ E(x_n, \Lambda) + \frac{1}{2\nu_n}\|\Lambda - \Lambda_n\|^2 \right\}$,

Iter 2 $x_{n+1} \in \underset{x\in B_{\Lambda_{n+1}}}{\arg\min} \left\{ E(x, \Lambda_{n+1}) + \frac{1}{2\mu_n}\|x - x_n\|^2 \right\}$,

with $\mu_n, \nu_n > 0$ for all n. We observe that Algorithms 4.2 and 4.1 become equivalent for $\mu_n, \nu_n \to \infty$.

Regarding the convergence of AMA for minimizing model (4.3), we appeal to the convergence analysis presented in [19]. In [19] Lipschitz continuity of the gradient of H is required with respect to one of the variables. This is satisfied by (4.3) for the variable Λ. Hence, the AMA algorithm converges for minimizing model (4.3) [19, Theorem 3.3], given that the model satisfyies the Kurdyka-Łojasiewicz inequality at the optimal point $(\hat{x}, \hat{\Lambda})$.

Note 4.3. *Model (4.3) is $\sqrt{N}$-semiconvex (see Proposition 4.2) hence choosing $\mu_n, \nu_n \leq \sqrt{N}$ makes the optimization problem convex. This fact can be used for selecting initial values μ_n, ν_n which make the problem convex at the beginning and progressively increase in order to better approximate the original biconvex optimization problem.*

Regarding the update of variable Λ, each element of its diagonal leads to the following quadratic problem

$$(\Lambda)_{i,i}^2 - a_n(\Lambda)_{i,i} - b\nu_n = 0, \tag{4.15}$$

with

$$a_n = (\Lambda_n)_{i,i} - \nu_n \left(\sum_{k=1}^{K} |(x_n)_i - (d_k)_i| + \frac{1}{2}(W_{i,i})^{-1} \right), \tag{4.16}$$

which has the following closed form solution

$$(\Lambda_{n+1})_{i,i} = \frac{1}{2}\left(a_n + \sqrt{a_n^2 + 4b\nu_n} \right). \tag{4.17}$$

The updates of the variable x can be estimated using the PDHG algorithm.

4.4.3 PDHG for spatially varying confidence values

In this section we examine the application of the PDHG algorithm for minimizing problems with spatially varying fidelity weights and the conditions under which the series $\{x_n\}_{n\in\mathbb{N}}$ converges. Our analysis is based on monotone operator theory. We refer the reader to [27, 39] and the references therein for further details.

For the convenience of the reader we consider here the general formulation (4.2) which is typically used for the PDHG algorithm. Let us consider a Hilbert space $\mathcal{H}$ and denote $\Gamma_0(\mathcal{H})$ the set of proper, lower semicontinuous, convex functions from $\mathcal{H}$ to $\overline{\mathbb{R}}$. Additionally, let

$$E(x, \Lambda) = F(Kx) + \sum_k H(\Lambda(Sx - d_k)) + G(\Lambda), \tag{4.18}$$

with:

- S a selection operator which depends on the order of the TGV operator K. E.g. for TV regularization $S = Id$ and $K = \nabla$;
- $F \in \Gamma_0(\mathcal{Y})$ and $G \in \Gamma_0(\mathcal{L})$;
- $H : \mathcal{X} \times \mathcal{L} \mapsto \mathbb{R}$ is proper, lower semicontinuous and biconvex in (x, Λ);
- $K : \mathcal{X} \mapsto \mathcal{Y}$ a bounded linear operator with induced norm $\|K\| = \{\|Kx\| \mid x \in \mathcal{X} \text{ with } \|x\| \leq 1\} < \infty$.
- All functions have closed-form resolvent operators or they can be solved efficiently with high precision.

The model (4.3) is of the general form (4.18), with

$$F(Kx) \coloneqq \mathrm{TGV}^l_\alpha(x), \tag{4.19a}$$

$$H(\Lambda(Sx - d_k)) \coloneqq \left\| \Lambda(x - d_k) \right\|_1, \tag{4.19b}$$

$$G(\Lambda) \coloneqq \frac{1}{2}\mathrm{Tr}(W^{-1}\Lambda) - b \log\det\Lambda. \tag{4.19c}$$

We consider here that Λ is fixed to the value $\bar{\Lambda}$ throughout the minimization. Applying the Legendre-Fenchel transformation to the functionals F and H, and substituting them in (4.3) we obtain the following equivalent formulation

$$E^\star(x, q, \{p_k\}) = \langle Kx, q\rangle + \sum_{k=1}^{K} \langle \bar{\Lambda}(Sx - d_k), p_k\rangle - F^*(q) - \sum_{k=1}^{K} H^*(p_k), \tag{4.20}$$

with q the dual variable corresponding to F and p_k the dual variables corresponding to H.

According to the Karush-Kuhn-Tucker conditions, the saddle points $\hat{\zeta} = (\hat{x}, \hat{q}, \hat{p}_1, \dots, \hat{p}_K)$ of (4.20) satisfy the following monotone variational inclusion

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \begin{pmatrix} K^\top \hat{q} + \sum_{k=1}^K S^\top \bar{\Lambda} \hat{p}_k \\ \partial F^*(\hat{q}) - K\hat{x} \\ \partial H^*(\hat{p}_1) - \bar{\Lambda}(S\hat{x} - d_1) \\ \vdots \\ \partial H^*(\hat{p}_K) - \bar{\Lambda}(S\hat{x} - d_K) \end{pmatrix}, \tag{4.21}$$

where each row corresponds to the optimality condition of each variable involved in the optimization. We assume that the saddle points of (4.20) form a non empty set. This assumption makes the previous condition also sufficient, hence every point satisfying (4.21) is a saddle point of (4.20).

Algorithm 4.3 (PDHG for spatially varying fidelity weights)**.** Choose an initial estimate $x_0 \in \mathcal{X}$. For every $n \geq 0$ iterate

$$x_{n+1} = x_n - \tau \left(K^\top q_n + \textstyle\sum_{k=1}^K S^\top \bar{\Lambda} p_{kn} \right), \tag{4.22}$$

$$\tilde{x}_{n+1} = 2x_{n+1} - x_n, \tag{4.23}$$

$$q_{n+1} \in (Id + \sigma_q \partial F^*)^{-1}(q_n + \sigma_q K \tilde{x}_{n+1}), \tag{4.24}$$

$$p_{kn+1} \in (Id + \sigma_p \partial H^*)^{-1}(p_{kn} + \sigma_p \bar{\Lambda}(S\tilde{x}_{n+1} - d_k)), \text{ for } k = \{1, \dots, K\}. \tag{4.25}$$

The iterations of Algorithm 4.3 can be rewritten as

$$-\begin{pmatrix} 0 \\ 0 \\ \bar{\Lambda} d_1 \\ \vdots \\ \bar{\Lambda} d_K \end{pmatrix} \in \begin{pmatrix} K^\top q_{n+1} + \sum_{k=1}^K S^\top \bar{\Lambda} p_{kn+1} \\ -K x_{n+1} + \partial F^*(q_{n+1}) \\ -\bar{\Lambda} S x_{n+1} + \partial H^*(p_{1n+1}) \\ \vdots \\ -\bar{\Lambda} S x_{n+1} + \partial H^*(p_{Kn+1}) \end{pmatrix} + P(\zeta_{n+1} - \zeta_n), \tag{4.26a}$$

with $\zeta \coloneqq (x, q, p_1, \dots, p_K)$ and

$$P = \begin{pmatrix} \frac{1}{\tau} Id & -K^\top & -S^\top \bar{\Lambda} & \cdots & -S^\top \bar{\Lambda} \\ -K & \frac{1}{\sigma_q} Id & 0 & \cdots & 0 \\ -\bar{\Lambda} S & 0 & \frac{1}{\sigma_p} Id & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ -\bar{\Lambda} S & 0 & 0 & \cdots & \frac{1}{\sigma_p} Id \end{pmatrix}, \tag{4.26b}$$

which can be represented in the following form

$$-B(\zeta_n) \in A(\zeta_{n+1}) + P(\zeta_{n+1} - \zeta_n). \tag{4.27}$$

Let $O_1 \circ O_2$ denote the composition of operators O_1 and O_2. Solving with respect to ζ_{n+1}, we obtain

$$\zeta_{n+1} = (Id + P^{-1} \circ A)^{-1} \circ (Id - P^{-1} \circ B)(\zeta_n). \tag{4.28}$$

Lemma 4.2. *Matrix P is be bounded, self-adjoint, and strictly positive, namely $\langle \zeta, P\zeta \rangle > 0$, for every $\zeta \neq 0$ for*

$$\sigma_q \tau \|K\|^2 \leq \frac{1}{K+1}, \tag{4.29a}$$

and

$$\sigma_p \tau \left\|\bar{\Lambda}\right\|^2 \leq \frac{1}{K+1}. \tag{4.29b}$$

Proof. P is bounded, and self-adjoint by definition. Considering $\langle \zeta, P\zeta \rangle$ we have

$$\begin{aligned}
\langle \zeta, P\zeta \rangle = &\frac{\|x\|^2}{(K+1)\tau} - 2\langle Kx, q\rangle + \frac{\|q\|^2}{\sigma_q} \\
&+ \sum_{k=1}^{K} \left(\frac{\|p_k\|^2}{\sigma_p} - 2\langle \Lambda x, p_k \rangle + \frac{\|x\|^2}{(K+1)\tau} \right) && \text{(4.30a)} \\
\geq &\frac{\|x\|^2}{(K+1)\tau} - 2\|K\|\|x\|\|q\| + \frac{\|q\|^2}{\sigma_q} \\
&+ \sum_{k=1}^{K} \left(\frac{\|p_k\|^2}{\sigma_p} - 2\|\Lambda\|\|x\|\|p_k\| + \frac{\|x\|^2}{(K+1)\tau} \right). && \text{(4.30b)}
\end{aligned}$$

For $\langle \zeta, P\zeta \rangle$ to be positive, we require that all the terms in parentheses in (4.30b) are positive. Using Young's inequality we recover (4.29). □

Proposition 4.4. *Let A and B be the operators defined in (4.26). If (4.29) is satisfied, Algorithm 4.3 converges to the zeros of the $A + B$ operator, namely* $\operatorname{zer}(A + B)$.

Proof. Equation (4.28) is an instance of the proximal point algorithm as described in [39]. Hence, Algorithm 4.3 converges to $\operatorname{zer}(A+B)$ operators if A is maximally monotone, and matrix P is bounded, self-adjoint, and strictly positive. The latter follows from Lemma 4.2.

To show that A is maximally monotone we follow [39]. The operator $\zeta \mapsto \emptyset \times \partial F^*(x) \times \partial H^*(p_1) \times \cdots \times \partial H^*(p_K)$ is maximally monotone by Theorem 20.40, Corollary 16.24, Propositions 20.22 and 20.23 of [27]. Moreover, the skew operator

$$\zeta \mapsto (M^\top q + \sum_{k=1}^{K} S^\top \bar{\Lambda} p_k, -Mx, -\bar{\Lambda} Sx, \ldots, -\bar{\Lambda} Sx), \tag{4.31}$$

is maximally monotone by [27, Example 20.30] and has full domain. Hence, by [27, Corollary 24.4(i)] A is maximally monotone. □

4.4.4 PDHG for biconvex problems

We consider now the biconvex problem of minimizing (4.3) with respect to the joint variable (x, Λ), using an extension of the PDHG algorithm for biconvex problems.

Algorithm 4.4 (PDHG for biconvex problems)**.** Choose an initial estimate $(x_0, \Lambda_0) \in \mathcal{X} \times \mathcal{L}$. For every $n \geq 0$ iterate

$$\begin{aligned}
\Lambda_{n+1} &\in (Id + \tau_\Lambda \partial G)^{-1} \left(\Lambda_n - \tau_\Lambda \textstyle\sum_{k=1}^{K} \operatorname{diag}((Sx_n - d_k) p_{k\,n}^\top) \right), \\
x_{n+1} &= x_n - \tau_x \left(K^\top q_n + \textstyle\sum_{k=1}^{K} S^\top \Lambda_{n+1} p_{k n} \right), \\
\bar{x}_{n+1} &= 2x_{n+1} - x_n, \\
q_{n+1} &\in (Id + \sigma_q \partial F^*)^{-1}(q_n + \sigma_q K \bar{x}_{n+1}), \\
p_{k n+1} &\in (Id + \sigma_p \partial H^*)^{-1}(p_{k n} + \sigma_p \bar{\Lambda}(S\tilde{x}_{n+1} - d_k)),\ k = \{1, \ldots, K\}.
\end{aligned} \tag{4.32}$$

The iterations of Algorithm 4.4 can be written in the form of (4.27) as before.

There are two important differences in this case with respect to Algorithm 4.3. The first, is that the matrix P is changing at each iteration. It is still possible to guarantee that P_{n+1} is strictly positive at every iteration by considering step sizes that vary in each iteration according to (4.29). The second, and more important, difference is that in this case the operator A is not monotone, and as a result the analysis based on proximal point methods cannot be directly applied to prove that the algorithm converges. We note though that if we find experimentally that the sequences $\{\Lambda_n\}_{n\in\mathbb{N}}$, $\{x_n\}_{n\in\mathbb{N}}$, $\{q_n\}_{n\in\mathbb{N}}$ and $\{p_{k_n}\}_{n\in\mathbb{N}}$ remain bounded and additionally $\|\Lambda_{n+1} - \Lambda_n\| \to 0$, $\|x_{n+1} - x_n\| \to 0$, $\|q_{n+1} - q_n\| \to 0$ and $\left\|p_{k_{n+1}} - p_{k_n}\right\| \to 0$, then the algorithm converges to critical points (see [112, 105]).

4.5 Results

In this section we present numerical results, demonstrating the performance of the proposed confidence driven TGV regularization model. We consider depth image fusion as an application domain for evaluating the confidence driven fusion process.

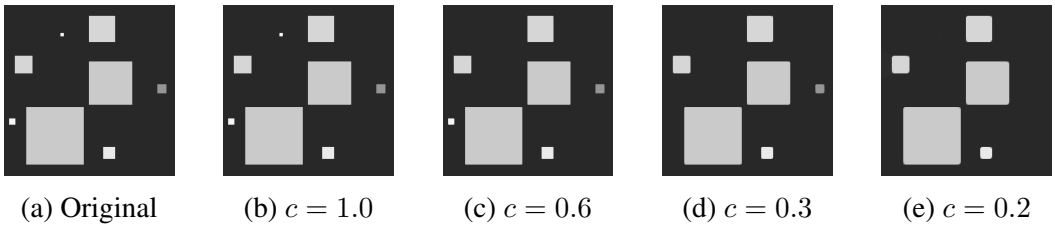
First, we demonstrate numerically relevant properties of the proposed model using synthetic data, highlighting the well-foundedness of the point-wise confidence operator. Then, we thoroughly evaluate the fusion performance of our model using synthetic datasets comprising several 3D models of objects and urban landscapes. Finally, we evaluate our model on real data using a publicly available dataset. The datasets used for the evaluation of the proposed model are provided at `www.diag.uniroma1.it/~alcor/site/index.php/software.html`. An implementation of the proposed model for the fusion of depth images in MATLAB and CUDA is available at `www.github.com/alcor-vision/confidence-fusion`.

4.5.1 Numerical results

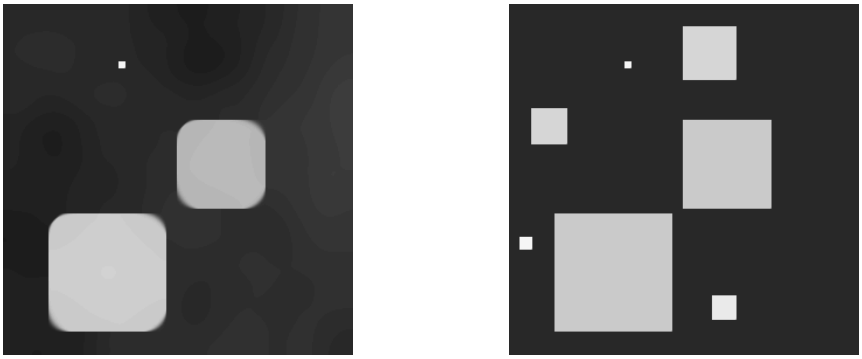
We illustrate here the main properties of (4.3), via numerical results. We considering the effects on a single depth image. In the case of uniform confidence values $\Lambda = cI$ with $c > 0$, the model reduces to the TGV^l- L1 model. This model, for $l = 0$, has been thoroughly examined in the literature (see [100, 129, 35, 32]). Here we are particularly interested in the relation of the confidence values with the scale of the imaged objects.

This relation has been examined in [100] for the original ROF model [33], and in [35] for the TV-L1 model. Indeed, Chan and Esedoglu in [35] argue that the regularization of an image using the TV-L1 model leads small scale objects to suddenly disappear in relation to the value of c. In particular, structures are affected independently of their contrast values, as opposed to the original ROF model where they start to lose contrast as c becomes smaller than a critical value. This observation justifies the use of the L1 fidelity term in (4.3) for the case of depth images, as changes of contrast correspond to distortions of the actual depth values.

The results in Figure 4.1 show how confidence values can affect imaged objects, according to their scale. Let us name B_1 the smallest box on the top-left and B_2 the third smallest box on the right. One can notice in panels (b)-(e) of Figure 4.1 that areas suddenly disappear as the uniform confidence value decreases, based on their size and regardless of their actual depth values. Notice in particular that both B_1 and B_2 disappear for decreasing values of c. The results in panels (f)-(g) of Figure 4.1 show the



(a) Original (b) $c = 1.0$ (c) $c = 0.6$ (d) $c = 0.3$ (e) $c = 0.2$



(f) $c\vec{1}_{\Omega/B_1} = 0.05$, $c\vec{1}_{B_1} = 1.0$ (g) $c\vec{1}_{\Omega/B_3} = 1.0$, $c\vec{1}_{B_3} = 0.2$

Figure 4.1: *First row:* Scale space for uniform confidence. Areas suddenly disappear for different critical values of c based on their scale and regardless of their values. *Left, second row:* Regularization with $c = 1.0$ for the region corresponding to the smallest box B_1 (top-left) and $c = 0.05$ everywhere else. *Right, second row:* Regularization with $c = 0.2$ for the region corresponding to the third smallest box B_3 (middle-right) and $c = 1.0$ everywhere else.

effects of the spatially adaptive regularization. Using spatially varying confidence values, the regularization is locally adapted resulting in smaller scale structures with high confidence values to survive excessive regularization, and, conversely, large scale structures with low confidence to disappear even when moderate regularization is applied. The results in Figure 4.2 show the difference between uniform and spatially adaptive confidence for depth fusion in the presence of Laplace noise.

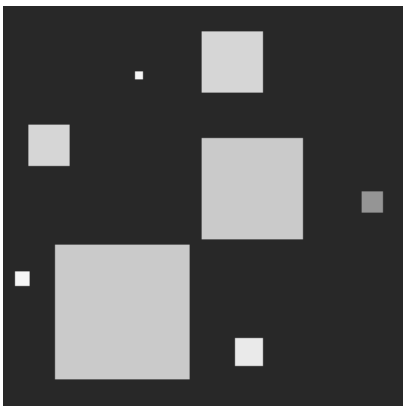
The same considerations hold for higher order TGV regularization, with the only difference that signals of higher order piecewise smoothness (e.g. affine, quadratic etc.) are exactly modeled in this case. This alleviates the well known 'stair-casing' effects of TV regularization.

Summarizing, we see that the proposed model is very effective and versatile for the fusion of depth maps. In fact, it allows for a point-wise median-like estimation of the depth, while at the same time it ensures adaptive regularization according to confidence values which depend on the data.

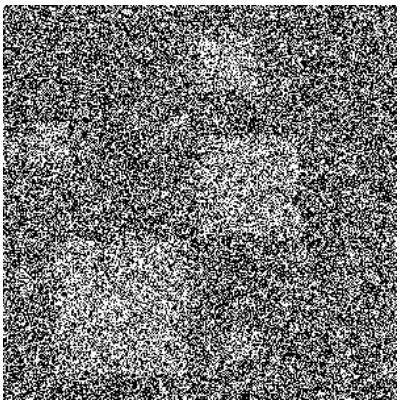
4.5.2 Depth Image Fusion

We consider K cameras. Let R_k be the orientation and $\mathbf{t}_k$ the position of the k-th camera with respect to a global reference frame, with $k \in \{1, \dots, K\}$. Then, each camera pose is represented by the homogeneous transformation

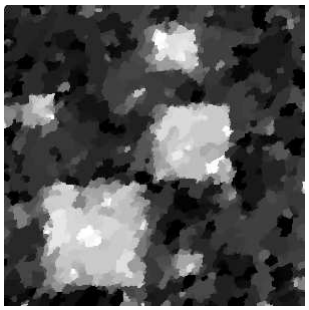
$$T_k = \begin{pmatrix} R_k & \mathbf{t}_k \\ 0 & 1 \end{pmatrix} \in SE(3),\ k = \{1, \dots, K\}. \tag{4.33}$$



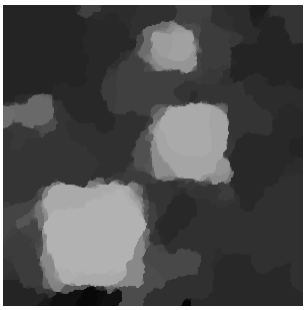
(a) Original Image



(b) Corrupted Image



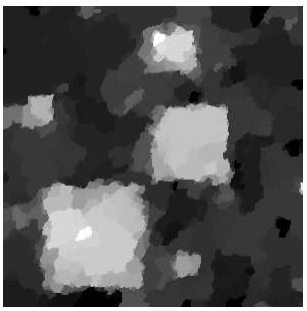
(c) $c = 1.0$



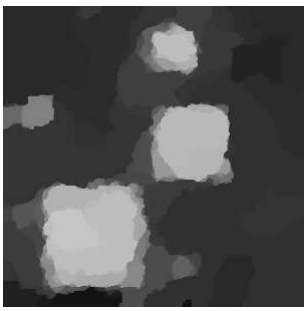
(d) $c = 0.7$

(e) $c = 0.5$

(f) $bW = \mathrm{diag}(1 \cdot \vec{1})$



(g) $bW = \mathrm{diag}(0.7 \cdot \vec{1})$



(h) $bW = \mathrm{diag}(0.5 \cdot \vec{1})$

Figure 4.2: Fusion results for images degraded by point-wise Laplace noise; (c-e) for uniform confidence values; (f-g) for estimated confidence values with different hyper-parameters.

We consider that the scene is projected to the image plane according to the pinhole camera model. Thus, a camera matrix defined as $P_k = APT_k$, with A the camera calibration matrix and $P = [I_{3\times3}, 0]$ the standard projection matrix, corresponds to each camera pose.

Let $\{P_k\}_{k=1}^K$ be a set of camera matrices and $(u, v)^\top = \mathbf{u} \in \Omega \subseteq \mathbb{R}^2$ the spatial variable in the image domain. We denote the corresponding depth images as $(d_1, \ldots, d_K)$, with $d_k : \Omega \mapsto (0, +\infty)$.

Considering a reference camera P_r we denote $\{d_k^r\}_{k=1}^K$ the depth images reprojected to the camera P_r. The reprojection process from camera P_k, $k = 1, \ldots, K$ to the reference camera P_r is defined as follows. Note first that back-projecting the depth map we obtain a 2.5D surface. This surface can be subsequently projected in the reference view, while the pointwise distance of the reference camera from the back-projected

surface forms the reprojected depth map. Let R_k^r and $\mathbf{t}_k^r$ denote the relative rotation and translation of the frame k to the frame r respectively. Each point of the surface, expressed in the frame of the reference view, is given by the linear mapping:

$$\widetilde{X}(\mathbf{u}) = d_k(\mathbf{u}) R_k^r A^{-1} \begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix} + \mathbf{t}_k^r. \tag{4.34}$$

These points are imaged in position $\mathbf{u}' = A\widetilde{X}(\mathbf{u})$ on the image plane of the reference camera. Let $S_{\mathbf{u}'} = \{\mathbf{u} \mid A\widetilde{X}(\mathbf{u}) = \mathbf{u}'\}$ and $e_3 = (0, 0, 1)^\top$. The reprojected depth map is given by $d_k^r(\mathbf{u}') = \min_{\mathbf{u} \in S_{\mathbf{u}'}} e_3^\top \widetilde{X}(\mathbf{u})$.

As a result, at each position of the reference depth image we have up to K depth observations. The fusion process combines these depth observations, taking into account corresponding confidence values, in order to produce a more accurate estimation of the real depth values.

Heuristic Confidence Estimation

We discuss here possible confidence measures for the case of depth image fusion. These heuristic confidence values can be used both as baseline methods for comparison with our complete model, as well as to compute the hyper-parameters W and b of our model. The heuristic confidence measures discussed here are based on the structure and the appearance of the scene.

First, we consider the geometry of the scene. The depth confidence at an image position $\mathbf{u}$ depends on the angle between the viewing ray, given by $\mathbf{r}(\mathbf{u}) = A^{-1}\mathbf{u}/\|A^{-1}\mathbf{u}\|$, and the normal of the surface back-projected at $\mathbf{u}$.

Letting $\mathbf{n} : \Omega \mapsto S^2$ the normal map corresponding to depth image d, with S^2 the unit sphere embedded in $\mathbb{R}^3$, the confidence values are given by

$$(\Lambda)_{\mathbf{u},\mathbf{u}} = \mathbf{n}(\mathbf{u}) \cdot \mathbf{r}(\mathbf{u}) \tag{4.35}$$

Denoting $\mathrm{P}_{S^2}(\cdot)$ the projection operator on the unit sphere S^2 and $\mathrm{D}_u^+, \mathrm{D}_v^+$ the forward differences with respect to directions u and v, the normal map can be estimated as:

$$\mathbf{n}(u, v) = \mathrm{P}_{S^2}\left(\mathrm{D}_u^+(d) \times \mathrm{D}_v^+(d)\right). \tag{4.36}$$

The second heuristic confidence is based on the appearance of the scene and it is based on the observation that image edges often correspond to occlusions and thus depth discontinuities. This suggests a weighting scheme which gives higher confidence on the regions around the edges in order to maintain clean edges.

For simplicity we consider here a linear weighting based on the gradient of the intensity image I, namely

$$\Lambda = \alpha \|G_\sigma * \nabla I\|^\beta, \tag{4.37}$$

with α, β parameters suitably shaping the confidence values, G_σ a Gaussian filter with standard deviation σ and $*$ the convolution operator. The Gaussian filter is useful to control the width of the affected region around the image edges. A similar weighting scheme has been proposed in [103], though the weights were applied, via an exponential function, to the regularization term rather than to the fidelity term. Another related weighting measure based on the Nahel-Enkelmann operator, also applied on the regularization term, was proposed in [104], which also uses the images of the scene.

We note that the geometric confidence tends to assign low confidence values to regions which are orthogonal to the view direction, which often correspond to regions near the edges. The two approaches can be combined to estimate confidence values with desired properties.

Synthetic dataset

We performed an extensive evaluation of the proposed model for the fusion of depth images using synthetic data. We have considered two different classes of 3D models: 1) ordinary small to medium scale objects and 2) models of urban landscapes and buildings. For the objects dataset, we considered the models Bunny, Dragon, Happy Buddha, and Armadillo from the Stanford 3D scanning repository [1, 2, 3] and the objects Chef, Chicken, Parasaurolophus and T-rex from [4]. The urban landscapes dataset contains four models taken from the Sketch-up 3D warehouse.

The two datasets have different characteristics. More specifically, the small and medium scale objects are made by higher-order polynomial terms due to the varying curvature of their surface, while the resulting depth images contain only a small amount of sharp discontinuities. On the other hand, urban landscapes are typically described by lower-order polynomial terms while the resulting depth images contain a significant amount of sharp discontinuities. The motion of the camera also differs (orbiting vs pure translation motion respectively), which affects the occluded regions of the depth images.

Objects We compute depth images corresponding to each of the objects by considering a virtual camera with parameters $(f, c_u, c_v) = (576, 320, 240)\,[px]$ that orbits around the object at a distance of $3\,[m.u.]$ (model units). Depth images are generated every $2\pi/72$ rads. The depth images are generated using [130]. Knowing the exact parameters of the camera the reprojection process produces depth images with correct point-wise correspondences of the depth values, resulting to a fusion problem with perfect data association.

We consider two sets of metrics, the first based on the depth image and the other on the corresponding disparity image. For the disparity image we use the average error in all the valid pixels (avg-all), and the percentage of pixels with disparity error greater than n (out-n) [131]. For the depth image evaluation we use the standard root mean square error (RMSE), the mean absolute error (ZMAE), and the mean angular error of the norms (NMAE) [132]. The average values reported for the synthetic datasets are geometric average values. For the objects dataset the disparity image is generated by considering a virtual baseline with length equal to half the distance between successive views ($3 \sin \frac{\pi}{72}$).

First, we explore the relation of the fused depth image accuracy with the type and the strength of noise for different versions and ablations of our model considering the minimization algorithms discussed in Section 4.4. Naturally, noise is added to the original depth images before the reprojection process. The top rows of Figures 4.3 and 4.4 show the results for Laplace and normally distributed noise with $b, \sigma \in [0, 1]$ respectively, using 11 successive depth images and Table 4.2 shows the error values, for the case of Laplace noise with $b = 0.6\,[m.u.]$ (model units). The abbreviations of the different versions of the proposed method are described in Table 4.1.

We observe that in the case of joint depth and confidence (biconvex) estimation problem, ACS and AMA algorithms give equivalent results in practice, with ACS marginally

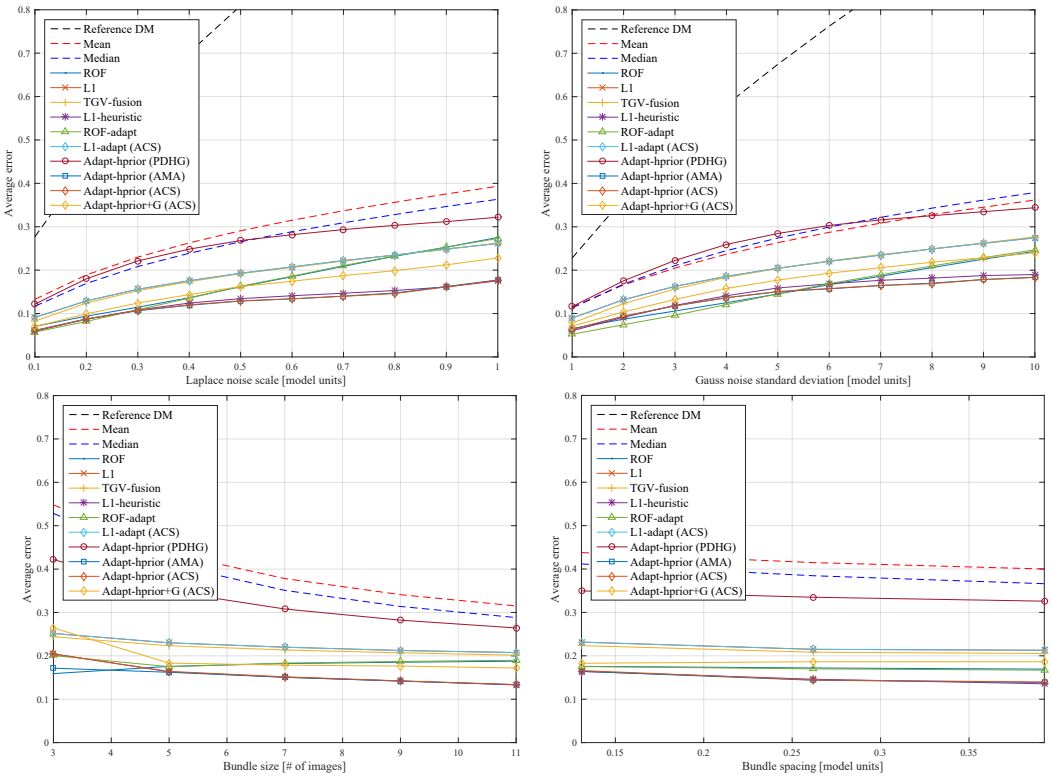


Figure 4.3: Average depth error in relation to (from top left to bottom right): a) Laplace noise scale; b) standard deviation of the Gauss noise; c) number of fused depth images; d) distance between the fused depth images. Laplace noise scale $b = 0.6\,[m.u.]$.

Table 4.1: Method names

L1-heuristic	confidence based on the scene geometry
ROF-adapt	L_2 fidelity with adaptive confidence values
L1-adapt	L_1 fidelity with adaptive confidence values
Adapt-hprior	L1-adapt with scene geometry based prior
Adapt-hprior+G	as Adapt-hprior plus appearance prior

better in average. For this dataset, PDHG algorithm gives results with errors close to the median and average baselines, as it does not converge numerically. This is possibly caused by the high signal to noise ratio (SNR) in the images of this dataset. We also observe that the *L1-heuristic* version of the model provides better results with respect to the *L1-adapt* version, and almost as good as the other two adaptive versions. This is indicative of the scene geometry confidence values effectiveness.

Additionally, the *Adapt-hprior* version performs better than the extended *Adapth-hprior+G* version. The reason for this is that lower regularization is applied near the image edges, hence noise is not suppressed in these areas. Finally, we see that all the adaptive versions with heuristic priors, as well as the *L1-heuristic* version perform better than the TGV-Fusion method [118], while *L1-adapt* gives similar results.

A visual comparison of the results is presented in Figures 4.8 and 4.9. For this example it is evident that only the *L1-heurisitic* and *Adapt-hprior* give results which are smooth on one hand and close to the ground truth on the other. *Adapt-hprior* actually is more faithful in terms of shape as the numerical results suggest. In all other cases residual high frequency noise can be observed on the surface. This is mainly due to the

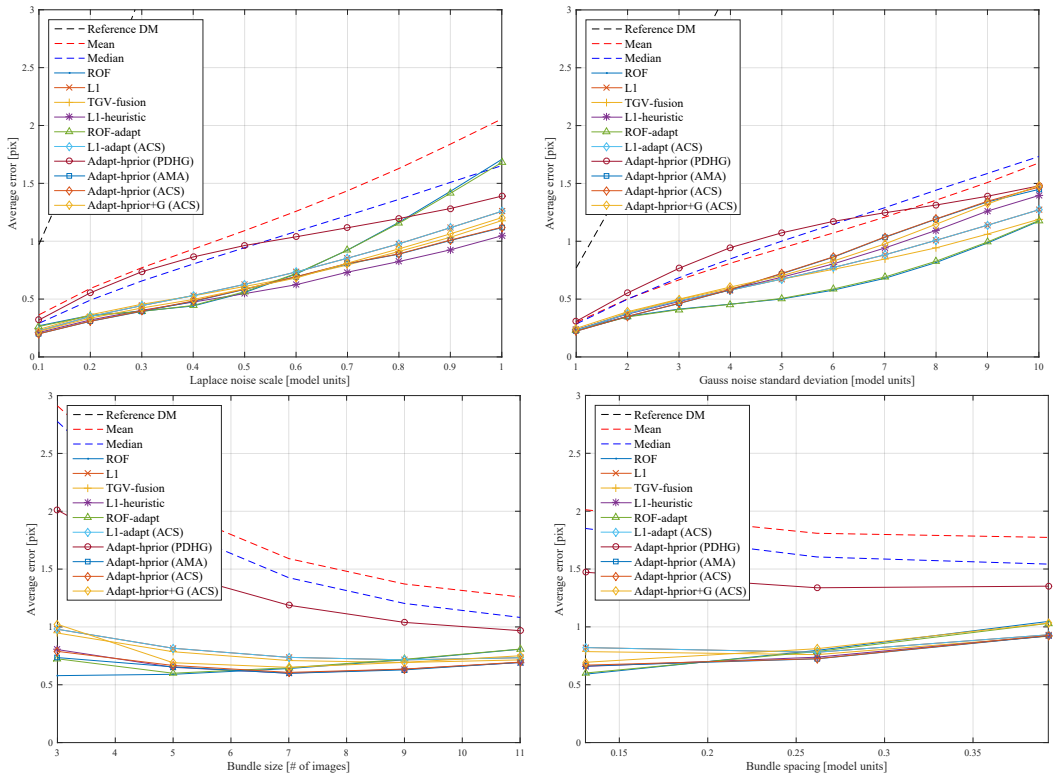


Figure 4.4: Average disparity error in relation to (from top left to bottom right): a) Laplace noise scale; b) standard deviation of the Gauss noise; c) number of fused depth images; d) distance between the fused depth images. Laplace noise scale $b = 0.6\,[m.u.]$.

Table 4.2: Results for the objects dataset for different versions and ablations of the proposed model for Laplace noise of scale $b = 0.6\,[m.u.]$, and 11 fused depth images. (Best values in bold)

	RMSE	ZMAE	NMAE	Z-avg	out-3 [%]	D-avg [px]
Reference DM	0.8152	0.5861	1.5464	0.9040	58.5764	7.1851
Mean	0.1631	0.1265	1.5156	0.3150	6.2219	1.2583
Median	0.1449	0.1100	1.5081	0.2886	3.9030	1.0839
ROF	0.0872	0.0632	1.1461	0.1848	0.4173	0.7162
L1	0.0943	0.0689	1.3754	0.2075	0.6818	0.7340
TGV-fusion [118]	0.0943	0.0685	1.3531	0.2061	0.7312	0.6912
L1-heuristic	0.0831	0.0572	0.5893	0.1410	0.2029	**0.6238**
ROF-adapt	0.0883	0.0643	1.1357	0.1862	0.4236	0.7204
L1-adapt (PDHG)	0.1272	0.0970	1.4892	0.2639	2.1541	0.9467
L1-adapt (AMA)	0.0943	0.0688	1.3756	0.2075	0.6807	0.7339
L1-adapt (ACS)	0.0943	0.0689	1.3754	0.2075	0.6818	0.7340
Adapt-hprior (PDHG)	0.1392	0.1068	1.5026	0.2817	3.1713	1.0383
Adapt-hprior (AMA)	**0.0776**	**0.0537**	0.5754	**0.1338**	0.1153	0.6914
Adapt-hprior (ACS)	0.0778	0.0539	**0.5719**	**0.1338**	**0**	0.6938
Adapt-hprior+G (PDHG)	0.1626	0.1235	1.5137	0.3121	5.8530	1.1945
Adapt-hprior+G (AMA)	0.1205	0.0629	0.6713	0.1720	1.9409	0.6675
Adapt-hprior+G (ACS)	0.1243	0.0645	0.6612	0.1744	2.2225	0.6821

very low SNR of the original depth images. The *L1-adapt* and *TGV-fusion* methods still are able to capture the shape of the surface, however the reconstructed surface is not smooth.

We examine also the relation of accuracy of the fused depth image with the bundle size and the spacing between the original depth images. The results are presented in the bottom row of Figures 4.3 and 4.4. In general one would expect that more data layers would produce more accurate results. This is confirmed up to a certain point for the disparity error, while for larger bundles the errors increase. This is attributed to the increase of errors in occluded regions resulting by the reprojection of distant depth images. This also evident in the disparity error results in the last column of Figures 4.3 and 4.4. Hence, more depth images are useful for decreasing the error as long as they are close to the reference view point, while more distant images tend to introduce errors as scenes are not consistent any more in the occluded regions. Average depth error slightly improves in both these cases instead. A closer examination reveals that the actual depth error increases, while the normal estimation error decreases and this positively affects the average. The decrease in normal errors is reasonable since the scene is captured from a wider view-point range hence their estimation is more robust. These observations can be used to determine the best size of the bundle based on the camera motion, however we will not treat this problem here as it is outside the scope of this work.

Urban Landscapes We performed the same set of experiments for the urban landscapes dataset. The intrinsic parameters of the virtual camera are the same, however the camera here follows a purely translational path, facing always the scene from above. The distance of the camera from the zero level of the scene is taken equal to $300\,[m.u.]$, and depth images are generated every $4\,[m.u.]$ forming a bundle of 11 images.

The top row of Figures 4.5 and 4.6 show the effect of Laplace and normally distributed noise on the final results.

Table 4.3: Results for the urban landscapes dataset for different versions and ablations of the proposed model for Laplace noise of scale $b = 6\,[m.u.]$. (Best values in bold)

	RMSE	ZMAE	NMAE	Z-avg	out-3[%]	D-avg [px]
Reference DM	8.4952	6.0155	1.4727	4.2221	0	0.1863
Mean	3.4925	2.1655	1.3122	2.1490	0	0.0973
Median	3.1017	1.7001	1.2664	1.8832	0	0.0699
ROF	3.3454	2.0385	1.2821	2.0601	0	0.0950
L1	2.4370	1.2554	1.0936	1.4957	0	0.0622
TGV-fusion [118]	2.4630	1.2654	1.0905	1.5035	0	0.0626
L1-heuristic	**1.5670**	0.5490	0.3168	0.6484	0	0.0565
ROF-adapt	1.7434	0.7264	0.5809	0.9027	0	0.0619
L1-adapt (PDHG)	1.5768	**0.4582**	**0.1647**	**0.4919**	0	0.0562
L1-adapt (AMA)	2.3775	1.1960	1.0502	1.4401	0	0.0612
L1-adapt (ACS)	1.7316	0.5385	0.2851	0.6430	0	0.0591
Adapt-hprior (PDHG)	1.5718	0.4874	0.1693	0.5062	0	0.0585
Adapt-hprior (AMA)	1.7229	0.6022	0.3091	0.6845	0	0.0548
Adapt-hprior (ACS)	1.7201	0.5694	0.2050	0.5855	0	**0.0528**
Adapt-hprior+G (PDHG)	1.9403	0.6254	0.3591	0.7582	0	0.0573
Adapt-hprior+G (AMA)	2.3256	1.1123	0.9194	1.3348	0	0.0618
Adapt-hprior+G (ACS)	1.9872	0.7723	0.5470	0.9434	0	0.0565

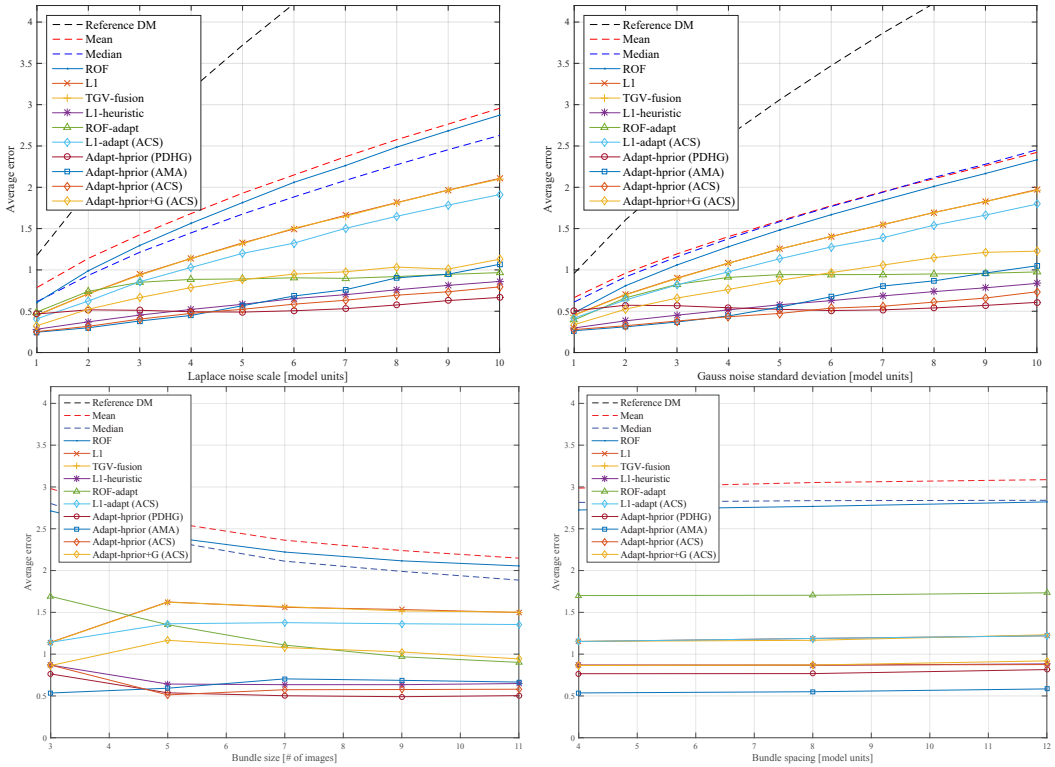


Figure 4.5: Average depth error in relation to (from top left to bottom right): a) Laplace noise scale; b) standard deviation of the Gauss noise; c) number of fused depth images; d) distance between the fused depth images. Laplace noise scale $b = 6\,[m.u.]$.

Table 4.3 shows the actual error values for the case of Laplace noise with $b = 6\,[m.u.]$. We observe also here that in the case of joint depth and confidence (biconvex) estimation problem, ACS gives better results with respect to AMA. In contrast to the previous dataset, we observe here that the PDHG versions of the adaptive methods always converged providing better results with respect to ACS and AMA methods. Nevertheless, ACS algorithm still gives results with similar errors. It is interesting to see that also for this dataset the *L1-heuristic* version give satisfactory results. Moreover, the *L1-adapt* version gives good results with respect to the methods which use prior confidence. This is important, especially considering that the heuristic priors explicitly use knowledge about the problem, and it highlights the power of the adaptive methods to infer suitable confidence values from the data.

A visual comparison of the results is presented in Figure 4.10. We see that the adaptive versions of the proposed model gives the best results. The results of this dataset better highlight the contribution of the automatically estimated confidence values. The difference with respect to the previous dataset, lies mostly in the ratio between the noise scale and the distance from the object.

Finally, the second row of Figures 4.5 and 4.6 show the effect of the bundle size and spacing on the fusion result for the urban scenes dataset. We see that the observations made for the objects dataset remain valid also here.

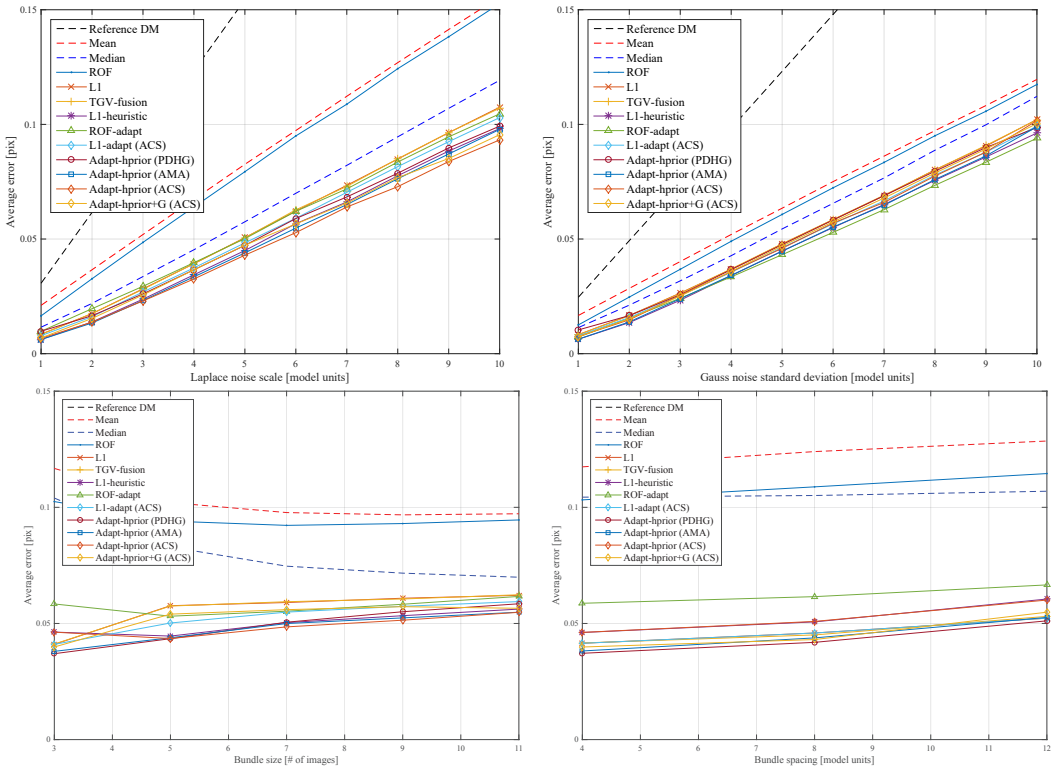


Figure 4.6: Average disparity error in relation to (from top left to bottom right): a) Laplace noise scale; b) standard deviation of the Gauss noise; c) number of fused depth images; d) distance between the fused depth images. Laplace noise scale $b = 6\,[m.u.]$.

Real data

We evaluated the performance of our model for the depth fusion on real data using the KITTI dataset [131]. Here, ground truth of the disparity and calibration data of the cameras are provided, while ground truth localization data are not given. To estimate the camera motion, we considered two different stereo-camera localization methods in order to recover the relative transformations between the reference and the other views. The first is based on [5], and the second is the one used in [6] for the localization of a head-mounted stereo-camera.

The dataset contains stereo-pairs of images hence depth images from each of these stereo-pairs have to be computed. We have considered two methods for computing the depth images. The first is semi-global matching (SGM) algorithm [133], while the other is the ELAS method [7]. As our method assumes that the depth maps are given as-is, the quality of the result depends on the quality of the original depth images, hence the results presented here should be compared to the results of the stereo evaluation of the respective methods. The results regarding the non-occluded areas are presented in Table 4.4 and in Table 4.5 for the different choices of localization and disparity estimation algorithms evaluated for the training set of the KITTI stereo benchmark. The average values reported here are arithmetic averages in order to be consistent with the values reported on the website of the KITTI benchmark. One can notice that the proposed model performs better in all combinations apart from combination [6] & [7]. This suggests that the proposed model is robust with respect to registration errors. The largest improvement in the out-3 metric with respect to the single view disparity estimation is equal to $4.25\,[\%]$ and it is observed for the combination [5] & [7].

Table 4.4: Results for KITTI stereo benchmark training set with localization according to [5].

	SGM [133]			VISO [7]		
	density [%]	out-3 [%]	D-avg [px]	density [%]	out-3 [%]	D-avg [px]
Reference	84.6221	12.6218	3.0169	93.4506	11.5387	2.0531
Mean	98.7363	12.7838	2.5826	99.6008	13.6108	2.0172
Median	98.7361	9.0966	2.1139	99.6008	7.6852	1.4663
TGV-fusion	100	8.6929	2.0184	100	7.4690	1.4149
L1-heuristic	100	8.6780	1.9994	100	7.3058	1.3741
Adapt-hprior (ACS)	100	**8.6466**	**1.9941**	100	**7.2947**	**1.3696**

Table 4.5: Results for KITTI stereo benchmark training set with localization according to [6].

	SGM [133]			VISO [7]		
	density [%]	out-3 [%]	D-avg [px]	density [%]	out-3 [%]	D-avg [px]
Reference	84.6186	12.6285	3.0334	93.4459	11.5412	2.0516
Mean	98.7740	13.1046	2.6234	99.6015	13.9590	2.0603
Median	98.7739	9.4853	2.1547	99.6015	7.9815	1.5191
TGV-fusion	100	9.0621	2.0516	100	7.7581	1.4665
L1-heuristic	100	9.0451	2.0333	100	**7.6057**	**1.4268**
Adapt-hprior (ACS)	100	**9.0162**	**2.0281**	100	7.8962	1.4834

Figure 4.7: Fused depth images for the KITTI dataset.

We used the *Adapt-hprior* versions, the best performing version of our method, to compute the disparity images of the testing set of the KITTI stereo benchmark, using the combination [5] & [7] for localization and single view disparity estimation, respectively. The results obtained are presented in Table 4.6. We can see that the results improve by 1.78% with respect to the single-pair disparity estimation algorithm in the out-noc-3 metric, and by 3.07% with respect to out-all-3.

Finally, we repeated the evaluation by computing individual disparity maps using [8], which corresponds to the current state of the art. The results are presented in Table 4.7, while the complete results are available under the short name *cfusion* on the KITTI benchmark's website. We note that the proposed fusion model is able to further increase the accuracy of the disparity maps. Considering also the occluded regions of the reference image, our model achieves better results with respect to all competing methods

Table 4.6: Results for KITTI stereo benchmark testing set with localization according to [5] and comparison to the single view results of [7].

	density [%]	out-noc-3 [%]	out-all-3 [%]	avg-noc [px]	avg-all [px]
Reference [7]	94.55	8.24	9.96	1.4	1.6
Adapt-hprior (ACS)	**99.70**	**6.46**	**6.89**	**1.2**	**1.3**

Table 4.7: Results for KITTI stereo benchmark testing set with localization according to [5] and comparison to the single view results of [8].

	density [%]	out-noc-3 [%]	out-all-3 [%]	avg-noc [px]	avg-all [px]
Reference [8]	**100**	2.61	3.84	**0.8**	1.0
Adapt-hprior (ACS)	99.93	**2.46**	**2.69**	**0.8**	**0.8**
Reference - Reflective [8]	-	18.45	21.96	3.5	4.3
Adapt-hprior (ACS) - Reflective	-	**15.31**	**16.20**	**2.6**	**2.8**

on the dataset, by the time of submission of this manuscript. The improvement on the reflective regions of the images is even more significant, where the accuracy improves by 3.14% in the out-noc-3 metric, and by 5.76% in the out-all-3 metric, with respect to [8]. Examples of fused depth images for this evaluation are presented in Figure 4.7.

4.6 Conclusions

We introduce a novel model for data fusion with spatially varying confidence values. The proposed model directly estimates the confidence values from the given data. We have proved the main properties of this model and also discussed methods to estimate optimal solution. Indeed, an optimal solution for this family of models can be estimated by solving a biconvex non-smooth optimization problem. We presented two algorithms for solving the biconvex optimization problem, corresponding to the ACS, AMA, and PDHG classes of algorithms, discussing their convergence to critical points. We also discuss possible ablations of the proposed model, and focus on the possibility to assign *a-priori* confidence values.

We demonstrated numerically the behavior of the proposed model for synthetic images and we evaluated its performance considering the fusion of depth images as application. The results show that model outperforms the considered baselines and state of the art algorithms for this problem. We also examined the performance of various ablations of the full model. Moreover, we have seen that for the case of depth image fusion, spatially varying confidence values estimated from the geometry of the scene can provide satisfactory results.

As future work on the theoretical front we shall examine the PDHG algorithm for biconvex problems and its convergence. On the application side we shall examine closer TV regularization on manifolds for 3D modeling as in [108] and [109] and study the consistency and coherence of surfaces generated from images.

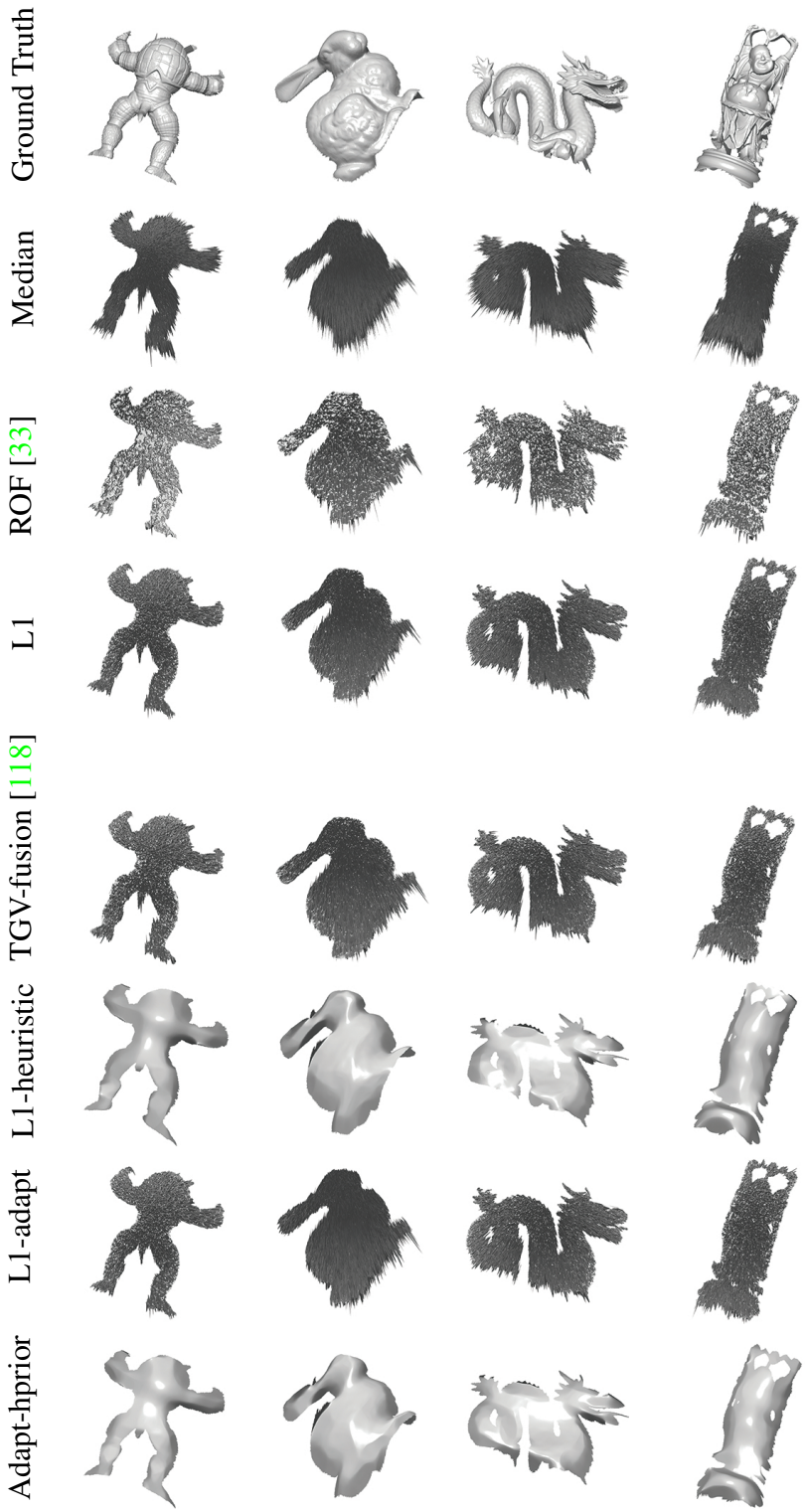


Figure 4.8: Surfaces obtained by different methods for the Stanford 3D scanning dataset [1, 2, 3].

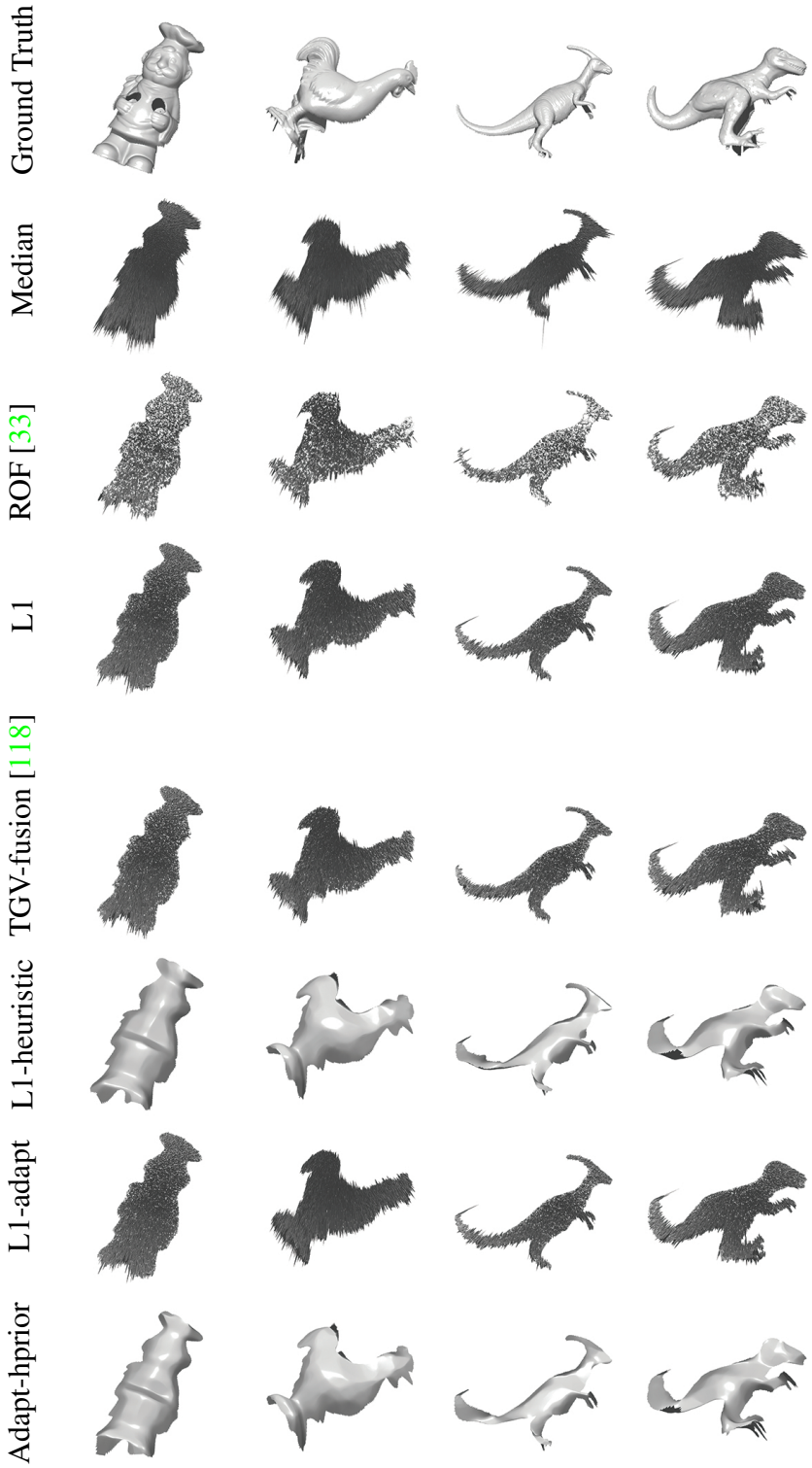


Figure 4.9: Surfaces obtained by different methods for the dataset of [4].

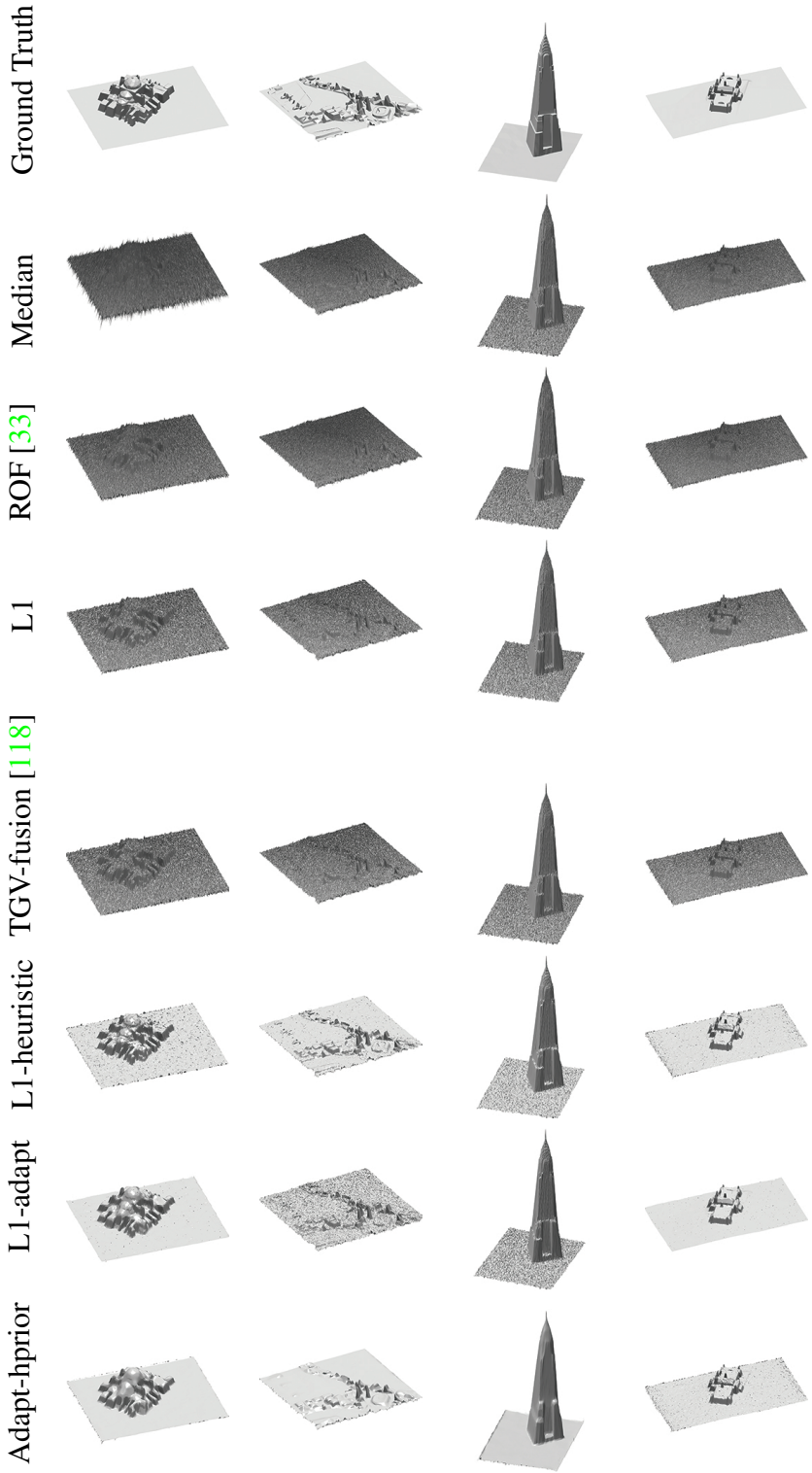


Figure 4.10: Surfaces obtained by different methods for the Urban Landscapes dataset.

Chapter 5

Action recognition

In this Chapter we present a novel approach to human action recognition, with motion capture data (MoCap), based on grouping sub-body parts. By representing configurations of actions as manifolds, joint positions are mapped on a subspace via principal geodesic analysis. The reduced space is still highly informative and allows for classification based on a non-parametric Bayesian approach, generating behaviors for each sub-body part. Having partitioned the set of joints, poses relative to a sub-body part are exchangeable, given a specified prior and can elicit, in principle, infinite behaviors. The generation of these behaviors is specified by a Dirichlet process mixture. We show with several experiments that the recognition gives very promising results, outperforming methods requiring temporal alignment.

5.1 Introduction

Human action recognition is still a challenging and stimulating problem especially when considering motion capture data (MoCap), which are relevant in several applications including robotics, sports, rehabilitation and entertainment. A considerable amount of work has been proposed so far to solve problems arising in action recognition, such as view-point change, occlusions, likewise variations in behaviors amid different subjects performing the same action. However there is a significant difference between MoCap and 2D/2.5D action representations, and it could be argued without fear that the two recognition problems are drastically different, as they address different feature spaces and representations and, consequently, different recognition methods. MoCap sequences represent actions by 3D points, and joints of the human skeleton with appropriate kinematics. These data can, for example, be acquired by means of an RGB-D sensor, such as the Kinect, by infrared marker tracking systems, such as the Vicon System, or via back-projection techniques using multiple cameras. With this kind of data, occlusions so far have not been considered a major issue, such as with 2D/2.5 D data, however variations amid behaviors are still a major problem to be handled. Among the most relevant approaches we recall [134, 135, 136, 137, 138], all using noise and occlusion free datasets. In [134] actions are represented as structured-time series, with each frame lying on a high-dimensional ambient space, from which a spatio-temporal manifold is obtained by a dimensionality reduction approach, based on dynamic manifold warping, accounting only for joints translation. In [139], instead, both joints rotations and translations are considered, so as to construct a novel class of features in $SE(3) \times \cdots \times SE(3)$, obtaining a full feature space mapped on the Lie algebra. In [136]

actions are represented via joint covariance descriptors, so as to work with symmetric positive definite matrices, which lie on Riemannian manifolds. In most approaches the representation of the joints space is a major issue, and the need for a viable compromise between space reduction and completeness seems evident. In this sense we propose a novel representation for MoCap data, by introducing a new skeleton model, which has the advantage of considering the ambient space of the joints and mapping it into a reduced space via Principal Geodesic Analysis. The advantage of the proposed representation is that it keeps the most from the joints information and, at the same time, it provides the most suitable transformation to approach the recognition problem with a non-parametric Bayesian model.

Indeed, the representation model is crucial, both for eliciting features and for the recognition method used. For example, [134, 139, 135] consider a time-based ordering for which a temporal alignment is needed. In particular, [135] decompose the 3D joints into subspaces representing either the motion of a single body part, or of the combination of multiple ones. In our approach, instead, for each joint of the skeleton, and for each configuration in the action space, we keep the global transformation of the joint reference frame with respect to the world inertial frame. These transformation matrices are elements of a Riemannian manifold, and joints of the human skeleton have ranges of variation, which can be gathered into groups. In particular, we consider 6 sub-body groups, corresponding to the head, left and right legs, torso, left and right arms, respectively. Each of these defined groups represents a set of possible motions of the associated sub-body part, and it is such that the elements in the set are order independent and exchangeable, making unnecessary the temporal alignment, as for example proposed in [134, 140, 139]. We provide a representation for these groups via the principal directions of each of them, in the configuration space. The obtained feature space proves to be good for classification, based on clustering. The basic idea is that every type of action generates a specific set of behaviors for each sub-body part. To capture similarities among behaviors we approach the classification problem with the Dirichlet process mixture model. Other approaches considering behaviors classification are [137, 138, 141]. In [137], the most informative joints are extracted by considering the fastest joints or the joints that mostly vary in angles. Similarly, [138] construct an actionlet ensemble, which is a collection of the most discriminative primitive actions, which in turn are the representative features of subsets of joints of an action sequence. These actionlets are learned within the SVM framework. [141] introduce eigenjoints as novel features so as to represent an action as the set of static pose, offsets and joints motion. Many approaches use datasets like [142, 143, 144, 145], which consider only 3D joints locations. Our approach, requiring full 3D poses, can be applied to these datasets too. In fact, following [139] the root joint (see Figure 5.1) can be simply considered as translated with respect to the world origin, without rotations, and each other joint rotation matrix can be evaluated as the minimum rotation required to carry the world x-axis onto the joint bone.

The advantage of our approach is that behaviors are generated by Dirichlet process mixtures, exhibiting a great flexibility, and performing well both with queries formed by a single frame and with queries formed by a set of frames which do not need to be ordered, in so showing to be robust with respect to frame occlusions, action interruptions, and looping repetitions. Indeed, the great benefit of the proposed method, called PGA-DPM, is that it provides a simple representation for basic actions, which is very suitable for learning. It can be used to generalize the recognition problem when time and

subsequence relations are effectively needed to define complex actions, by combining different basic actions.

The chapter is organized in the following manner. In Section 5.2, we focus on some preliminary definitions and methods that will be used to define the feature space. How groups of joints are obtained by collecting these features into groups, according to the limbs of the human skeleton, is explained in Section 5.3. In Section 5.4, we introduce the classification model based on Dirichlet process mixtures generating a representation of an action, which can possibly exploit some empirical knowledge of the action itself. In Section 5.5 results are presented, and a comparison with a state of the art method (the Dynamic Manifold Warping, [140, 134]) is proposed. Finally, in Section 5.6, we address some future developments together with some conclusive discussion.

5.2 Background

In this preliminary part we provide some basic notions that are used for the feature space representation, for further details on the basic concepts we refer the reader to [146, 147]. In the following, vectors are denoted by boldface symbols and matrices by upper case letters. We start considering the set of transformations T in $SE(n)$, $n = 3$:

$$T = \begin{bmatrix} R & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{5.1}$$

Here $R \in SO(3)$ is the rotation matrix, and $\mathbf{d} \in \mathbb{R}^3$ is the translation vector. $T \in SE(3)$ has 6 DOF and is used to describe the pose of the moving body with respect to the world inertial frame. $SO(3)$ and $SE(3)$ are Lie groups and their identity elements are the 3×3 and 4×4 identity matrices, respectively. The *tangent space* of a Lie Group at its identity element defines its *Lie algebra*. The Lie algebra $so(3)$ of $SO(3)$ is formed by skew-symmetric matrices of the form:

$$so(3) = \{\Omega \,|\Omega \in \mathbb{R}^{3\times 3}, \Omega = -\Omega^\top\}. \tag{5.2}$$

Ω can be uniquely identified with a vector $\boldsymbol{w} \in \mathbb{R}^3$. The Lie algebra $se(3)$ for $SE(3)$ is defined as follows:

$$se(3) = \left\{ \begin{bmatrix} \Omega & \mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix} \middle| \Omega \in so(3), \mathbf{v} \in \mathbb{R}^3 \right\}. \tag{5.3}$$

Given an element $U \in se(3)$ on the tangent space $\mathcal{T}_I SE(3)$ at the identity I of $SE(3)$, the corresponding element $T \in SE(3)$ can be evaluated just by using the exponential map: $\exp : se(3) \to SE(3)$, where $\exp$ in $SE(3)$ is the matrix exponential. The inverse mapping is $\log : SE(3) \to se(3)$, where $\log$ in $SE(3)$ is the principal matrix logarithm. The same mappings hold when restricting to $SO(3)$. Elements of $se(3)$ can be associated with the tangent vector of a curve $A(t) \in SE(3)$, at t, representing the local motion of a rigid body. Elements of this kind are called *twists*, and can be uniquely represented by a 6-dimensional vector $(\boldsymbol{\omega}(t)^\top, \mathbf{v}(t)^\top)^\top$, physically corresponding to the *instantaneous* angular velocity and the *instantaneous* linear velocity of the body, both expressed in the moving body reference frame. The operation $(\cdot)^\vee$ converts a 4×4 twist into the 6 dimensional vector $(\boldsymbol{\omega}(t)^\top, \mathbf{v}(t)^\top)^\top$.

Given a metric specifying properties of the rigid body, [148] show that a geodesic is a locally length-minimizing curve on a manifold, such that, for two configurations $A, B \in SE(3)$:

$$A = \begin{bmatrix} R_A & \mathbf{d}_A \\ \mathbf{0} & 1 \end{bmatrix}, \quad B = \begin{bmatrix} R_B & \mathbf{d}_B \\ \mathbf{0} & 1 \end{bmatrix} \tag{5.4}$$

the geodesic $\Gamma(t)$ is:

$$\Gamma(t) = \begin{bmatrix} R_A \exp(\Omega_0 t) & (\mathbf{d}_B - \mathbf{d}_A)t + \mathbf{d}_A \\ \mathbf{0} & 1 \end{bmatrix}. \tag{5.5}$$

Here $\Omega_0 = \log(R_A^\top R_B)$. The problem to solve in this preliminary part is the following: given a set of Euclidean transformations $T_1, ..., T_n \in SE(3)$, find the principal directions maximizing the variance of the data. This can be obtained by applying the Principal Geodesic Analysis (PGA) introduced for the first time in [149], which is a generalization of PCA when a manifold is considered. The authors define the variance, the subspaces and the projections in a manifold setting. In particular, the subspaces, that in PCA were linear, now are *geodesic sub-manifolds*. An extension of the algorithm provided in [149] to $SE(3)$ is straightforward and illustrated in Algorithm 5.1. Indeed, given the set of body transformations, the centroid $\bar{T}$ is computed, so as to minimize the distance of $\bar{T}$ with all the Ts in the starting set. If the Ts are close enough to each other, it is known that the centroid is unique as stated in [150, 151]. This is the intrinsic mean on the manifold, a generalization to $SE(3)$ is straightforward.

Algorithm 5.1: Principal Geodesic Analysis in $SE(3)$

Data: $T_1, ..., T_n \in SE(3)$

Result: Principal directions $\mathbf{e}_i \in \mathcal{T}_\mu SE(3)$ (tangent space of $SE3$ at μ) with associated variances $\lambda_i \in \mathbb{R}$

1 Compute $\mu = [\bar{R}\,|\bar{d}]$ with $\bar{R}$ Karcher Mean in $SO(3)$ [150] and $\bar{\mathbf{d}} = 1/n \sum_i \mathbf{d}_i$ on $T_1, ..., T_n$;

2 Compute $\Gamma_{\mu,T_i}(t)$, $t \in [0, 1]$ as in eq.(5.5) with R_A replaced by $\bar{R}$ and R_B replaced by R_i, obtained from T_i, $i = 1, \dots n$ (eq. (5.1));

3 $\forall T_i$ compute the twist $U_i = \Gamma^{-1}_{\mu,T_i}(t)\dot{\Gamma}_{\mu,T_i}(t)$, $t \in [0, 1]$;

4 Compute the vector $(\boldsymbol{\omega}(t)^\top, \mathbf{v}(t)^\top)_i^\top = U_i^\vee$;

5 $S = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{\omega}(t)^\top, \mathbf{v}(t)^\top)_i^\top (\boldsymbol{\omega}(t)^\top, \mathbf{v}(t)^\top)_i$;

6 $\{\lambda_i, \mathbf{e}_i\}$ = eigenvalues and eigenvectors of S;

Fact: The twist U_i physically interprets the local motion of a joint, and using its vector representation $(\boldsymbol{\omega}(t)^\top, \mathbf{v}(t)^\top)_i$ we obtain that S is in $\mathbb{R}^{6\times6}$ and clearly symmetric. Each principal direction $\mathbf{e}_i$, resulting from the PGA algorithm, as an eigenvector of S is in $\mathbb{R}^6$. As Γ_{μ,T_i} is a geodesic, the product $(\Gamma^{-1}_{\mu,T_i}\dot{\Gamma}_{\mu,T_i})$, once applied the $\vee$ transformation, according to the fact that a twist can be uniquely represented by a 6-dimensional vector, specifies the motion between the joint and the Karcher mean $\bar{R}$.

5.3 Action Representation Model

In MoCap representation, input data are sequences of joints configurations. Each sequence is about a single subject performing a specific action. Joints are associated with a subject skeleton and are expressed along time as transformation matrices, of the form given in eq. (5.1), with respect to the global coordinate system. We consider $K = 19$ joints, see Figure 5.1, left. To properly obtain a representation for each sub-body part we introduce some notation.

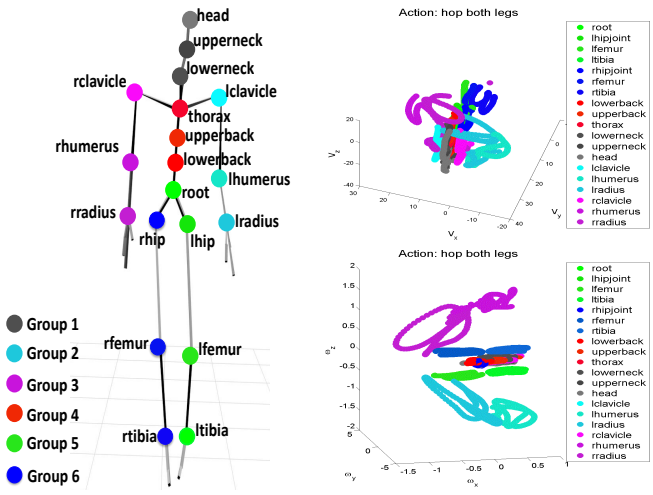


Figure 5.1: On the left a skeleton with the whole set of joints; groups are highlighted by color. On the right joints motion with respect to $\mathbf{v}$ and $\boldsymbol{\omega}$ highlighting motion similarities within groups (better seen in color).

Notation In the following we denote j_i an unordered sequence of frames of the action A_i, which we call *sample sequence*. The length of each sample sequence j_i is denoted by L_{j_i}. Given N_i sample sequences for action A_i, $j_i = 1_i, \ldots, N_i$, their length is $L_{1_i}, \ldots, L_{N_i}$. Each sample sequence is divided in 6 groups, indexed by m. A feature vector of a number of sample sequences for action A_i is $v^l_{j_i,m}$, where $m = 1, \ldots, 6$, $j_i = 1_i, \ldots, N_i$, and the superscript l varies on the sequence length.

D_{j_i} denotes the block matrix for the MoCap joints transformations, for each sample sequence j_i:

$$D_{j_i} = \begin{bmatrix} T^1_{j_i,1} & T^1_{j_i,2} & \cdots & T^1_{j_i,K} \\ \vdots & \vdots & \vdots & \vdots \\ T^{L_{j_i}}_{j_i,1} & T^{L_{j_i}}_{j_i,2} & \cdots & T^{L_{j_i}}_{j_i,K} \end{bmatrix}. \tag{5.6}$$

Here each block $T^l_{j_i,k}$, $k = 1, \ldots, K$ is a 4×4 transformation matrix (see eq. (5.1)) with respect to the world's inertial frame of the sample sequence j_i of action A_i, relative to the k-th joint in frame l.

C_{j_i} denotes the block matrix of all configurations of a single sample sequence j_i of action A_i, taking into account all 6 sub-body groups:

$$C_{j_i} = \begin{bmatrix} g^1_{j_i,1} & \cdots & g^1_{j_i,6} \\ \vdots & \vdots & \vdots \\ g^{L_{j_i}}_{j_i,1} & \cdots & g^{L_{j_i}}_{j_i,6} \end{bmatrix}. \tag{5.7}$$

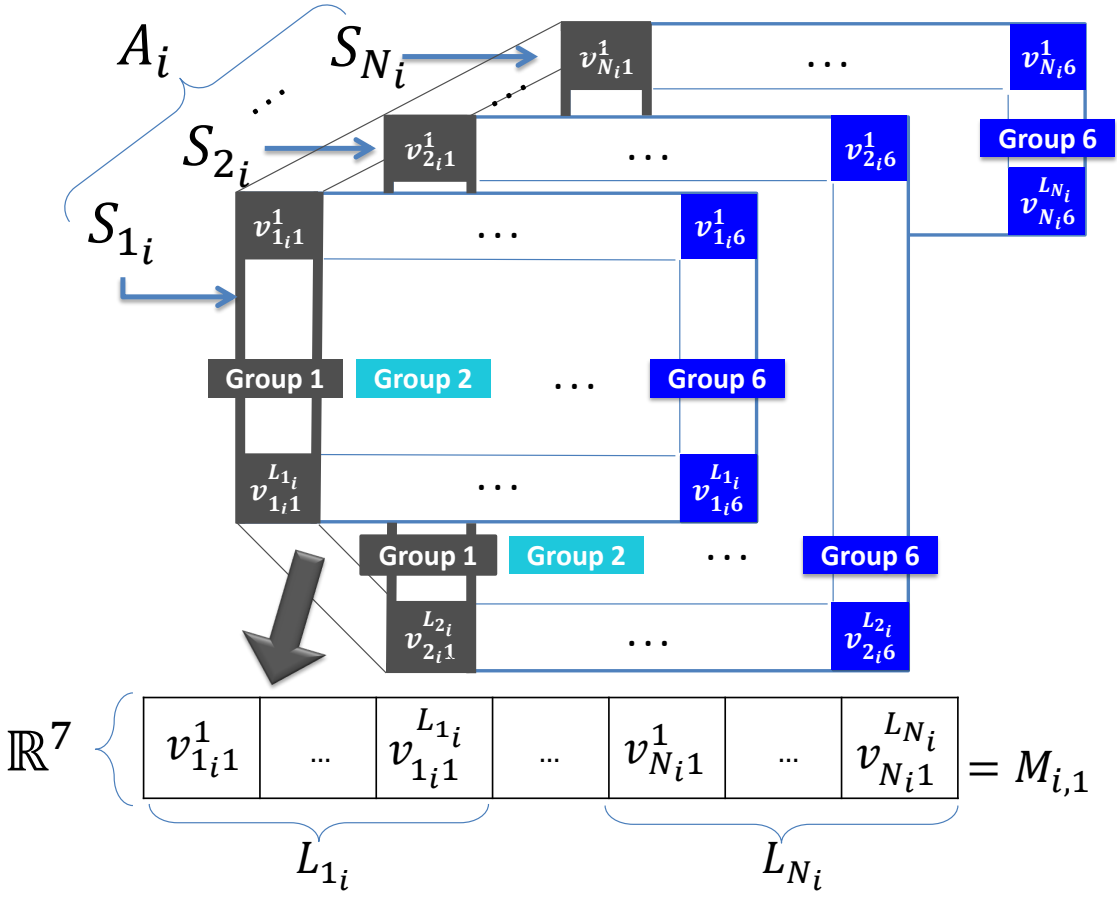


Figure 5.2: Stack of feature vectors $\mathbf{v}^l_{j_i,m}$ of the first group ($m = 1$) of joints into a $7 \times (L_{1_i} + L_{2_i} + \cdots + L_{N_i})$ matrix.

Here each $g^l_{j_i,m}$ is a block of the form $(T^l_{j_i,a}, \ldots, T^l_{j_i,b})$, of dimension $(4 \times 4) \cdot h$, with h the number of joints of the m-th sub-body group, for $m = 1, \ldots, 6$ and $1 \leqslant a < b \leqslant K$.

Matrices like C_{j_i} are used to compute the features of sample sequences of action A_i, as shown in Algorithm 5.2.

5.4 Classification via preferences on DPM

In this section we present the classification approach used, in so making more clear the reasons behind the choice of the data structure, illustrated in Figure 5.1 and explained in Section 5.3. Note that in this section, subscripts and superscripts are, in general, different from those used in the previous section, to simplify the notation. Given a domain $\mathcal{X} \subset \mathbb{R}^7$, the feature-vector $\mathbf{v}_{ik}$ takes values in $\mathcal{X}$ within a range that depends from the implied sub-body part motion properties. For example, the head has specific limited motions, which differ from those of the torso or the legs. Therefore an action can be specified by a number of behaviors, generated by the body parts involved in the action.

In this section we investigate this concept and show how to model the action classification problem via the popular Dirichlet process mixtures. The approach, in this basic formulation, proves that temporal alignment (see e.g. [139]) can be avoided, in so significantly improving the classification process. This result makes possible further in-

Algorithm 5.2: Features extraction

Data: N_i sample sequences C_{j_i}, as in eq. (5.7), of lengths L_{j_i} for action class A_i

Result: Feature vectors of action A_i organized into matrices $\{M_{i,m}\}_{m=1,...,6}$

For each block $g^l_{j_i,m}$, of C_{j_i}, compute the first principal direction $\mathbf{e}^l_{j_i,m} \in se(3)$, according to Algorithm 5.1.

Map $\mathbf{e}^l_{j_i,m}$ into a transformation matrix $T^l_{j_i,m} \in SE(3)$, via exponential mapping.

Build the feature vector $\mathbf{v}^l_{j_i,m} \in \mathbb{R}^7$, using the rotation angles and the translation obtained from $T^l_{j_i,m}$, and the norm of the instantaneous linear velocity, obtained from $\mathbf{e}^l_{j_i,m}$.

for $m = 1 : 6$ **do**

$$M_{i,m} = \left[\mathbf{v}^1_{1_i,m}, ..., \mathbf{v}^{L_{1_i}}_{1_i,m}, \ldots, \mathbf{v}^1_{N_i,m}, ..., \mathbf{v}^{L_{N_i}}_{N_i,m}\right];$$

vestigation on temporal relations among behaviors, to study complex actions built from several primitive ones. Here, we consider the matrix M_{is}, collecting feature vectors for a group s, as random variables X_{ik}, indexed by $k = 1, \ldots, L_k$, with L_k the number of configurations of the feature vectors of action A_i available for training, where the subscript for the group s, $s = 1, \ldots, 6$ is made implicit. Hence, the set of variables X_{ik}, for a group s of sub-body parts motions, induced by the configuration of action A_i, has size $7 \times J_K$, namely, each column has the size of a feature-vector $\mathbf{v}_{ik}$.

Let $\mathbf{x}_{ik} = (x_{i1}, \ldots, x_{iJ_k})^\top$ represent the observed response vectors for the s-th group of action A_i, and $y_{ik} \in \{\ell_1, \ldots, \ell_K\}$ the class labels. Recall that the feature vectors specify the principal directions of a group of joints whose rigid motions are referred to a global frame. Therefore within the set of observations for the same group the response vectors are considered an exchangeable sequence, and ordering is irrelevant.

Let $X^m_i = \{(\mathbf{x}_{ik}, y_{ik}) | i = 1, \ldots, n_i, k = 1, \ldots m_k\}$ be the set of all training data for the group s. So the classification problem, for a group, is to classify the unknown action X^{m+1}, given observations from that group of all the actions settled for training. Classification amounts to reporting $p(y_{m+1}|X^m, \mathbf{x}_{i,m+1})$, where $\mathbf{x}_{i,m+1}$ is the action query given as a partial response matrix of J_k configurations for the group s. In order to compare with [140] we shall also consider a configuration sequence.

The probability model that we consider for the classification problem is the popular Dirichlet process (DP) mixtures (DPM) [46, 152]. A DP places a distribution on the space of distribution, generating a distribution on the countable set of mixtures; we consider a set of DPMs, one for each group s, for each action A_i. Let s a group, $k = 1, \ldots, J_k$ the configuration of action A_i, and $\mathcal{N}_7(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ the multivariate normal, with $\boldsymbol{\mu} \in \mathbb{R}^7$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{7\times 7}$:

$$\begin{aligned} &\mathbf{x}_{ik}|\theta_{ik} \sim \mathcal{N}_7(\mathbf{x}_{ik}|\boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \\ &\theta_{ik}|G_s \sim G_s \\ &G_s \sim DP(\alpha H) \end{aligned} \tag{5.8}$$

Here we are assuming that observations are i.i.d sampled from a parametric family, namely a multivariate Gaussian distribution, with parameters θ_{ik}, which are in turns independently sampled from an unknown distribution G_s on which a Dirichlet process $DP(\alpha H)$ is placed where α is the concentration parameter affecting the number of clusters that will be generated, and H is the base distribution, namely, for a subset S of $\mathcal{X}$, $H(A) = E[G_s(A)]$ and typically H is taken to be the conjugate prior of the observation distribution. Here we follow the conjugate approach for the multivariate

normal, by choosing:

$$\begin{array}{lcl} (\Sigma_{ik}|\beta, W) & \sim & \mathcal{W}(\beta, (\beta W^{-1})) \\ (\mu_{ik}|\Sigma_{ik}, \boldsymbol{\nu}, \rho) & \sim & \mathcal{N}(\boldsymbol{\nu}, (\boldsymbol{\rho}\Sigma_{ik})^{-1}) \\ (\mu_{ik}, \Sigma_{ik}) & \sim & \mathcal{NW}(\boldsymbol{\nu}, \boldsymbol{\rho}, \beta, \beta W) \end{array} \tag{5.9}$$

Here $\mathcal{W}$ is the Wishart distribution, with $\beta > 7$ DOF, 7 the dimension of $\mathcal{N}_7(\cdot)$. $\mathcal{NW}$ is the normal Wishart joint prior distribution with $\boldsymbol{\nu}, \boldsymbol{\rho}, \beta, \beta W$ common to all mixture components of the group s. In turn the priors for $\boldsymbol{\nu}$ and $\boldsymbol{\rho}$ are Gaussian and Gamma, while for W and β the priors are the Wishart and Gamma (see [153, 154] for further details).

The unknown distribution is evaluated at observation points, and according to its discreteness generates clusters of observations. Namely, in any sample $\theta_{i1}, \ldots, \theta_{im}$ from G_s there is a positive probability of identical values (see [155, 46]). Then each sample can either be assigned to an existing partition or it can generate a new one. This is regulated by the probabilities $n_h/(\alpha + n - 1)$ and $\alpha/(\alpha + n - 1)$, which induce the Chinese restaurant process (CRP), and the mixing proportion probabilities π_{ik}. Where n_h is the number of elements of the cluster to which the repeated sample θ_{ih} would belong to.

The classification probabilities for each group s is then obtained as:

$$\begin{array}{l} P_i(y_{m+1} = y|\mathbf{x}_{m+1}, X^m) = \\ \int_{\mathcal{X}} p(y_{m+1} = y|\mathbf{x}_{m+1}, X^m, \Theta) p(\Theta|\mathbf{x}_{m+1}, X^m) d\Theta, \end{array} \tag{5.10}$$

with Θ the vector of all parameters in the model. Then using the loss function based on the percentage of correct classifications, the label assigned to each group s is estimated by the maximum a posterior MAP:

$$\hat{y}_{m+1} = \arg\max_y \{p(y_{m+1} = y|\mathbf{x}_{m+1}, X^m\}. \tag{5.11}$$

Inference of the parameters and hyperparameters is obtained for each group by Gibbs sampling and updating them from their posterior distribution as specified above, using the steps for conjugate prior as in [48] and adopting the clever solutions indicated in [154].

Many approaches have highlighted the need to investigate the dependences among data in different groups when these are generated by DPMs, since the work of [156]. In particular, the problem of how to determine clusters of data in the presence of partial exchangeability and unknown partition of the observations has been addressed. A solution has been indicated in [157] via the hierarchical DPM (HDPM), which can discover dependencies, generating shared clusters with different weights but same locations.

In the representation we propose, considering the domain of the sub-body part features, two subgroups might take values in space regions that intersect. Despite this the range are usually different, and also the observations come separated at the source and the groups are known. Therefore, we combine the groups, in terms of the behaviors that are generated by the DPM for each of them, and use the MAP on the combined groups. To this end we define a preference matrix W of size $n_A \times n_G$, with n_A the number of action classes considered and n_G the number of groups. The stochastic matrix W, which will provide the optimal combination for the groups, is a matrix of multinomial variables, evaluated according to a success matrix Q. Each row of Q represents the experiment assigning a success to the group, which provides the best contribution to

characterize the action. This is assessed by assigning a success to the group that has higher concentration parameter, since this is sensible to the number of behaviours, and the fact implies that the group undergoes several changes during the action execution, hence the involved sub-part is more active and characerizes the action. Hence, the successes recorded for the multinomial at Q_{is} are the values α estimated for the DPM of the group. The parameters of W are estimated at the final step of the Gibbs sampling and kept common to all groups estimations. An initialization of Q is provided assigning a success to the group/groups that are considered the more active ones in the action execution, according to a rule of thumb, for example as groups G_5 and G_6 for walking.

A step t is the final Gibbs sampling step for group s, of action A_i, and α is assigned a value for the group, according to the number of behaviors the group generated. Considering that each group is evaluated in turn, for each action i, the following steps are performed, where W_i is the i-th row of W corresponding to the current action observed, and similarly for Q_i, the recording of the experiments. Let κ_s be the prior assigned to the Dirichlet distribution for the group s:

$$W_{is}^{(t)} = \frac{Q_{is}^{(t)} + \kappa_{is} - 1}{n_A + \sum_{s=1}^{6}(Q_{is}^{(t)} - 1)}. \tag{5.12}$$

Then the new mixture is obtained simply as $W^\top F$, where F is the matrix of the DPM distributions computed for each group s. We can note that the final mixture is still a mixture combining the DPMMs for each group, weighting the groups in a way sensible to the number of behaviors elicited by the DPMM. Clearly without a non-parametric approach this last mixing, which so to say meta-evaluates the estimation, would have not been possible.

5.5 Implementation and Experiments

In this section we report experimental results on the performance of the proposed method for MoCap action recognition. The goal of the experiments is to verify the accuracy of the prediction of a new observed action.

Data We consider 11 types of "cut actions" (i.e. a single type of action per sequence) obtained from HDM05 [158], where each cut action is performed by 4 different subjects, and similar types of actions from CMU [159]. Results from [159] are not reported, though they are almost the same, the data being noiseless. The actions considered from [158] are: grab an object from high with right arm (3401 frames), hop with both legs (5941 frames), kick with left leg (3828 frames), kick with right leg (3374 frames), punch with left arm (3144 frames), rotate both arms backward (1632 frames), run on place (139440 frames), sit down on chair (2884 frames), squat (9519 frames), throw an object with right arm (2254 frames), walk (3470 frames). We have also considered the datasets [160, 161, 162, 163], and adapted it to our full 3D model. Despite these datasets are noisier than CMU and HDM05, results are comparable but not reported for lack of space.

Method All data available are structured according to the description provided in Section 5.3, then they are transformed to obtain the PGA features according to the description provided in Section 5.2. We have trained the DPM model as follows. For each

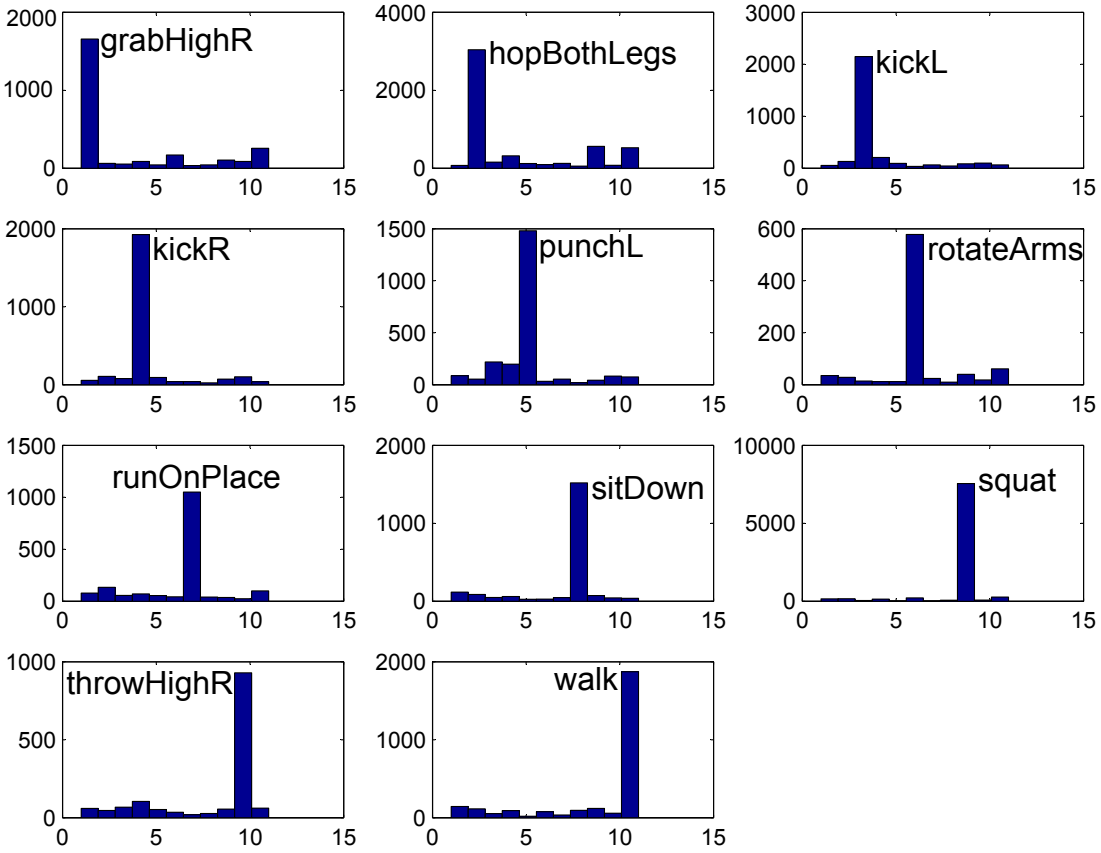


Figure 5.3: Histograms of MAP response for each action category

action we consider 800 data for training. From this set we then define the training set for each group by randomly sampling from the chosen training set. All remaining frames are considered for test. Running the Gibbs sampler we obtain a model for each group of each action and store it into a data structure. We distinguish between a set of frames, randomly chosen from a sequence of frames, in which data are ordered according to the action evolution.

Now, given a set of frames (or an action sequence) from the data test, we first estimate the probability of each group according to the parameters of the model and the mixture components, and then we combine the groups using the estimated weight matrix F, eq. (5.12). The resulting classification is obtained by MAP estimation, eq. (5.11). Estimation of either a set or a sequence of actions takes less than one sec. of computation time. Similarly, geometric transformations and feature computation are on the order of 10^2 sec. On the other hand the computational cost for learning is quite high, of the order of 10^6 sec.

Experiments We have conducted the following experiments. In the first experiment we have tested the test data and verified the MAP on the whole set; this is illustrated in Figure 5.3. Each panel, in the figure, shows the histogram of the classification on the whole test set. We can note that the maximum is always correctly assigned. In the second experiment, given that the number of test data is N, we have randomly sampled from them $N/10 + k$, $k > 10$ data and the results are reported in Figure 5.4.

Finally we have extracted actions as sequences from the test data and the classifi-

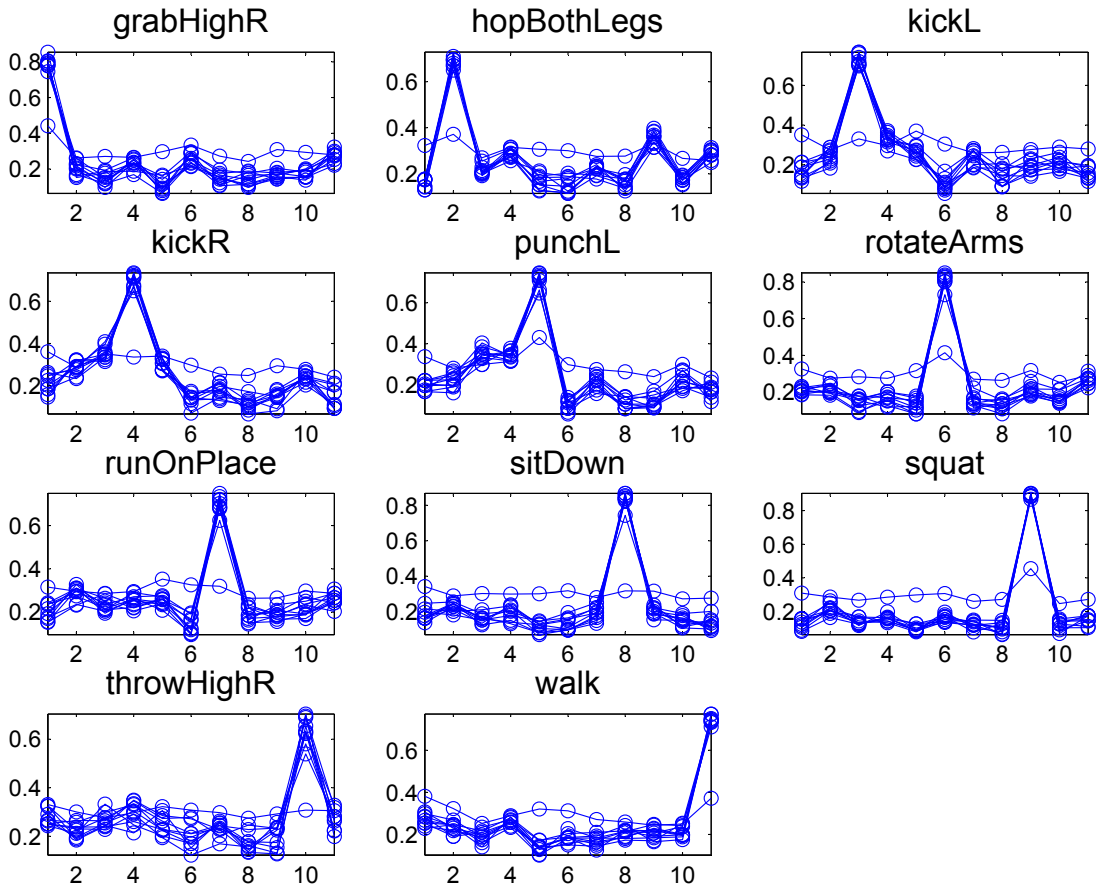


Figure 5.4: MAP evaluation with repeated random samples from test data, for each action category

cation results are reported in the confusion matrix in Figure 5.5, where the results have also been compared with [140].

Comparisons We have chosen the algorithm of Dynamic Manifold Warping [140, 134]. DMW is basically an instance-based learning in which the action sequences are represented as structured time series. The authors, in [140], first temporally align the testing sequence with all the training labeled sequences. They then extract for each aligned sequence frame a similarity measure between the testing sequence and the temporally aligned training sequences, and the action performed in the testing sequence is labeled with the label of the training sequence from which the testing sequence has minimum distance. In our approach, instead, we learn a model so as to estimate the most representative behaviors made by each of the groups of joints, not considering structured sequences along time, but rather considering each feature conditionally independent from the other ones. Therefore, while DMW depends on the sequences considered and for each new input sequence has to compare it with all the labeled training sequences, our algorithm has a learning process so that the testing process is immediate and the accuracy in recognition increases with the number of features considered in the DPM process, following the ”rich get richer” fashion, typical of the DPMs. It is worth mentioning that in order to evaluate the DMW accuracy, we have implemented a version of DMW with a choice of parameters and methods that are hidden in [140].

In order to compare our algorithm with DMW, we have considered 10 configura-

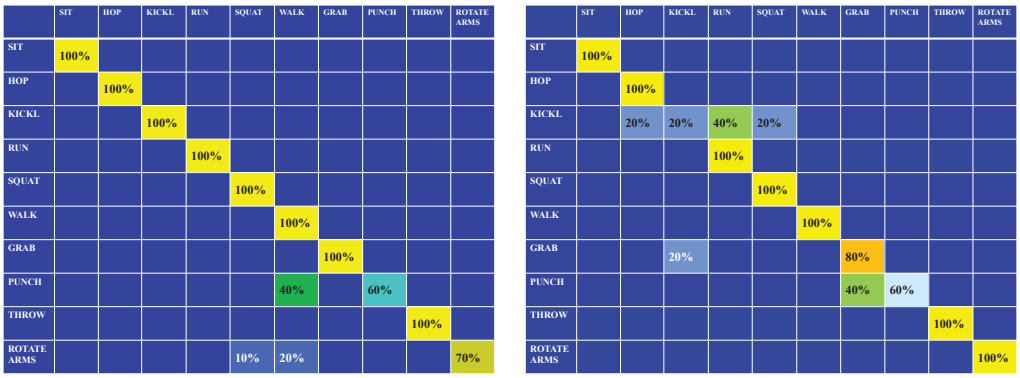


Figure 5.5: Confusion matrices for comparing the PGA-DPM algorithm, on the left, with the one presented in [134], on the right.

Table 5.1: Total Accuracy for the PGA-DPM based method and for DMW [134]

Approach	**Total Accuracy**
PGA-DPM	93.86%
DMW	85.78%

tion sequences of PGA-based features for each action category group. We have used the term configuration sequence, since in our model we do not have ordered data, but instead features that are exchangeable. The tests have been made on 10 actions. In this case, the MAP estimate for our algorithm is computed for each single query frame of a configuration sequence, and the accuracy for each query sequence is evaluated as the percentage of correctly recognized query frames in the query sequence over the total number of frames of that sequence. For DMW, instead, the accuracy is simply the number of sequences correctly recognized, over the total number of sequences. In Figure 5.5, it is possible to see the confusion matrix for our approach and for DMW. In Table 5.1, it is shown for the two approaches the accuracy computed as the total number of recognized query frames over the total number of considered sequences.

Table 5.2: Number of clusters generated for each group for 4 different actions

Action Class	#Clusters Group 1	#Clusters Group 2	#Clusters Group 3	#Clusters Group 4	#Clusters Group 5	#Clusters Group 6
Kick with Left Leg	10	17	14	10	**26**	15
Throw with Left Arm	13	**23**	15	12	19	18
Squat	12	13	13	3	11	13
Walk	8	11	10	4	9	8

Evaluation Table 5.2 shows the number of clusters estimated by the PGA-DPM (as explained in Section 5.4) for each of the sub-body group of joints for 4 different types of actions. Note that in the kick and throw actions, a large number of clusters is estimated for the most representative groups of joints (i.e. the left leg and the left arm, respectively). For the squat and the walk actions, instead, excluding the joints of the torso (group 4), all sub-body groups are involved in the motions, and therefore a more distributed number of clusters is estimated. Furthermore, in Figure 5.6 it is possible to visualize some of the generated clusters for an arbitrary sub-body group in 4 different actions categories: kick with left leg, rotate arms, punch with left arm, walk.

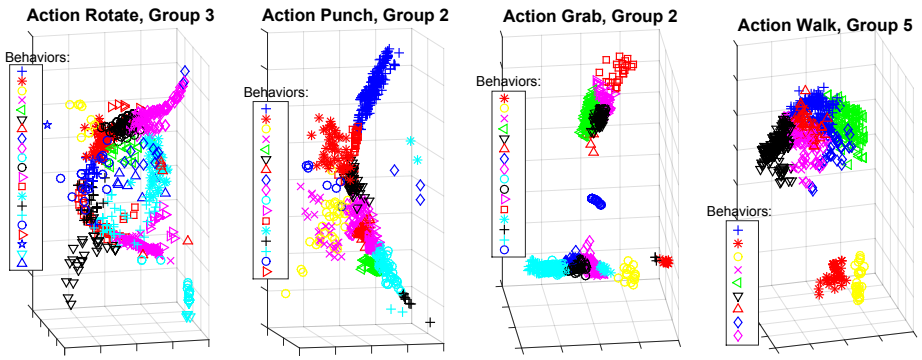


Figure 5.6: Behaviors clustering for 4 sub-body parts of 4 different actions.

5.6 Conclusions

We have presented a novel approach to the human action recognition problem, by considering a new MoCap feature representation, which has been verified to be suitable for developing a non-parametric Bayesian method for classification, via the DPM. In particular, we have combined the skeleton joints into groups and reduced their dimensionality by means of PGA, so as to maintain a solid information on motion. Assuming features to be conditionally independent, for each group, given a specific prior, we have applied DPM to generate the most representative behaviors for each group of joints and each action category so as to perform classification. Our approach proves that a time-ordered representation for MoCap sequences is not needed. and indeed, as shown in Section 5.5, performances are good and our approach outperforms exact time-alignment based approaches as [134]. Basing on these promising results we are now investigating more complex actions, in particular the collaborative ones, in which two different subjects must pass objects between them, and carry objects together.

Chapter 6

Articulated object modeling

In this chapter we discuss a novel framework for modeling articulated objects based on the aspects of their components. By decomposing the object into components, we divide the problem in smaller modeling tasks. After obtaining 3D models for each component aspect by employing a shape deformation paradigm, we merge them together, forming the object components. The final model is obtained by assembling the components using an optimization scheme which fits the respective 3D models to the corresponding apparent contours in a reference pose. The results suggest that our approach can produce realistic 3D models of articulated objects in reasonable time.

6.1 Introduction

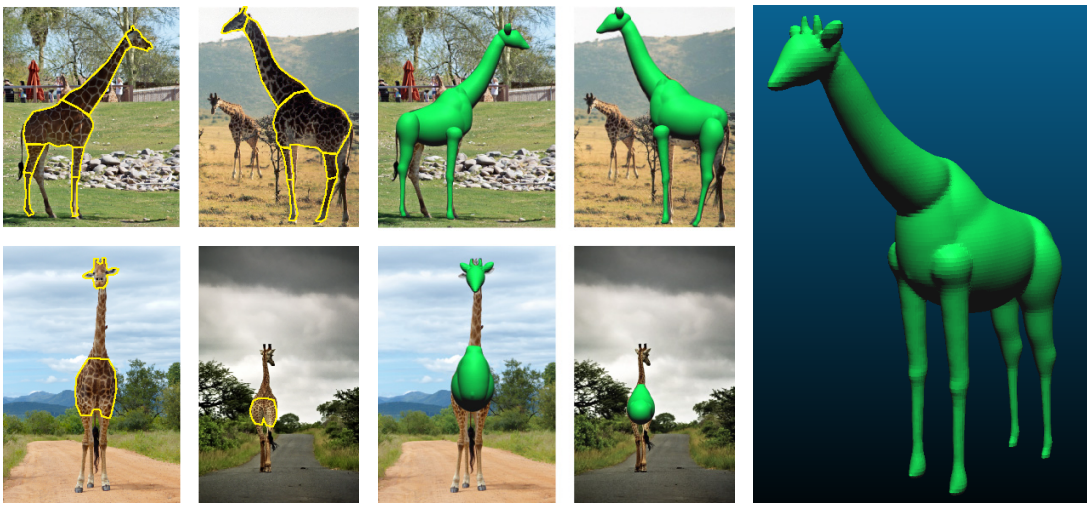


Figure 6.1: **Left:** Images of an animal downloaded from the web overlaid with segmentation masks, **Center:** modeled components overlaid on the input images, **Right:** final 3D model obtained with the proposed method.

The problem of modeling articulated objects, like people, animals and complex human artifacts has a long history in computer vision. Obtaining 3D models of objects from images is essential for many high-level vision tasks. Early approaches suggested a hierarchical composition of the object components, represented as generalized cylinders [164], *geons* [165], or superquadrics [166, 167], just to cite a few well known approaches to the structural descriptions theory. In these early works, components were

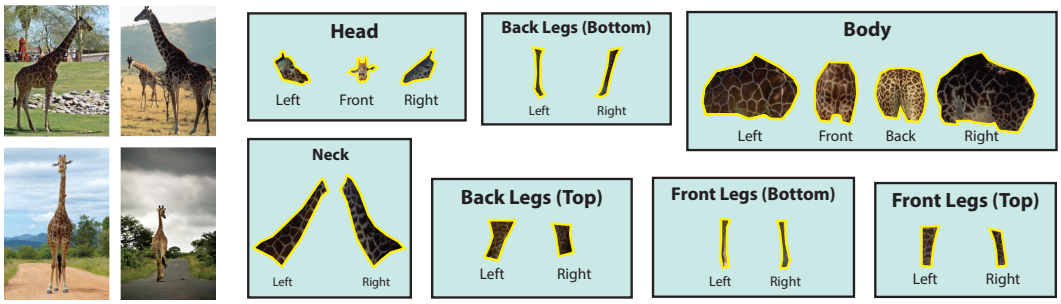


Figure 6.2: **Left:** Input images of Giraffes providing different aspects of each component. **Right:** representative aspects of each component of the Giraffe model.

modeled with parametric 3D shapes of few degrees of freedom, leading to limited resemblance to the actual geometry of the component.

With the popularization of accurate deformable models, introduced also by the computer graphics community (see [168] for a review), more realistic models of the components of an object are obtained. Recent works [169, 170, 171] have successfully shown how some types of animals can be modeled from a single image, relying mainly on the symmetry of their shape. These approaches differ from the ones proposed in computer graphics (e.g. [172, 173, 174]), where input from the 3D artist is essential. The single view modeling methods, however, are not suitable for modeling articulated objects since some of their assumptions become not valid. In particular, the components of the object do not share the same plane of symmetry.

In this work, we provide a solution to the problem of modeling articulated objects by explicitly modeling their components from various aspects. We consider a hierarchical decomposition of the object into components. Depending on the geometric complexity of the component, a different number of views is required for modeling. For example, an animal's torso typically requires three to four representative views (left, right, front and back). Views of a component lead to the component aspects. From each aspect an approximate model of the imaged component is obtained using the deformation paradigm. Then, these aspect models are merged together to form a component. Components are typical of an object class and, in turn, are assembled considering a reference pose of the object, providing a 3D model of the whole object. Here, we assume that the object components have been segmented out in the respective views. It is important to note that the different views need not correspond to the same physical object as far as objects belong to the same specific class. We focus our study on animals as they typically satisfy this property. An example of a 3D model obtained with our approach is shown in Figure 6.1 while an example of the decomposition in components and aspects is presented in Figure 6.2.

The chapter is organized as follows. In the next section we review related work. In Section 6.3 we describe how components are modeled by their aspects. In Section 6.4 we show how components are assembled to form the final model. In Section 6.5 we evaluate the proposed method and Section 6.6 addresses conclusions and future work.

6.2 Related work

Geometric modeling of objects is becoming popular in computer vision. Following the deformation methods introduced in the pioneering work of Terzopoulos [175], shape

generation from images provides good results by exploiting the contour generator. Single view modeling of objects with predefined genus and topology was introduced in [176, 169] using images of the same object family. Additional image cues have been considered in [177, 170] to model object classes from single views, and a similar approach has been taken by [171], exploiting the contour generator. A recent review is found in [178]. Multiple-view reconstruction of different object classes from few images has been successfully obtained using networks of objects with similar viewpoints [179], or for large scale shape reconstruction [180].

Differently from the 3D reconstruction methods we model an object not as a single rigid structure but as an articulated one. As opposed to SfM and factorization techniques, we model the views by deformation, we merge the obtained aspect models into components, and combine the components by a global optimization scheme, in order to estimate the view direction without requiring user input. The method allows us to join the components in several poses, which is the main novelty of our approach. The relation between the apparent contour and the contour generator, that we exploit here for assembling the components, has been studied since the early days of computer vision. Koenderink [181] studies various properties of the contour generator based on the results of differential geometry, establishing in [182] a rule relating contour and surface curvatures, which is also investigated in [183]. A comprehensive study of the contour generator of evolving implicit surfaces is found in [184]. The problem of fitting 3D objects in their apparent contour has been treated in [185] where optimization is performed to find 3D-2D correspondences, considering a parametric representation of the surface and an estimation of the view direction, initialized by the user. The problem has been also treated in [186] for non-rigid surface sequences.

The final visual quality exploits surface smoothing. Level-set based methods have been widely used for this task (for a survey see [187]), based on an implicit surface representation, and have the advantage of topological flexibility. We follow the approach of [188], enabling Boolean graphics operations, for obtaining a model with no internal faces.

6.3 Modeling object aspects into components

We consider an articulated object to be formed by *components*, such as head, torso, limbs, where each component can be mapped into a viewer-centered *aspect*. An aspect represents a view of the component from the viewing vantage point [167], as illustrated in Figure 6.2. The number of components of an articulated object, can be freely determined, the choice being based on common sense. The number of views needed to model a single component depends on the regularity of its shape. However, we do not rely on shape regularity because the component model is obtained from its aspects by optimization (see Section 6.3.2). Therefore, if a component is quite irregular, one would want to collect each of its idiosyncratic aspects.

The image selection task, leading to a choice of the components and their aspects in the spotted views, requires some user input. Such as, for example, the judgment of what is needed to recover a good model. In principle few images are needed, and in our examples we used four images, as shown in Figure 6.2. This said, the complex problem of automatically determining the number of components and aspects of a natural kind is not faced in this work.

6.3.1 Aspect modeling

Assume N_I available images $I_1, \ldots, I_{N_I}$ showing different views of some articulated object category C, which is supposed to have N_c components. Let $\Omega_i \subset \mathbb{R}^2$ be the domain of image I_i, $i = 1, \ldots, N_I$, and assume there is a chart of the segments of all visible components in image I_i, as shown in Figure 6.2, as for example provided in PASCAL-Part dataset [189] as well as in [190, 191]. Each segment α_{ic} in an image I_i of the object C defines an aspect of the specific component c. This aspect is mapped into a binary mask after translation and isometric scaling, keeping the proportions of the components w.r.t the original image. Let $T{:}\Omega_i \mapsto \Omega_{Tc}$ be the transformation applied to α_{ic}, then we define the mapping $A_{ic}{:}\Omega_{Tc} \mapsto \{0,1\}$, which returns precisely the binary mask of the transformed segment $\hat{\alpha}_{ic}$. The projection of the binary mask back into $\hat{\alpha}_{ic}$, is $A_{ic}^{-1}{=}\{(u,v){\in}\Omega_{Tc} \ | A_{ic}(u,v){=}1\}$. Let $\partial A_{ic} = \{(u,v) \mid |\frac{dA_{ic}}{du} + \frac{dA_{ic}}{dv}| > 0\}$, ($|\cdot|$ absolute value). We assume that ∂A_{ic} is a closed simple (Jordan) curve dividing the Euclidean plane in interior and exterior regions, where the interior is defined to be $int(A_{ic}) = \{A_{ic} = 1\}$ and has a prescribed sense of rotation. We define $F(u,v)$ the distance field at point $(u,v) \in int(A_{ic})$, namely:

$$F(u,v) = \min\{\|(u,v) - (\hat{u},\hat{v})\| \mid (\hat{u},\hat{v}) \in \partial A_{ic}\} \tag{6.1}$$

Let $\mathbf{q} \in int(A_{ic})$ be the center of a circle bitangent to ∂A_{ic}, having radius $r_{\mathbf{q}}$, namely $\mathbf{q}$ is on the medial axis of $int(A_{ic})$. We define:

$$h(u,v){=}\|(u,v) - \hat{\mathbf{q}}_{(u,v)}\|{+}r_{\hat{\mathbf{q}}_{(u,v)}}, \tag{6.2a}$$

with

$$\hat{\mathbf{q}}_{(u,v)} \coloneqq \min\{\|(u,v) - \mathbf{q}\| \mid \mathbf{q}{\in}MedAxis(int(A_{ic}))\}. \tag{6.2b}$$

To obtain the 3D model from A_{ic} we minimize the elastic energy deforming the distance between nearby points, which is driven both by internal forces, inducing local stretching and bending, and external forces. A surface $\boldsymbol{\varphi}{\subset}\mathbb{R}^3$, parametrized by the function $g{:}\Omega_{Tc} \mapsto \mathbb{R}$, is computed by minimizing the strain energy functional defined by the first and second fundamental forms [192], plus an external force G, or load. Energy strain linearization is attained by considering the first and second derivatives of g [168]. The energy functional is:

$$E(g) = \int_{\Omega_{Tc}} \mathbf{g}_\lambda^\top \mathbf{Q}_\lambda \mathbf{g}_\lambda + \mathbf{g}_\beta^\top \mathbf{Q}_\beta \mathbf{g}_\beta - 2Gg \ \mathrm{d}u\mathrm{d}v \tag{6.3}$$

Here $\mathbf{g}_\lambda{=}(g_u, g_v)^\top$, $\mathbf{g}_\beta{=}(g_{uu}, g_{vv}, g_{uv})^\top$, $\mathbf{Q}_\lambda$ is a $2{\times}2$ matrix of stretching parameters, $\mathbf{Q}_\beta$ is a diagonal $3{\times}3$ matrix of bending parameters, assumed known, and G is the load:

$$G(u,v){=}\frac{F(u,v)}{h(u,v)}\left(\delta_1(u,v)\gamma_1{+}(1-\delta_1(u,v))\gamma_2\right) \tag{6.4}$$

Here F and h are defined in eq.(6.1, 6.2), $\delta_1(u,v)$ is the indicator of ∂A_{ic} convexity at (u,v) and $\gamma_1, \gamma_2 \in \mathbb{R}_+$ are weights. This external force is applied to make the final surface growing steeper both near the boundary and where the initial mask is thinner and convex (see Figure 6.3). The scheme for finding the solution $g(\cdot)$ of the energy functional (6.3) is based on the Finite Element method, as described in [193], applied to

Algorithm 6.1: Aspects modeling

Input: Aspects $A_{ic}, i = 1, \ldots, N_{A_c}, c = 1, \ldots, N_c$, aspect parameters $\mathbf{Q}_\lambda$, $\mathbf{Q}_\beta$
Output: Aspect models $B_{ic}, i = 1, \ldots, N_{A_c}, c = 1, \ldots, N_c$

```
for c = 1 : N_c do
    for i = 1 : N_{A_c} do
        Generate a triangulation for A_ic;
        Choose the set of shape functions (at least quadratic) and the quadrature
          nodes;
        Assemble the stiffness matrix and loads vector using the quadrature rule;
        Find the weights of the shape functions solving the equation KX = H;
        Find the displacements g_ic using eq. (6.5);
        Compute mesh B_ic based on the triangulation, and closure by reflection,
          of φ_ic.
```

the associated Euler-Lagrange equation. The approximation of the displacement $g(u, v)$, which minimizes the energy functional (6.3) is obtained as:

$$g(u, v) = \mathbf{X}^\top \mathbf{\Phi}(u, v), \tag{6.5}$$

Here $\mathbf{\Phi}$ is the coefficient matrix of the continuous shape functions, $\mathbf{X}$ is the matrix of the unknown weights, obtained by solving the following quadratic minimization problem:

$$\min_{\mathbf{X}} \left\{ \mathbf{X}^\top \mathbf{K}\mathbf{X} - \mathbf{H}^\top \mathbf{X} \right\}, \tag{6.6}$$

with $\mathbf{K}$ the stiffness matrix and $\mathbf{H}$ the vector of the loads. To constrain the solution at the boundary ∂A_{ic}, homogeneous Dirichlet conditions are applied into the PDE problem formulation. A smooth closed surface B_{ic} for each aspect (segment $\hat{\alpha}_{ic}$) of component c of object C, as viewed in image I_i, is obtained by joining φ_{ic} with its reflection along the z=0 plane, see Figure 6.3. Algorithm 6.1 describes the main steps involved.

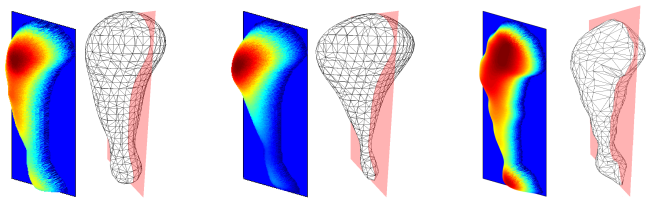


Figure 6.3: Comparison of the solutions (depth maps) and reconstructed surfaces (meshes) of a cat's leg, **Left**: obtained with (6.4), **Center**: without the load. (Best seen in colors). **Right**: with load, given noise on the contour segmentation.

6.3.2 Component building

Let $\mathcal{B}_c$ be the set of closed surfaces, obtained as described above, which we denote the aspect models of the component $c = 1, \ldots, N_c$. For each component there are N_{A_c} aspect models, namely, $\mathcal{B}_c = \{B_{1c}, \ldots, B_{sc}\}$, with $s \leq N_{A_c}$. To obtain a consistent model for c, the aspect models in $\mathcal{B}_c$ need to be combined. To achieve this we chose a reference model $B_{rc} \in \mathcal{B}_c$ and estimate the 3D transformation between each aspect

model $B_{ic} \in \mathcal{B}_c$ and the reference model B_{rc}, as illustrated in Algorithm 6.2. Each aspect model B_{ic} is labeled with respect to the image I_i it was obtained from, and with respect to the component c it is a view point of, hence we use feature points extracted from the image I_i (see Figure 6.2) to compute the relative transformation $T_{ri}^{(0)}$ between B_{rc} and B_{ic}. A refined solution is then obtained by 2.5D registration.

Algorithm 6.2: Aspect registration

Input: Indices of reference aspect models rc, $\mathcal{B}_c$, $\hat{\alpha}_{ic}$, $i = 1, \ldots, N_{A_c}$, $c = 1, \ldots, N_c$

Output: Transformation T_{ri} between reference B_{rc} and aspect models $B_{ic} \in \mathcal{B}_c$, $i = 1, \ldots, N_{A_c}$

1 **for** $c = 1 : N_c$ **do**
2 **for** $B_{ic} \in \mathcal{B}_c$ **do**
3 Detect a set of feature points F_{ic} in the segment $\hat{\alpha}_{ic}$ (e.g. by keypoints, SURF [194] features or similar);
4 Project F_{ic} on B_{ic} to obtain the 3D feature points X_{ic};
5 Find feature matches $F_{ic} \leftrightarrow F_{rc}$;
6 **if** $\#\text{matches} > 3$ **then**
7 Estimate 3D transformation $T_{ri}^{(0)}$ based on $X_{ic} \leftrightarrow X_{rc}$ up to an affine transformation
8 **else**
9 Ask user for manual initialization
10 Apply $T_{ri}^{(0)}$ on B_{ic};
11 Compute depth image $\bar{d}_{ic}$;
12 Dense 2.5D registration of $\bar{d}_{ic}$ w.r.t. d_{rc}.

The last step of Algorithm 6.2 (line 12) is a dense 2.5D registration between the depth image d_{rc} of the reference aspect and the depth image $\bar{d}_{ic}$ corresponding to the transformed i-th aspect of component c. In the following we drop the subscript c as reference is intended to the component c. The registration is obtained via the minimization problem

$$\min_{\xi_i \in \mathfrak{a}(3)} \|d_r - \bar{d}_i(\xi_i)\|_{L_1}, \tag{6.7}$$

with $\mathfrak{a}(3)$ the Lie algebra of the 3D affine transformation group and ξ_i a twist belonging to this Lie algebra. The objective function involved is non-smooth and non-linear in ξ_i. We consider a local convex approximation of the objective function by iterative linearization with respect to ξ_i and we then apply the Legendre-Fenchel transform, transforming the original minimization problem to a sequence of saddle-point problems. Optimization is performed in a coarse-to-fine framework to avoid local minima. Let $\mathbf{q}$ be the dual variable, Q the union of pointwise L_1 balls, $\delta\xi_i^{(k)} = \xi_i - \xi_i^{(k)}$, $\mathbf{d}_r$ the vectorized reference depth image, and $\bar{\mathbf{d}}_i(\xi_i^{(k)})$ the vectorized depth image of aspect i transformed according to $T^{(k)} = \exp(\delta\xi_i^{(k)})T^{(k-1)}$. Let $\frac{d\mathbf{p}}{d\xi_i}\big|_{\xi_i^{(k)}}$ be the directional derivative of $\mathbf{p}(\xi_i) = \mathbf{d}_r - \bar{\mathbf{d}}_i(\xi_i)$ with respect to ξ_i evaluated at $\xi_i^{(k)}$. The saddle-point problem at the k-th iteration is

$$\max_{\mathbf{q} \in Q} \min_{\delta\xi_i^{(k)} \in \mathfrak{a}(3)} \mathbf{q}^\top \left(\mathbf{p}(\xi_i^{(k)}) + \frac{d\mathbf{p}}{d\xi_i}\bigg|_{\xi_i^{(k)}} \delta\xi_i^{(k)} \right). \tag{6.8}$$

A solution is computed by applying primal-dual optimization to estimate the saddle-point at each level.

The optimization significantly improves the registration, provided that the initialization $\bar{d}_i$ is situated in the convex basin of the optimal solution. The final solution depends on the choice of the reference aspect and the order in which the remaining aspects are considered; however, given that N_c is a small number, the solutions are virtually equivalent.

Given the transformations, leading to a consistent registration of the aspect models, we merge them into a single component model (Figure 6.4). To achieve this, we first compute a volumetric representation of each model surface. We use the definition of Inner Product Field (IPF), as described in [188]. The IPFs grants an implicit representation of the aspect models B_{ic} and we can exploit the following result: given $n \geq 2$ implicit surfaces $\phi_1(x), \ldots, \phi_n(x)$, then $\phi_\cup(x) = \min\big(\phi_1(x), \ldots, \phi_n(x)\big)$ is the union of their interior regions and corresponds to the envelope of the surfaces. As a final step, the component model is slightly smoothed to attenuate possible irregularities and artifacts. The smoothing is applied on the volumetric representation of the aspect model using the Level Set method according to the mean curvature flow [195]

$$\phi_t + V_n \|\nabla\phi\| = 0, \tag{6.9}$$

where $V_n = -b\kappa$ is the velocity field in the normal direction generated from the surface curvature κ, and $b \in \mathbb{R}$. A mesh is then generated by standard meshing techniques (e.g. [196]).

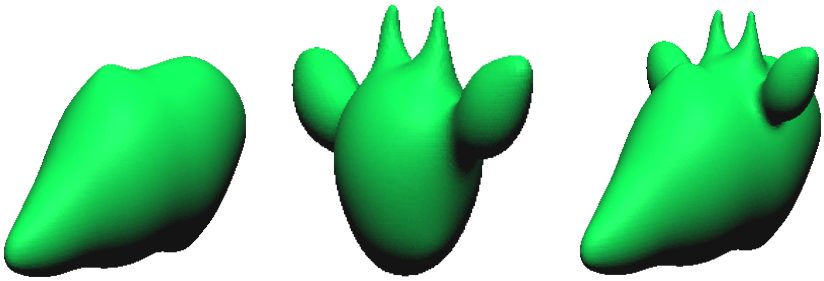


Figure 6.4: Aspects modeling and component building of the giraffe head. **Left**: side aspect, **Center**: front aspect, **Right**: component model.

6.4 Assembling of the articulated object

Components are assembled in order to obtain a model of the entire object in a reference pose. In particular, we use the apparent contours of the components in two or more views of the object in a reference pose, as the ones displayed in Figure 6.5. We assume here that all components are partially visible in the chosen views, that segments are available in each view and obtained by an orthographic projection. The visibility requirement can be relaxed as the number of views increases.

First, we recover the optimal transformation for each component, which makes its projection comply with the apparent contour. We treat this as a 3D-2D registration problem (see [197] for a review). We consider each component as a sufficiently smooth surface S (e.g. of class C^2) and the apparent contour is a planar contour γ. These two entities are related by the contour generator (CG), which is a space curve Γ, defined

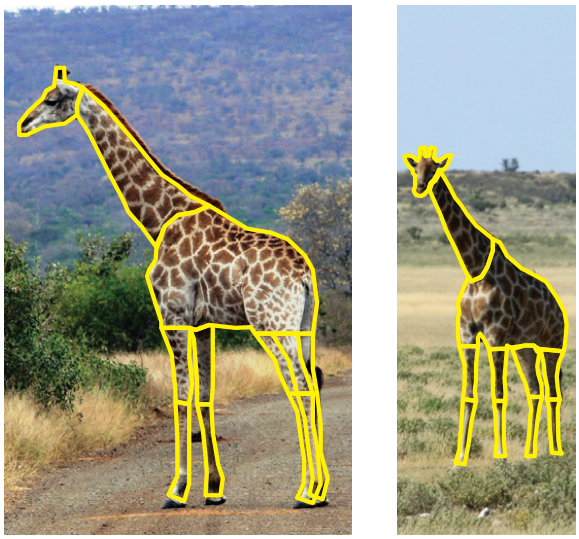


Figure 6.5: Two views of a giraffe in reference pose with overlaid component masks.

by the set of visible points on S, where the view direction $\mathbf{v}$ is locally tangent. The projection of Γ according to $\mathbf{v}$ produces γ up to a 2D similarity transformation. To register each 3D component in its apparent contour, we find a view direction and the corresponding CG, which projects to a contour $\hat{\gamma}$ as similar as possible to γ.

Let $\mathcal{Y}(S)$ be a set of points sampled on S. Under the given assumptions, it suffices to identify two points $Y_1, Y_2 \in \mathcal{Y}(S)$ lying on Γ, to compute the view direction. Indeed, observe that Γ depends only on $\mathbf{v}$, and two points with non parallel normals $\mathbf{n}(Y_1)$ and $\mathbf{n}(Y_2)$ define the view direction up to a sign, as $\mathbf{v} = \mathbf{n}(Y_1) \times \mathbf{n}(Y_2)$. Given two points $\mathbf{y}_1, \mathbf{y}_2 \in \gamma$ we seek the corresponding points $Y_1, Y_2 \in \mathcal{Y}(S)$. We identify the best matches by minimizing the energy function

$$E(Y_1, Y_2; \mathbf{y}_1, \mathbf{y}_2) = \sum_{l=\{1,2\}} \big(E_{cg}(Y_l; \mathbf{y}_l) + E_{curv}(Y_l; \mathbf{y}_l)\big)$$
$$+E_{ang}(Y_1, Y_2; \mathbf{y}_1, \mathbf{y}_2) + E_{dist}(Y_1, Y_2; \mathbf{y}_1, \mathbf{y}_2). \tag{6.10}$$

The term E_{cg} specifies that the points must lie on the CG corresponding to the estimated viewpoint. The last three terms take into account local geometric properties that the contour and CG have to satisfy. All these terms are invariant with respect to 2D similarity transformation, which is a computational bottleneck when considered. We examine now in detail each term.

E_{curv} is based on the relation between the curvature of the surface and the curvature of the apparent contour. First, the sign of the curvature of γ at point $\mathbf{y}$ $\kappa^{\gamma}(\mathbf{y})$ should match the sign of the Gaussian curvature of S at the corresponding point Y [181]. Additionally, $\kappa^{\gamma}(\mathbf{y})$ and the curvature of Γ at the corresponding point $\kappa^{\Gamma}(Y)$ satisfy the relation

$$\kappa^{\Gamma}(Y) = \sin^2\theta\, \kappa^{\gamma}(\mathbf{y}), \tag{6.11}$$

with θ the angle between $\mathbf{v}$ and the CG at Y [181, 198]. Based on this result, suitable bounds regarding the curvature of γ, Γ and S are provided by the following proposition:

Proposition 6.1. *Let S be a smooth surface and $\pi(\cdot)$ the projection operation. The curvature of the contour γ at a non-cusp point $\mathbf{y}$, the curvature of Γ at the corresponding point Y and the principal curvatures of the surface κ_1^S (minimum) and κ_2^S (maximum) at Y satisfy the inequality*

$$\begin{aligned} \kappa_1^S(Y) \leq \kappa^\Gamma(Y) \leq \kappa^\gamma(\mathbf{y}) \leq \kappa_2^S(Y), \\ \textit{with: } \mathbf{y} \in \gamma,\, Y \in \Gamma, \mathbf{y} = \pi(Y). \end{aligned} \tag{6.12}$$

Proof. Consider a generic point $Y \in \Gamma$. We assume first that Y is not umbilical. The leftmost inequality is trivial as the curvature of Γ at Y cannot be smaller than the minimum curvature of the surface at Y. The second inequality follows from (6.11). To show the last inequality we consider the osculating sphere O_Y of the surface at Y which has curvature $\kappa^{O_Y} = \kappa_2^S(Y)$. Regardless of the view direction, γ at $\mathbf{y}$ can at most locally lie on the projected contour of O_Y which is a circle with curvature κ^{O_Y}. Hence, the curvature of γ at $\mathbf{y} = \pi(Y)$ is locally bounded by the curvature κ^{O_Y} which is equal to $\kappa_2^S(Y)$. If the point is umbilical then all equalities trivially hold. □

Corollary 6.1. *Considering a point $\mathbf{y} \in \gamma$, a region $R \subseteq S$ is an admissible region of the corresponding point $Y \in \Gamma$ iff $\kappa_1^S(\mathbf{Z}) \leq \kappa^\gamma(\mathbf{y}) \leq \kappa_2^S(\mathbf{Z})$, $\forall \mathbf{Z} \in R$ and the sign of $\kappa^\gamma(\mathbf{y})$ matches the sign of the Gaussian curvature G^S in R.*

In the following for brevity we omit the explicit relation with the surface/curve points. Based on the previous result the curvature term can be expressed as

$$\begin{aligned} E_{curv} = \omega_\kappa \, D_{[\kappa_1^S, \kappa_2^S]}(\kappa^\gamma) + \\ \omega_G \max(-\operatorname{sgn}(G^S \kappa^\gamma), 0), \end{aligned} \tag{6.13}$$

with $D_{\mathcal{J}}(v) = \min_{w \in \mathcal{J}}(\|v - w\|)$ and $\omega_\kappa, \omega_G > 0$ weights relating the terms.

The term E_{ang} expresses the fact that the angle between the normals $\mathbf{n}(Y_1)$, $\mathbf{n}(Y_2)$ matches the corresponding angle on the apparent contour. The same holds for the angle between each of the normals and the connecting segment $(Y_2 - Y_1)$ projected on the plane spanned by the normals. Letting c be the cost function that penalizes differences between the corresponding angles (e.g. $c(\theta, \phi) = \tan(|\theta - \phi|)$), we define

$$E_{ang} = \omega_n c(\theta_n, \theta_\eta) + \omega_b c(\theta_B, \theta_b), \tag{6.14}$$

with θ_n, θ_η the angles between the 3D and 2D normals respectively, θ_B, θ_b the angles between the base segment and one of the normals in 3D and 2D respectively, and $\omega_n, \omega_b > 0$ the relative weights. The term E_{dist} is defined as

$$E_{dist} = \omega_d \left(\frac{\|Y_1 - Y_2\|}{d(S)} - \frac{\|\mathbf{y}_1 - \mathbf{y}_2\|}{d(\gamma)} \right)^2, \tag{6.15}$$

with $d(\cdot)$ the diagonal length of the corresponding entity's bounding box and $\omega_d > 0$ the relative weight.

Finally, the term E_{cg} is taken equal to the maximum penetration depth of the view ray passing through Y with respect to S and specifies the constraint that Y is on Γ.

We find the global minimum of the energy function with a branch-and-bound search strategy [199, 200]. First, we find the two points on γ which result into the most restricted region on S based on the previous corollary, and use them as initial points for the search. The pair of points which corresponded to the lowest energy value returns the view direction $\mathbf{v}$. The remaining 2D similarity transformation is then recovered by applying a shape matching technique between the resulting contour and the measured one (see [201]). This procedure gives the relative pose of each component with respect to the view. Not depending on all points of the apparent contour, it is robust with respect to the visible portion of the contour and the shape of the 3D component. The solution can be refined by performing an iterative LSE minimization. We should note that the assembling step is robust with respect to noise as the components are smoothed before it is applied. An example is shown in Figure 6.3.

By registering each component in the given view we recover their relative position with the only exception of the translation in the viewing direction. We solve this ambiguity by using the other views. In particular, since the object is imaged in the same pose from two or more known views, the depth ambiguity is resolved. A single model is computed from the assembled components by following the steps presented at the end of Section 6.3.2.

6.5 Evaluation

Modeling time The implementation of the proposed method consists of a mixture of Matlab and CUDA code. In particular, 2.5D registration of the modeled aspects, IPF computation and surface smoothing of the models are implemented in CUDA, while aspect modeling and component assembling are implemented in Matlab. A report of the time required for computing the models shown in this section is presented in Table 6.1.

Table 6.1: Modeling time report (AM-aspect modeling, CB-component building, CA-component assembling, Sm-smoothing).

Model	AM [sec]	CB [sec]	CA [sec]	Sm [sec]	Total [sec]
Cat	532	1.8	1942	0.09	2521
Dog	514	2.1	1026	0.08	1855
Cow	598	2.2	1311	0.10	1919
Sheep	426	1.9	1417	0.07	1826
Hippo	577	1.8	1514	0.07	2008
Giraffe	479	2.2	1410	0.06	1901
Kangaroo	441	2.0	1396	0.09	1723
Standing Horse	484	1.9	1613	0.05	2017
Landing Horse	505	2.1	1855	0.07	2090
Rearing Horse	494	1.9	1951	0.06	2034

The experiments were performed on a PC equipped with an Intel i7 3.6GHz CPU, 16GB RAM and an NVIDIA GTX970 graphics card. All models presented in the section have been modeled from four input images. Further results are presented in the accompanying video.

Model comparison We performed an extensive comparison of models obtained with our method using images taken from the Web, against models downloaded from the Web. All images were taken from Flickr, while most of the downloaded models were obtained from the 3D warehouse of SketchUp, the rest have been taken from other repositories. We evaluated the similarity of our models to the downloaded ones using two different similarity measures, the Hausdorff distance [202] and the normalized symmetric difference. We considered our model as reference and preprocessed the models taken from Web to make the results comparable. Preprocessing consisted of the following steps: (a) model clean-up; remove internal faces, recover manifoldness and close holes; (b) manual orientation w.r.t. reference model; (c) automatic non-isotropic scaling for matching the bounding box with the reference model.

The Hausdorff distance was computed directly on the meshes of the models. For the symmetric difference we used the volumetric representation obtained via IPF. The distance is computed as the difference between the number of voxels in the union and the number of those in the intersection of the two volumes, normalized by the total number of voxels. The results of the comparison are presented in Figure 6.6, where numbers correspond to the average values of the distances w.r.t. all downloaded models of each class (3-4 models). These results show that the models computed with our method actually represent the modeled class. Indeed, the average distance with respect to the downloaded models of the same class is consistently smaller in comparison to the distances with respect to the other classes.

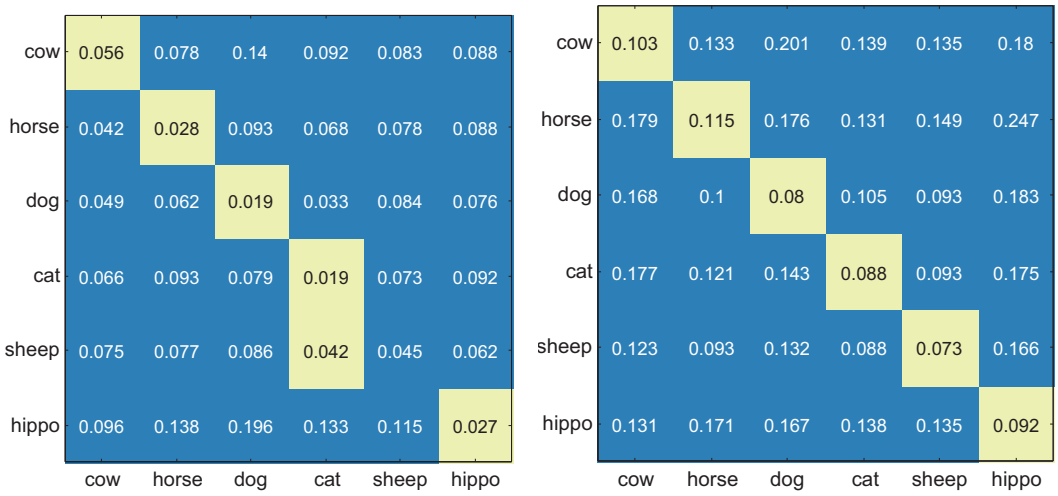


Figure 6.6: Model comparison (smallest values are highlighted); normalized symmetric differences (**left**) and normalized Hausdorff distances (**right**) between the models.

For a more objective evaluation, we applied the proposed approach to images of 3D models downloaded from the Web. In particular, we generated images of the rendered 3D models from four vantage points, on which the segmented aspects were extracted. In this way, the downloaded models acted as ground truth with respect to which our models were compared using the normalized Hausdorff distance. The results of this comparison are presented in Figure 6.7 and in Table 6.2, where the mean values are given. We should note here that as this procedure allowed us to easily obtain two images of the object in more "unstable" poses, we were able to model the objects in different poses, as seen for example for the horse (standing, landing and rearing poses).

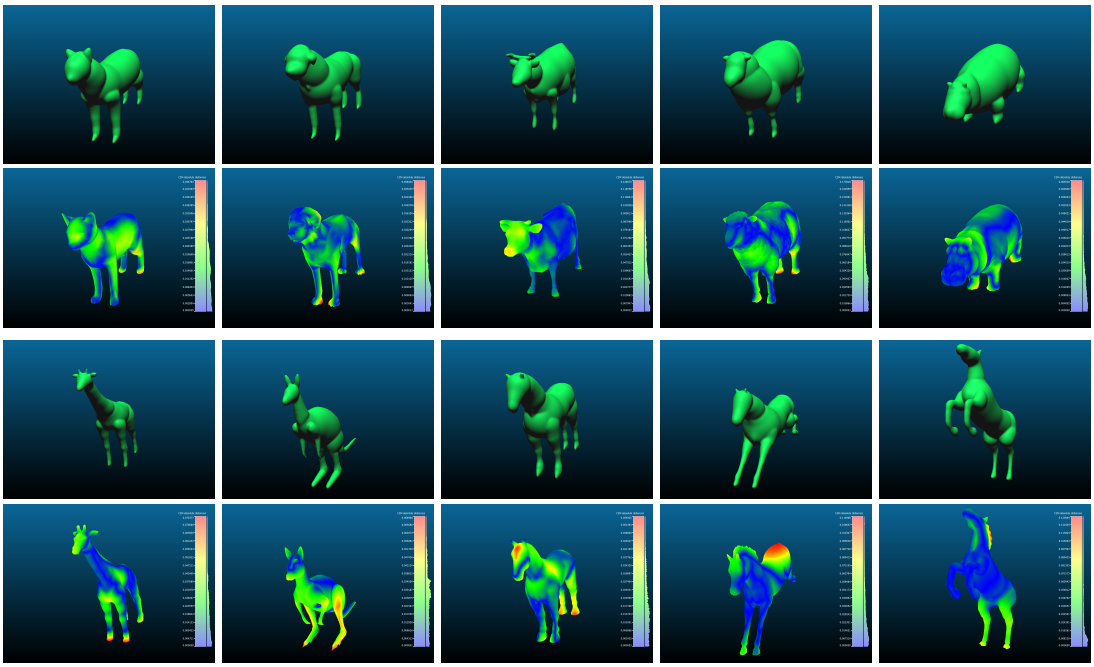


Figure 6.7: Comparison between animals modeled with our approach (odd rows) from images of models downloaded from the web (even rows) which were used as ground truth. The images of the bottom group show the distribution of the normalized Hausdorff distance on the ground truth model. (Best seen in color and on-screen)

Table 6.2: Mean normalized Hausdorff distance between the models reconstructed with our approach and ground truth.

Cat	**Dog**	**Cow**	**Sheep**	**Hippo**
0.012	0.012	0.030	0.040	0.013
Giraffe	**Kangaroo**	**Standing Horse**	**Landing Horse**	**Rearing Horse**
0.018	0.023	0.016	0.028	0.020

Perceptual study Because of the nature of the problem, similarity distances may not always be representative. To further evaluate the quality of our models we performed a perceptual study with the help of volunteers.

Ten volunteers who did not know the purpose of the study participated in the experiment. Six participants were male and four female, 60% had from 22 to 25 years and 40% from 25 to 29 years. Finally, three subjects reported corrected-to-normal vision and the rest normal vision.

The models presented in Figure 6.8 (left) were used for conducting the study. Participants were invited to ask questions before the experiment. After providing the necessary information and consent the task presented to the participants was:

> Various 3D models will be shown on the screen during the experiment. For each model, you need to identify the corresponding animal and give a mark for its quality. You can interact with the model for as long as you prefer before answering.

The models were presented on the screen with a uniform green shaded material on blue background, as shown in Figure 6.8. The participants marked the answers on a

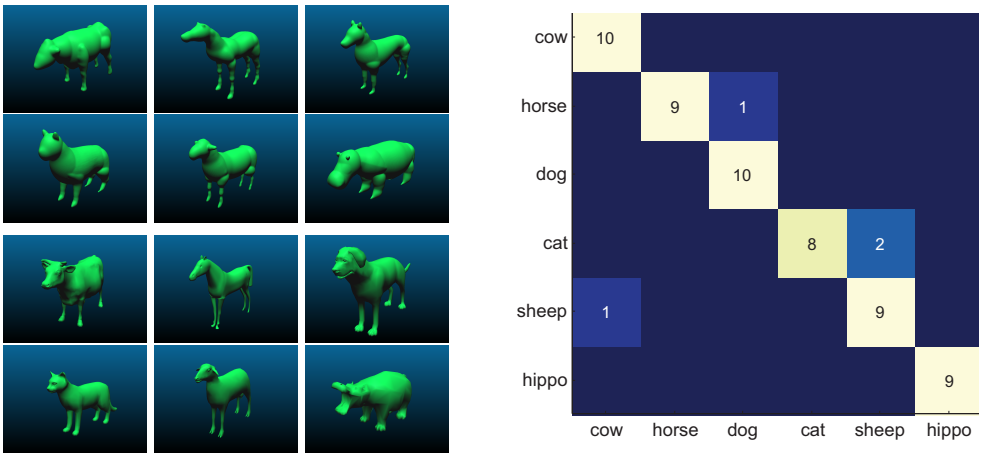


Figure 6.8: **Left**: Animal models used in the perceptual study. *Top group:* models computed with our method; *Bottom group:* models downloaded from the Web. **Right**: Confusion matrix from the perceptual study.

special form, where the animal class could be specified freely and a scale of discrete values from 0 to 5 was used for evaluating the quality of the model. The models were presented in a random order to avoid bias caused by repeated ordering.

We consider the null-hypothesis H_0 that participants randomly selected the animal class, while the alternative hypothesis H_1 is that users correctly recognized the animal. Cross-tabulation was performed on the answers provided by the participants regarding the class of animal represented by our models, and the resulting confusion matrix is shown in Figure 6.8 (right). One can observe that the participants almost always identified successfully the animal class. In fact, the null hypothesis is rejected as the chi-square value is $\chi^2 = 247$, corresponding to a practically vanishing p-value. It is important to note that the participants did not know in advance the classes of animals involved. This justifies also the last row of the confusion matrix, as one participant recognized the hippo as a pig.

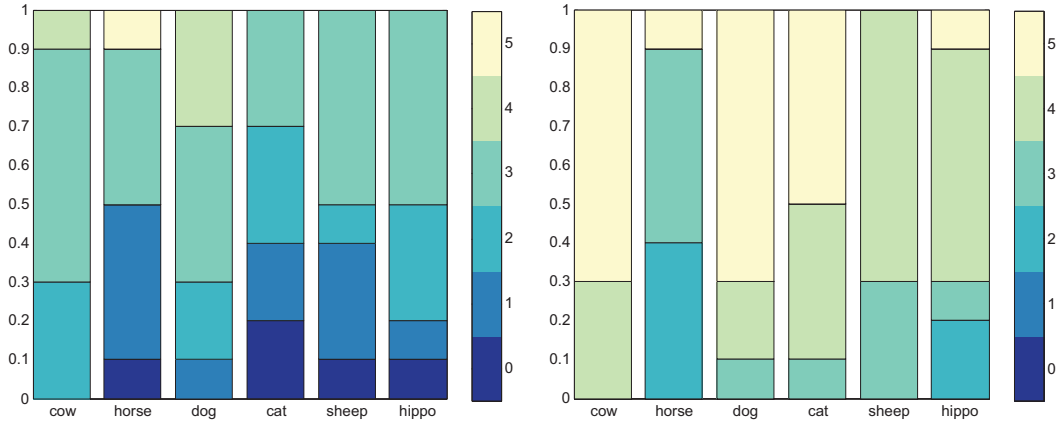


Figure 6.9: Vote distribution for the models produced with our approach (**left**) and models taken from the Web (**right**).

The distribution of votes given by the participants for the model quality is presented in Figure 6.9. The models downloaded from the web received higher votes in average, with a difference of 1.9 scale units with respect to the average vote for our models. This

Table 6.3: Per-class percentage of votes above 3 (good) given to the models reconstructed by our method (first row), and the models downloaded from the web (second row).

Cow	**Horse**	**Dog**	**Cat**	**Sheep**	**Hippo**
70%	50%	70%	30%	50%	50%
100%	60%	100%	100%	100%	80%

is understandable considering that our models correspond to more abstract class models, lacking particular details like eyes, nose and tail. Nevertheless, the percentage of the participants who gave a vote above 3 (good) for the quality of our models (Table 6.3) indicates that the models are of satisfying quality.

6.6 Conclusions and future work

We propose a method for computing 3D models of articulated objects, by decomposing them into components. Realistic models of the object components are built by merging together 3D models obtained from different aspects, considering a kind of aspect graph [167], which indicates the essential aspects. Aspects are extracted from images downloaded from the Web. The entire object is obtained by reassembling the components using two or more images of the object in a reference pose. Our experiments suggest that our method is able to provide realistic models of the objects, both in terms of a perceptual analysis and by a quantitative analysis of their similarity with respect to human-created 3D models.

An important extension of this work is the possibility to model the object in different configurations by using a single image. This can be made possible by learning spatial relations between the components (joints, joint range etc.) and possibly also a distribution of the object poses, which would allow to compute realistic models even when some of the components are occluded. Finally, another useful extension would be the automatic selection of the most representative aspects for each component from a set of images.

Chapter 7

Single image Non-Lambertian surface modeling

In this Chapter, a methodology for 3D surface modeling from a single image is proposed. The principal novelty is concave and specular surface modeling without any externally imposed prior. The main idea of the method is to use BRDFs and generated rendered surfaces to transfer the normal field, computed for the generated samples, to the unknown surface. The transferred information is adequate to blow and sculpt the segmented image mask in to a bas-relief of the object. The object surface is further refined basing on a photo-consistency formulation that relates for error minimization the original image and the modeled object.

7.1 Introduction

There is an increasing need for 3D models of objects, from single images, for several applications such as digital archives of heritage and monuments, anatomy models for pathology detection, small artifact models for populating rendered 3D scenes with objects or augmenting a MOCAP sequence with tools for manipulation and, finally, for robotics. Likewise, there is a growing awareness that 3D modeling from a single image helps to navigate through the sea of terabytes of images, for the object recognition challenge.

That surface modeling from a single view has to deal with shading and the way materials shine and reflect the light has become clear since the works of [21] and [203]. Yet, only recently a great deal of work has been done to merge the rich information that light conveys about an object and its shape. Relevant examples are studies on specular reflection of materials and light incidence [204, 205], so as to dismiss the Lambertian hypothesis, and on how illumination and reflectance combine to influence an object shape perception [132] and its geometry [206].

Here, we address these problems by introducing a novel method, which is unbiased to changes of the ambient light, taking care of both concavities and sharp parts of an object. This is the main contribution of this work. Our approach is related to SIRFS [132], who introduced priors for shape, albedo and illumination, so as to learn the most likely shape. Here we do not introduce any prior, instead we formulate a hypothesis.

Our hypothesis is that a sufficiently large number of patches, with varying surface curvature, rendered with different materials, with known reflectance properties, and varying incidence and reflection angles, can be used to estimate these properties in un-

Figure 7.1: An example of 3D surface of an object from ImageNet

known objects. Through this generalization, the reflected, specular and diffuse light of a new object, seen in a single image, can be recovered. We show that this hypothesis is plausible and proves to give interesting results. Indeed, the normal field of the rendered surfaces, applied as an external deformation force, basing on finite element method [207], is used to sculpt the unknown object surface. This gives very appealing results, that are further refined to meet photo-consistency requirements. An example of the input image and the rendered surface recovered by the proposed method is shown in Figure 7.1.

The chapter is organized as follows. In the next section we give some pointers to related works, although we are not able to cover the whole extraordinary literature on the topic. In Section 7.3 we introduce the basic concepts supporting this work, namely the BRDF [21], the MERL database [208], how rendered surfaces (r-surfaces) are generated, and few hints for the reference database ImageNet [209] and for recovering the object contour [210]. In Section 7.4 we introduce the unsupervised learning method to validate the hypothesis that the r-surfaces convey sufficient information about unseen objects. The distribution of the data is inferred via a nested Dirichlet process mixture model [46, 211]. Features of the highest level in the hierarchy are obtained by sparse stacked autoencoders [212, 213]. The outcome is a selection of a BRDF and of the most plausible normals on each patch covering the object image. These data, as described in Section 7.5, form the external forces of the energy which deforms the planar patches, covering the object mask, into the object surface. This extends the deformation method [192] to concavities and sharp object parts. Finally, the resulting surface model is made consistent with the object appearance in the image, by revising the light effects, as described in Section 7.6. This is obtained with a rich energy term taking care of both photo-consistency and surface depth, optimized via total variation minimization. The high level ideas of the approach are visualized in Figure 7.2. Results shown in Section 7.7 are very promising and new, with respect to the state of the art.

7.2 Related Works

The concept of Bidirectional Reflectance Distribution Function (BRDF) has been largely used in the computer vision community [214] to infer the material reflectance properties of a known object. Some approaches model objects in 3D by imposing an unknown BRDF such as in [204], where object shape is recovered with two different methods

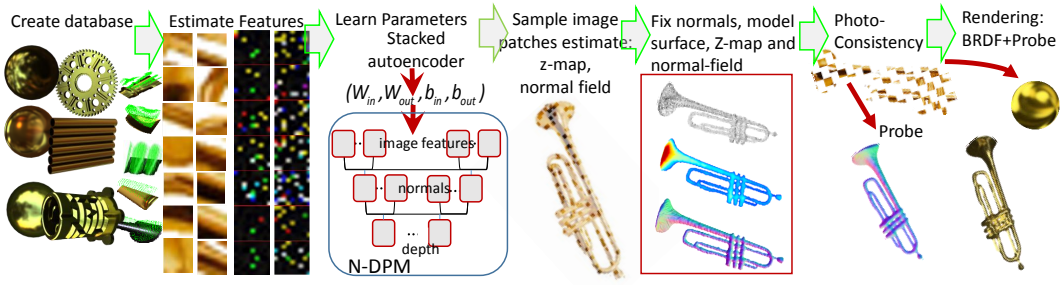


Figure 7.2: High level ideas of the work.

requiring, however, multiple images of the same object. Retinex theory [215] has been used for separating the shading component from the reflectance one in an image. A similar distinction is made in [216] for extracting the intrinsic characteristics of surface orientation, reflectance and incident illumination from a single image. Very recently, in [217] the authors propose a convolutional neural network approach to separate the albedo component from the shading. Shape from Shading (SFS) recovers the shape of an object from a single image, provided the illumination and the reflectance are given (see [218] and references therein). SFS makes strict assumptions, usually a Lambertian material with a single light, to find the solution for the otherwise unconstrained problem. In [206], reflectance and geometry are jointly recovered by assuming a statistical BRDF model and known lighting environment. In our work, instead, we learn a non-parametric model of surface appearance directly from the measured BRDFs in unknown illumination environment. [219] propose a discriminative learning approach for the SFS problem, considering an uncalibrated illumination without the assumption of a single point light. [220] examine the light locally on small patches in a Lambertian setting, and for each image patch a set of 3D surface patches, that may have generated the imaged ones, is sampled. Differently from them, our approach is not based on Lambertian assumptions. In [221], a 3D model from a single image is reconstructed basing on super-pixels segmentation and the Random Markov Field approach. In [222], both inter-reflections and photometric stereo are combined to resolve the generalized bas-relief ambiguity, but in a Lambertian setting. Finally, [223] consider specular objects estimating the corresponding 3D shapes by means of shape from specular flow approach with general motion.

7.3 Multivariate reflectance model and r-surflets

In this section, we introduce some preliminary concepts concerning the BRDF, the method for rendering object surfaces (r-surfaces), and finally the segmentation algorithm for objects taken from ImageNet.

BRDF. The model considers incident directions (ϕ_i, φ_i), in spherical coordinates, defined on the local reference frame of the surface element, within some solid angle $d\omega_i$ and the direction of reflection (ϕ_r, φ_r) over some solid angle $d\omega_r$. We assume that the observer line of sight is orthogonal to the image plane and centered on the object center of mass. We assume also a geometric optics model, that is, the electromagnetic character of light can be ignored [224]. Under this hypothesis wave interference and diffraction can be disregarded. We consider three kinds of reflections: specular, diffuse, and ambient. Specular reflection, in its ideal form, is a Dirac delta function, so

that $\phi_r = \phi_i$ and $\varphi_r = \varphi_i + \pi$. The specular reflection preserves the solid angle of the incident ray, namely $d\omega_i = d\omega_r$. Diffuse scattering is Lambertian, not depending on the direction of reflection. Ambient scattering collects all other kinds of reflection. In particular, lighting due to environment reflections on the surfaces is here treated as noise, so that we actually model arbitrary environment light probes.

Given the incoming light direction $d\omega_i$ and the reflected light direction $d\omega_r$, both defined with respect to the normal of an infinitesimal surface element, the BRDF [21] is the ratio between the amount of light reflected from the surface along $d\omega_r$, namely radiance L_r, and the total amount of light incoming to the surface element along $d\omega_i$, namely irradiance $\mathcal{E}_i$.

There are two main databases for the BRDF values of several materials under different light conditions, the MERL Database [208], for isotropic materials, and the UTIA one for the anisotropic materials [225]. We have considered the isotropic BRDFs (see [225] for a discussion on isotropic and anisotropic BRDF), where the material reflectance properties are invariant under rotation of the surface about its normal. This is because the MERL database is rich of most of the everyday objects materials like aluminum, brass, chrome, plastic, and acrylic.

3D models and surface rendering. We have created a synthetic dataset using 3D models of a number of real objects, obtained from different databases such as 3D Warehouse and TurboSquid. To ensure a wide variety of surface curvatures and curvature maps in our dataset, and to guarantee its semi-completeness, we consider a number S of both smooth objects, such as tubes and rings, and irregular ones such as gear wheels, see Figure 7.2, Panel 1, for some examples. Each object surface is then rendered with Blender. Each of the obtained r-surfaces, is of dimension $m \times m$ pixels, with $m \in \{256, 512\}$ and, such that for each angle of incident and reflected light $(\phi_i, \varphi_i, \phi_r, \varphi_r)$, and BRDF material, an r-surface is made available. Note that the light direction varies according to (ϕ_i, φ_i), while the view direction according to (ϕ_r, φ_r). Light is distributed considering a hemisphere with the surface at the center of it. The angles ϕ_i and ϕ_r vary with step size $\Delta\phi \in \big(0, \pi/2\big)$, along the elevation direction. While φ_i and φ_r vary with step $\Delta\varphi \in (0, 2\pi)$ along the azimuthal direction. All in all, the total number of rendered objects per BRDF material is $N = 2Sa^2c^2$, with $a = \lceil \frac{\pi}{2\Delta\phi} \rceil + 1$ and $c = \lceil \frac{2\pi}{\Delta\varphi} \rceil$. The set of rendered objects is $\mathcal{B} = \{B_1, \cdots, B_b\}$, with b the number of considered BRDF materials, and each B_i is made of N rendered objects. For the ambient light we used 16 different light probes, see [226].

Segmentation. Images sample are taken from the ImageNet database [209]. ImageNet has plenty of objects of several categories, many of which challenging for 3D modeling in terms of concavity, sharpness and specularity. We have sampled some of them, provided they are not occluded. Each testing image is well segmented, choosing manually a main object of interest. We have implemented the level-set based method of [210], a generalization of the active contours approach considering a multi-level set framework.

7.4 Properties transfer from r-surflets to objects

In this section we address the following problem. Given examples $\mathbf{X}_B \in \mathbb{R}^{h \times N}$ of image patches of shaded surfaces with varying illumination and curvature, about which we know probe, material, normals, and depth, with $\sqrt{h}$ the size of the patch, we wish to recover the normals to the surface of a segmented image I_Q, of an unknown object Q, the

material it is made of, and the probe. To this end we have to establish a correspondence between the patches of the unknown surface I_Q and the patches of the known r-surfaces $\mathbf{X}_B$, in the synthetic database. We can see the problem under the following perspective. If we consider a hierarchy of properties of a patch, such as surface features like depth, normals, probe, and image features, we can see that each group of features is a scattered realization of a multivariate variable with unknown probability distribution, whose density is an infinite mixture. We thus use a nested Dirichlet process mixture as introduced in [211], see also [227, 228], defining prior distributions on recursive data structures. Assuming that samples of specific patches have been collected for each of J distributions and are contained in vector $\mathbf{y} = (y_1, \ldots, y_J)$, here we consider that each one provides a different distribution modeling mixtures for each group of features, though we deliberately neglect a sharing level. We obtain a k-ary tree of infinite mixtures, such that each level provides classification paths for the specific feature set, within which the next level of features is nested. At each level of the hierarchy each mixture component gathers patches of similar appearance, namely we have Z-patches for depth, $\mathbf{n}$-patches for normals, $\mathbf{p}$-patches for probes and F-patches for visual features.

The idea is that a patch of a segmented image I_Q, showing only image features, is classified according to the highest level of the hierarchy. Then, following the path of the corresponding branch of the tree of infinite mixtures, the probe, the normals and the depth of the patch can be recovered, considering the mean representative of the corresponding component. The advantage of this non-parametric Bayesian approach is that even with 10^4, up to 10^5 patches, it is possible to obtain good classification results. Note that at each node of the tree the infinite mixture estimates parameters, hence components, according to reallocated indices of the parents nodes, ensuring interchangeability at each level, along a path. Note that the number of samples that can be used along a path j at level ℓ is about $N(\prod_{i=1}^{j\ell} n_{c_{j\ell}})^{-1}$, with $n_{c_{j\ell}}$ the number of components in the branch at level ℓ.

A hierarchical model is built for each BRDF in the synthetic database (see Section 7.7 for details). For each model $\mathcal{M}_B$, $B \in \mathcal{B}$, at the base level of the hierarchy the mixture components are generated from the Z-patches, at the next level from the $\mathbf{n}$-patches, then the probes $\mathbf{p}$-patches, and the leaves level is generated from the F-patches. Here the F-patches are obtained by mapping the RGB values into a feature space, so as to extract the features coded in their representation, ensuring statistical independence of the data [213, 229]. Autoencoders are a popular computational architecture to learn features from data [230, 231], here we introduce a sparse stacked autoencoder, to obtain the F-patches for each BRDF $B \in \mathcal{B}$, which determines the features size from sparsity.

Distribution linking object image and r-surfaces. Let Y be a multivariate whose density is an infinite Gaussian mixture, with unknown parameters. The nested DPM model we consider is $Y|c_{k,j\ell}, \boldsymbol{\theta}_{k,j\ell} \sim \mathcal{N}(\mu_{c_{k,j\ell}}, \Sigma_{c_{k,j\ell}})$, $k \to \infty$ and $j\ell$ the level on the path j in the tree. Here $c_{k,j\ell}$ indicates the mixture component k, at level ℓ, on the path j and the $\boldsymbol{\theta}_{k,j\ell}$ are in turns independently sampled from an unknown distribution $\boldsymbol{\theta}_{k,j\ell}|G_{j\ell} \sim G_{j\ell}$, on which a Dirichlet process $G_{j\ell} \sim DP(\alpha_\ell G_{0,\ell})$ is placed . Here α_ℓ is the concentration parameter, affecting the number of components that will be generated, and $G_{0,\ell}$ is the base distribution, typically the conjugate prior of the observation distribution (for the DPM at each level in a path, we refer the reader to the recent [157, 154] though the models go back to [46, 152]). Assume, now, that the parameters have been computed for each group of features, that a nested DPMs $\mathcal{M}_B$ is obtained for each $B \in \mathcal{B}$, actually each with 4 levels. Each nested DPM has a number of j-paths according to the

recursive structure induced by the groups of features. Given a nested DPM for each $B \in \mathcal{B}$ we are concerned with the computation of the data likelihood for a realization $\mathbf{h}_{Q_B}$, of a patch X_Q whose BRDF has been identified to be B (see below). Once $P(c_{j\ell} = k_{j\ell}|\mathbf{h}_{Q_B}, \mathcal{M}_B)$, is established for the leaf components at level $\ell = 4$, along the path j then, going back along the path and picking the mean value of the nodes in the path, we obtain the most plausible features $\mathbf{p}$-patch and $\mathbf{n}$-patch matching $\mathbf{h}_{Q_B}$. Note that when the DPM is trained, the realizations of Y are the patch features $\mathbf{h}_B$ of the X_B in the synthetic database. To compute the nested DPM we have used conjugate priors and an extension of [232], see also [154, 233].

Stacked sparse autoencoder for each BRDF. Let $\Omega \subseteq \mathbb{R}^h$ be the data space, H the feature space, and $X \in \Omega$ be a patch. Autoencoders [212, 231] provide a structured representation of the sample data, by estimating an encoding map $f : \Lambda \times \Omega \mapsto H$, and a decoding map $g : H \times \Lambda \mapsto \Omega$. Features generated by an autoencoder $\beta(B)$ take values $\mathbf{h} = f(\Lambda_\beta, X) = \sigma(W_{in}X + \mathbf{b}_{in})$. Optimization for minimizing the loss function is here obtained by the orthant projection method [234, 235]. The result of the optimization for the stacked autoencoder are the parameters $\Lambda_\beta^{(1)} \cup \Lambda_\beta^{(2)}$.

The final features for patches $\mathbf{X}_B$, for $B \in \mathcal{B}$, is $\mathbf{h}_B = \sigma(W_{in}^{(2)}\mathbf{h}_B^{(1)} + \mathbf{b}_1^{(2)} \otimes \mathbf{1}_{1\times M})$, of size $k \times M$; here $\mathbf{h}_B^{(1)} = \sigma(W_{in}^{(1)}\mathbf{X}_B + \mathbf{b}_1^{(1)} \otimes \mathbf{1}_{1\times M})$ are the lighter feature values, and $\otimes$ is the Kronecker product.

On the other hand, let $\mathbf{X}_Q = (X_{Q_1}, \ldots, X_{Q_K}) \in \mathbb{R}^{h\times K}$ be the K patches of I_Q (segmented image of Q). The feature set for I_Q is:

$$
\begin{aligned}
&H_{Q/B} = \\
&\{\mathbf{h}_Q = \sigma(W_{in}^{(2)}\sigma(W_{in}^{(1)}\mathbf{X}_Q + \mathbf{b}_1^{(1)} \otimes \mathbf{1}_{1\times K}) + \mathbf{b}_1^{(2)} \otimes \mathbf{1}_{1\times K}) | \\
&\quad (W_{in}^{(2)}, W_{in}^{(1)}, \mathbf{b}_1^{(2)}, \mathbf{b}_1^{(1)}) \in \Lambda_\beta^{(1)} \cup \Lambda_\beta^{(2)}, \forall B \in \mathcal{B}\}.
\end{aligned}
\tag{7.1}
$$

These features are obtained by evaluating each stacked autoencoder $\beta(B), B \in \mathcal{B}$, at $\mathbf{X}_Q$. To choose one, consider the average features for $B \in \mathcal{B}$: $\mathbf{s} = 1/M \sum_{\forall X_B} \mathbf{h}_B$. Let $\varepsilon(x) = -\log(x)$ be the Burg entropy, then according to [236] we obtain Bregman divergence to measure similarity between the object features and $\mathbf{s}$:

$$
\begin{aligned}
&\mathbf{X}_Q \in B^\star \text{ if } B^\star = \arg\min_B d(\mathbf{X}_Q, B), \quad \text{with} \\
&d(\mathbf{X}_Q, B) = \sum_{\forall \mathbf{h}_Q \in H_{Q/B}} (\varepsilon(\mathbf{s}) - \varepsilon(\mathbf{h}_Q)) - \nabla\varepsilon(\mathbf{h}_Q)(\mathbf{s} - \mathbf{h}_Q).
\end{aligned}
\tag{7.2}
$$

This results in a full identification of the specific BRDF B for each X_Q, as the material of the patch. Once the BRDF B is chosen, the features $\mathbf{h}_Q$ are the specific realizations of the multivariate Y. Hence the nested DPM can be applied, as gathered in the previous paragraph, in order to obtain the sought for properties to be transferred to X_Q.

7.5 Bas-relief modeling of objects

In this section we present the method for modeling an object shape, given the information obtained from the inference, described in Section 7.4. Accordingly, we are given a number of patches $\mathbf{X}_Q$ covering the segmented image of object Q, the normal field transferred from some X_B, and the position of the top left corner within the domain Ω. Note that the patches are not overlapping.

Object modeling using normals and curvatures Here we define a binary mask $A \subset \mathbb{R}^2$ for image I_Q by the mapping $\nu: \Omega \mapsto \{0, 1\}$. The surface, parametrized by the function $\mathbf{w}: A \mapsto \mathbb{R}^3$, where $\mathbf{w}(u, v)$ is the vector $[x(u, v), y(u, v), z(u, v)]^\top$, is obtained by minimizing an energy functional $\mathcal{G}(\mathbf{w})$. The energy functional $\mathcal{G}(\mathbf{w})$ is defined by the first and second fundamental forms [192], and it embeds surface stretching and bending, plus external forces F acting on it [108].

To correctly identify the external forces we compute the mean curvature $\kappa(u, v)$ for each $(u, v) \in A$, given the normal $\mathbf{n}(u, v)$ at each point of the surface, as estimated by the N-DPM, see Section 7.4. The external forces are needed to sculpt the surface inflation and are of the form $F(u, v) = \text{sign}(\kappa(u, v))q\mathbf{n}(u, v)$, with $q \in \mathbb{R}^+$. The scheme for finding the solution $\mathbf{w}(\cdot)$ is based on the Finite Element method, as described in [207], applied to the Euler-Lagrange equations associated to the functional $\mathcal{G}(\mathbf{w})$. Furthermore, we require that each normal to the surface $\mathbf{w}(u, v)$ is a unit vector along $\mathbf{w}_u \times \mathbf{w}_v$, with $\mathbf{w}_u, \mathbf{w}_v$ the partial derivatives of $\mathbf{w}$. These conditions are imposed as follows:

$$\begin{aligned} \mathbf{n}(u, v) \cdot \mathbf{w}_u(u, v) &= 0 \\ \mathbf{n}(u, v) \cdot \mathbf{w}_v(u, v) &= 0. \end{aligned} \tag{7.3}$$

To linearize the constraints in the model parameters, we add to $\mathbf{w}$ further degrees of freedom including partial derivatives: $\hat{\mathbf{w}}(u, v) = [x, y, z, x_u, y_u, z_u, x_v, y_v, z_v]^\top$. The constraints for (u, v), (7.3), can now be formulated as follows:

$$\begin{bmatrix} 0 & 0 & 0 & \mathbf{n}^x & \mathbf{n}^y & \mathbf{n}^z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{n}^x & \mathbf{n}^y & \mathbf{n}^z \end{bmatrix} \hat{\mathbf{w}}(u, v) = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

with $\mathbf{n}^x, \mathbf{n}^y, \mathbf{n}^z$ the components of $\mathbf{n}(u, v)$ in the x, y, z directions. The constraints in linear form can be expressed as a matrix equation $D\mathbf{U} = C$, with $D \in \mathbb{R}^{2\omega \times l}$, $C \in \mathbb{R}^{2\omega \times 1}$, and $\mathbf{U} = [\hat{\mathbf{w}}(u_1, v_1)^\top, \ldots, \hat{\mathbf{w}}(u_\omega, v_\omega)^\top]^\top \in \mathbb{R}^{l \times 1}$ the vector including the total number l of d.o.f. of the system, and ω being the total number of points inside A. The quadratic minimization problem becomes:

$$\min_{\mathbf{U}} \left\{ \mathbf{U}^\top K \mathbf{U} - F^\top \mathbf{U} + (D\mathbf{U} - C)^\top \Gamma (D\mathbf{U} - C) \right\}, \tag{7.4}$$

with $K \in \mathbb{R}^{l \times l}$ the stiffness matrix [207], $F \in \mathbb{R}^{l \times 1}$ the vector of the external forces and $\Gamma \in \mathbb{R}^{2\omega \times 2\omega}$ a diagonal matrix with elements the weight $\gamma_i \in \mathbb{R}$ of each constraint, for $i = 1, \ldots, \omega$, defined as $\Gamma = \text{diag}(\gamma_1, \gamma_1, ..., \gamma_N, \gamma_N)$. To constrain the solution at the boundary ∂A, homogeneous Dirichlet conditions are applied to the PDE problem. Once the solution $\mathbf{U}$ is computed, the surface and corresponding mesh, obtained from the triangulation over A, are reconstructed. Some modeled surfaces are shown in Figure 7.3.

7.6 Photo-consistency and smoothness

To resolve irregularities of the surface due to noise and outliers we refine the initial surface. Function $z(u, v)$ provides the height of the initial surface, as discussed in Section 7.5. We model the image $\hat{I}(z)$ considering the surface $z(u, v)$ rendered with the recovered probe and BRDF. The goal of the surface refinement is to enforce photo-consistency with the given image while smoothing out the initial surface. The photo-consistency error between the modeled image $\hat{I}$ and the shading of the surface I_s in the

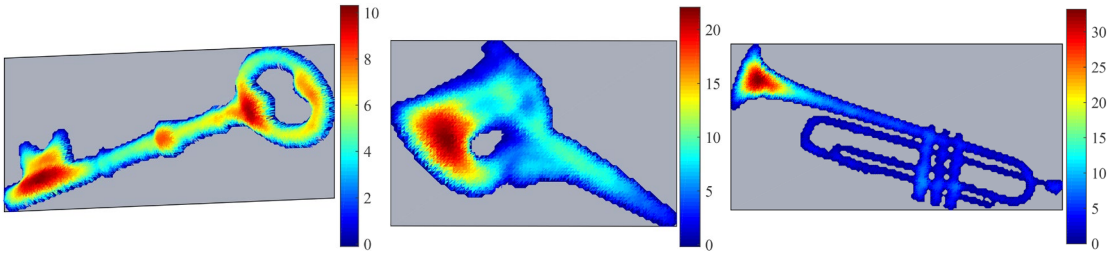


Figure 7.3: Modeled surfaces from segmented images of a key, a mask and a trumpet.

given image is given by

$$E_{photo}(z) = \|I_s - \hat{I}(z)\|_1. \tag{7.5}$$

As we consider objects of specular BRDF, intensity values of the images are strongly affected by the surrounding environment. We considered the reflected environment as a texture modulating the intensities of the imaged object and we approximate the shading image I_s by separating the shading and specularity components of the object via Retinex [215].

Smoothing of the initial surface is achieved by applying total generalized variation (TGV) regularization of the height map $z(u, v)$ corresponding to the initial surface. TGV regularization favors a piece-wise smooth reconstruction of the height map with polynomial terms up to order η [30, 31]. This leads to

$$E_{depth}(z) = TGV^{\eta}(z). \tag{7.6}$$

Finally, to avoid excessive distortion of the surface due to the presence of outliers in the shading image I_s, we require that the normals of the refined surface are similar to the ones of the initial surface. Letting $\mathbf{n}(u, v)$ be the normal of the surface at the point (u, v) and $\mathbf{n}_0(u, v)$ the initial normal at the same point, we consider the following fidelity term

$$E_{norm}(\mathbf{n}) = \|\mathbf{n}(u, v) - \mathbf{n}_0(u, v)\|_1. \tag{7.7}$$

The final surface is obtained by minimizing the resulting energy-like functional, which for TGV^0 is:

$$E(z)=E_{depth}(z) + w_1 E_{photo}(\hat{I}(z)) + w_2 E_{norm}(\mathbf{n}(z)), \tag{7.8}$$

with w_k the weights of the fidelity terms, $k = 1, 2$.

The function (7.8) is non-convex due to the terms E_{photo} and E_{norm}. We relax the problem by considering a local linear approximation of the $\mathcal{S}^2$ manifold as described in [22]. Let $\mathbf{n}_l$ be the linearization point of the normal field and $T = \text{null}(\mathbf{n}_l)$, then $\mathbf{n}(z) = T\nabla z + \mathbf{n}_l$, up to a normalizing constant. Integrability of the normal field [237, 238] is automatically satisfied in this case. The functional of the relaxed problem is:

$$\begin{aligned} E(z, \zeta) = \int_{\Omega} & |\nabla z| + w_1 \|T\nabla z + \mathbf{n}_l - \mathbf{n}_0\| \\ & + \frac{1}{2\theta}(\zeta - z)^2 + w_2 |I_s - \hat{I}(\zeta)| du dv. \end{aligned} \tag{7.9}$$

The auxiliary variable ζ is purposefully added in (7.9) to separate the photo-consistency from the rest of the terms, in so separating the problem into two distinct minimization

sub-problems. At each iteration the minimizer of the photo-consistency term is estimated by point-wise search, while a minimizer with respect to z is identified by primal-dual optimization [32].

Considering the part of (7.9) depending only on z, we obtain its primal-dual form by applying the Legendre-Fenchel transformation. Let $\mathcal{P}$ be the convex set obtained from the union of L_1 balls, D the discretized gradient operator, and $\mathbf{z}, \boldsymbol{\zeta}, \bar{\mathbf{n}}$ the vectorized variables corresponding to $z, \zeta, \mathbf{n}$, respectively, then the primal-dual form of (7.9) is:

$$\max_{\mathbf{p},\mathbf{q}\in\mathcal{P}} \quad \frac{1}{2\theta}\|\boldsymbol{\zeta}^* \;-\; \mathbf{z}\|^2 \;+\; \langle \mathbf{p}, \mathrm{D}\,\mathbf{z}\rangle \;+\; w_1\langle \mathbf{q}, \mathrm{T}\,\mathrm{D}\,\mathbf{z} \;+\; \bar{\mathbf{n}}_l \;-\; \bar{\mathbf{n}}_0\rangle. \quad (7.10)$$

Choosing suitable step sizes $\sigma, \tau > 0$, a saddle point is found by the proximal point iterations summarized below:

$$\begin{aligned}
\mathbf{p}^{(k+1)} &= \Pi_{\mathcal{P}}\left(\mathbf{p}^{(k)} + \tau\,\mathrm{D}\,\hat{\mathbf{z}}^{(k)}\right),\\
\mathbf{q}^{(k+1)} &= \Pi_{\mathcal{P}}\left(\mathbf{q}^{(k)} + \tau w_1(\mathrm{T}^{(k)}\,\mathrm{D}\,\hat{\mathbf{z}}^{(k)} + \bar{\mathbf{n}}_l^{(k)} - \bar{\mathbf{n}}_0)\right),\\
\mathbf{z}^{(k+1)} &= (1+\tfrac{\sigma}{\theta^{(k)}})^{-1}\left(\mathbf{z}^{(k)} + \tfrac{\sigma}{\theta^{(k)}}\boldsymbol{\zeta}^*\right.\\
&\quad \left. - \sigma\,\mathrm{D}^\top(\mathbf{p}^{(k+1)} + w_1 {\mathrm{T}^{(k)}}^\top \mathbf{q}^{(k+1)})\right),\\
\hat{\mathbf{z}}^{(k+1)} &= 2\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)},\\
\bar{\mathbf{n}}_l^{(k+1)} &= \Pi_{S^2}(\mathrm{T}^{(k)}\,\mathrm{D}\,\mathbf{z}^{(k+1)} + \bar{\mathbf{n}}_l^{(k)}),
\end{aligned}$$

with $\mathrm{T}^{(k)}$ a matrix formed by the the null spaces of the corresponding vectors $\bar{\mathbf{n}}_l^{(k)}$, Π_X the projection on set X, and w_k as mentioned in (7.8). θ decreases at each iteration, enforcing the variables $\boldsymbol{\zeta}$ and $\mathbf{z}$ to converge, approximating in this way a solution of the original minimization problem.

The refinement produces smooth surfaces while preserving sharp discontinuities of the initial surface supported by the appearance of the object in the image.

7.7 Experiments and results

Unsupervised learning experiments. We consider the following BRDFs: aluminum, brass, PVC, steel and plastic. For each material up to $N{=}430$ r-surfaces are generated, and about $23.30{\times}10^4$ patches obtained. Transformation of patches into feature space lasts $32.12{\times}10$ sec., for each $\beta(B)$. DPM training lasts about $60.40{\times}10^4$ sec. for each B. This refers to a computer equipped with four Xeon E5-2643 3.7GHz CPUs and 64GB RAM.

MSE prediction error for autoencoders is shown in Figure 7.4. Material choice (eq. 7.2) is 100% correct. To evaluate the accuracy of component prediction for the observed object with the DPM, we use 3D models with computed normals and rendered with BRDF (Figure 7.6). Results are given in Figure 7.5, where the size N of the r-surfaces samples varies from 48 to 430. Mixtures components range from a minimum of $18{\times}10$ to a maximum of $27{\times}10^2$. Ground truth (GT) objects are also used to evaluate the NMSE of mean normals between each X_Q and each representative X_B of the chosen DPM component, Figure 7.5 right.

Synthetic data We examine first the performance of the framework using synthetic images for which the ground truth is available. We render various 3D models using the

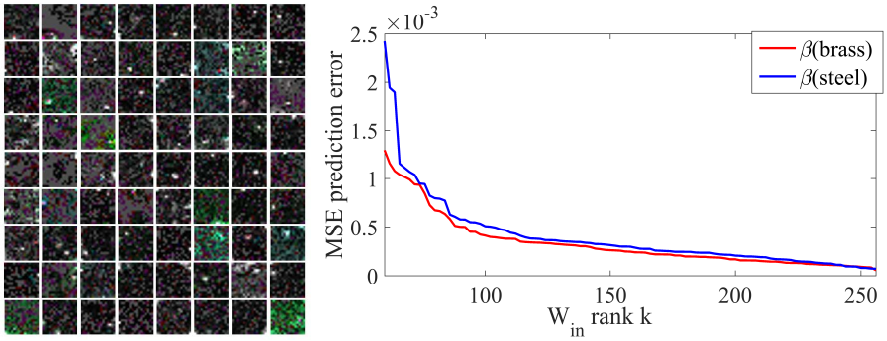


Figure 7.4: On the left the deep features predicted by $\beta(brass)$, with rank k=72, m=256. On the right autoencoders $\beta(steel)$ and $\beta(brass)$ MSE prediction error, according to reduced $W_{in}^{(2)}$ rank. Rank k is varied from a 22.6% reduction, up to no reduction.

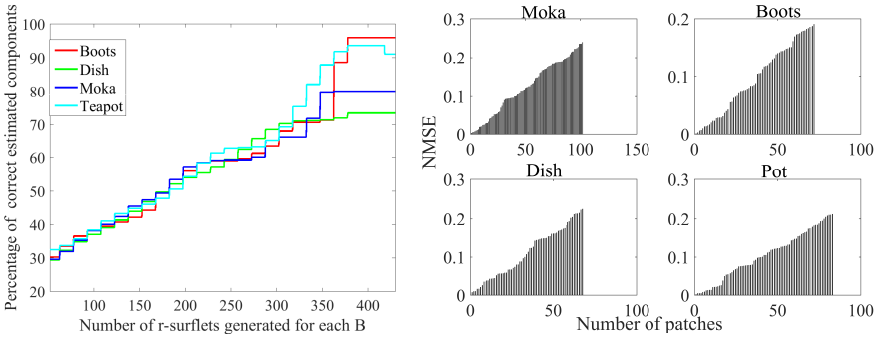


Figure 7.5: On the left components prediction accuracy for the ground truth objects shown in Figure 7.6, varying the size of the sampled r-surfaces. On the right accuracy w.r.t. mean normals.

BRDFs of the materials we consider in this work, taken from the MERL dataset [208]. Renderings using the measured BRDFs are obtained by using a data-driven light closure of the Cycles 3D render engine in Blender. Photorealistic views of the 3D models are composed by using suitable HDR light probe images for simulating surrounding environments. Moreover, we compute the ground truth depth map and the normal map of the rendered object with respect to the current view, by using specialized OSL shaders.

We apply our method on these synthetic views and compare the results with the ground truth. For evaluating the error in the depth field we use the Z-MAE measure [132], normalized with respect to the object bounding box diagonal. For the error of the normal field we use the median angular error (N-MAE) [132] and the mean-squared error of the normal field (N-MSE). The shading error is evaluated using the L-MSE error introduced in [9], considering a window of size 20. Finally, the error between the modeled surface and the GT object is measured using the normalized Hausdorff distance [202]. The average values are computed by taking the geometric mean of the values, as in [132]. The results are shown in Table 7.1, and images of the rendered 3D objects and the surfaces obtained from our method are presented in Figure 7.6. In the same figure, the absolute shading distance and the distance between the meshes are also visualized. The images are best viewed in color and on screen.

The results show that our algorithm produces plausible surfaces of the imaged object from a single image. The material was successfully recognized every time, while the

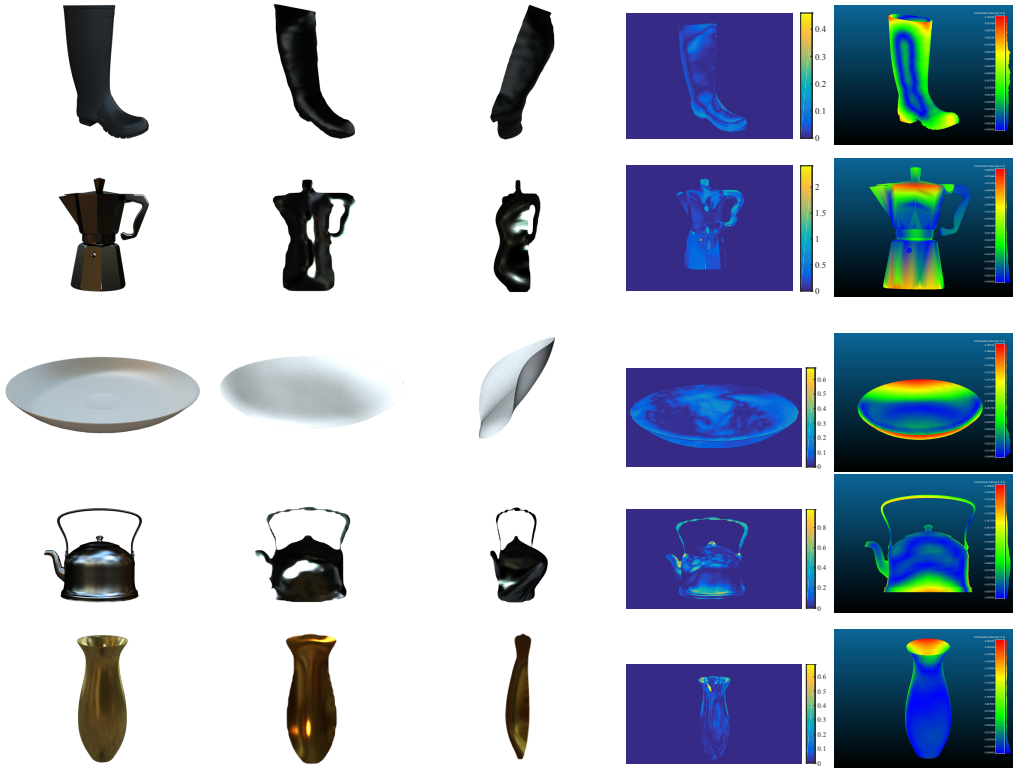


Figure 7.6: Models with ground truth. **1st col.** GT 3D model with BRDF; **2nd col.** modeled surface with BRDF; **3rd col.** rotated view; **4th col.** shading difference; **5th col.** Hausdorff distance.

Object	Z-MAE	N-MAE	N-MSE	L-MSE	Hausdorff	Average
boot	0.0749	0.6397	0.4052	0.0012	0.0460	0.1160
moka pot	0.0632	0.4260	0.2842	0.0808	0.0340	0.0640
dish	0.2434	0.3060	0.2426	0.0009	0.0594	0.0627
teapot	0.1265	0.4325	0.3976	0.0348	0.0713	0.1401
vase	0.0494	0.1737	0.1990	0.0193	0.0721	0.0750
Average	0.0936	0.3626	0.2944	0.0090	0.0544	0.0867

Table 7.1: Synthetic images results.

average value of the median angular error is about 22°. We observe that the shading distance does not always follow the angular and depth error, justifying the use of different error metrics for assessing the modeled surface quality. Three of the objects have significant concave parts (boot, plate, vase) which are evident also in the modeled surfaces. Finally, we see that the metallic objects although showing an increased shading error, due to residual reflections of the environment, are still modeled faithfully, according to the shape metrics.

MIT dataset For an evaluation of our method with respect to publicly available data we use the MIT intrinsic image dataset [9], as augmented in [132] to include the shape of each object. We consider the objects *apple, potato, teabag1, teabag2, paper1* as they exhibit specularity and/or concavities. The objects of this dataset are made of different materials with respect to the ones existing in the MERL BRDF dataset. To overcome

this problem we combine the shading and specularity images of the objects to obtain new composite images without texture. The algorithm recognizes *plastic* as the most similar material to the shaded-only object. Figure 7.7 shows the reference images and the modeled surfaces for each object of the dataset.

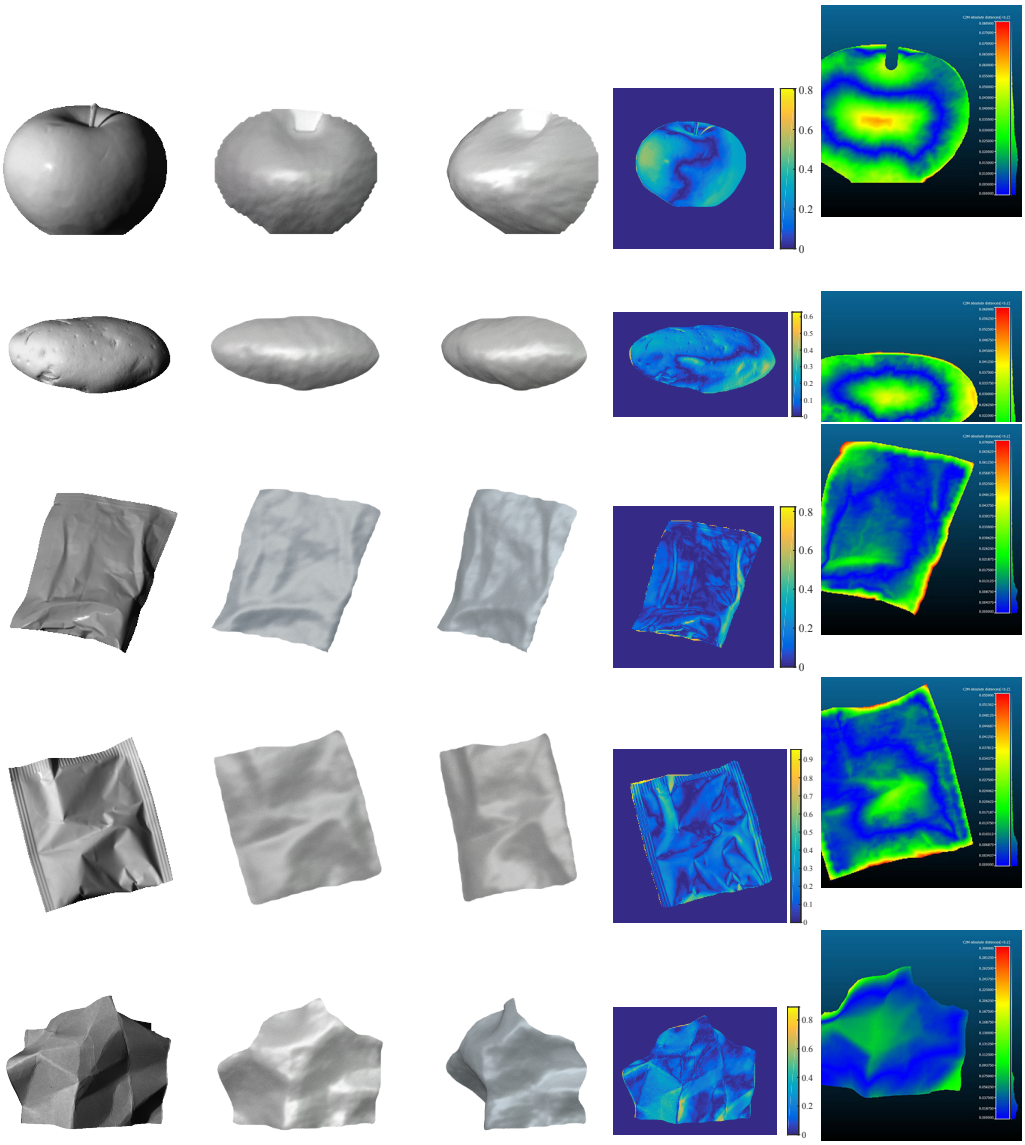


Figure 7.7: MIT dataset. **1st col.** reference image; **2nd col.** modeled surface with BRDF; **3rd col.** rotated view; **4th col.** shading distance (L-MSE); **5th col.** Hausdorff distance.

Algorithm	**Z-MAE**	**N-MAE**	**S-MSE**	**L-MSE**	**Avg.**
Ours	**7.0197**	**0.2692**	0.0261	0.0174	0.1712
Ours no FC no S	26.9816	0.5872	0.0394	0.0217	0.3412
Ours only contour (SfC)	37.1768	0.7728	-	-	-
Retinex+SIFS[132]	17.1914	0.9361	0.0006	0.0019	0.0654
SIFS[132] (grey, lab. light)	20.1445	0.9772	**0.0005**	**0.0017**	**0.0640**

Table 7.2: Results of full and ablated model on MIT dataset [9].

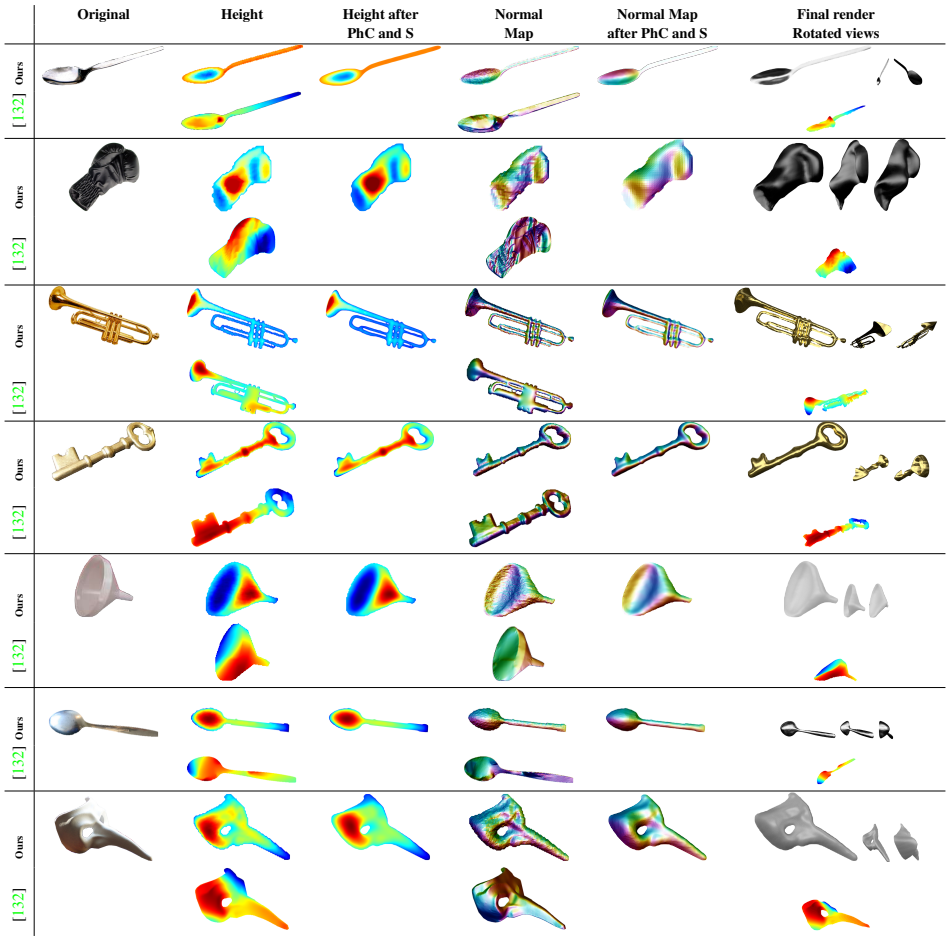


Figure 7.8: Visual comparison between height and normal maps estimated before and after the photo-consistency (PhC) and smoothing (S). Visual comparison with [132] for the height and normal maps.

Table 7.2 compares our results with [132]. As the input images are albedo-less, SIFS [132] was used as a baseline. For the comparison the Z-MAE metric is reported with no normalization and the S-MSE metric [132] is also considered. On one hand the results show that SIFS achieves better results on shading metrics. This is reasonable, since [132] directly optimizes over the rendering error, while in our approach photo-consistency is sought after shape has been recovered. Still, our method achieves higher accuracy on shape metrics, since it primarily recovers the surface normals. On comparing the shape recovered with the two approaches one can notice that [132], due to the Lambertian assumption, distorts shape near reflections and specularities, trying to interpret intensity changes as changes in shape. Additionaly, [132] cannot always capture concavity of the surface (e.g. the bowl of the spoon in Figure 7.8). Note that in Table 7.2 we considered also a pre-processing with Retinex before applying SIFS, which helps in reducing specularities, leading to better results in terms of shape, slightly penalizing the shading distance. Table 7.2 presents also ablated versions of our method, highlighting the importance of surface refinement.

Modeling of ImageNet objects We have manually selected from the ImageNet dataset [209] images of objects made from the materials described above. The 3D surfaces of the visible parts of these objects are computed with the proposed framework. Figure 7.8

shows the selected images together with renderings of the recovered surface as well as the computed depth and normal maps before and after refinement. Comparison with the results of [132] is also provided. We observe that the modeled surfaces closely resemble the reference objects, when viewed from the image vantage point with the recognized probe and BRDF. This is also evident by the values of the shading difference and the L-MSE metric, reported in Table 7.3.

Algorithm	Concave spoon	Glove	Trumpet	Key	Funnel	Convex spoon	mask	**Average**
Ours	0.0792	0.0559	0.0571	0.0271	0.0189	0.0321	0.471	0.0570
[132] (color, natural ill.)	0.0669	0.0097	0.1600	0.0204	0.0072	0.0337	0.0077	0.0169

Table 7.3: L-MSE for ImageNet objects.

7.8 Conclusions

We proposed a novel approach for BRDF aware modeling of 3D objects from a single image. The contributions of this work are twofold. On the one side, we are able to fully model non-Lamberitan surfaces with either concave or sharp parts, with limited error both in shading and shape. On the other side, we have proved that the normal field of the surfaces to be modeled can be learned from renderings of different objects surfaces. The contribution builds on three main achievements. The first, is that we can represent the material reflectance and specular properties, basing on deep features, as a hierarchy of features that can be transferred via a nested Dirichlet process mixture to an unknown surface. The second, is that the normal field can be used to define an external force needed to sculpt a deformed surface into a refined shape representation of the unknown object. Finally, we contribute with a new method based on TGV to enforce photo-consistency between the generated surface and the appearance of the object in the image. These results prove to be very promising, despite the fact that the whole process seems to be still complex and time demanding.

In future work we will examine the steps needed to retrieve the geometry of the full object, even if a prior is needed. Moreover, we will extend the categories our model can accommodate and simplify the whole framework.

Chapter 8

Conclusions - Future work

In this thesis we have discussed various applications of inverse problem theory on computer vision problems related with the modeling and reconstruction of shape and action. We have seen that a large number of problems in the field of computer vision are ill-posed problems. Inverse problem theory provides a powerful mathematical framework for dealing with these problems.

We have considered various shape and action modeling tasks, as well as saliency modeling, and we have introduced novel methods for dealing with this tasks in the context of inverse problem theory. In particular, we have considered two types of inverse problem methods, namely regularization methods, and Bayesian statistical methods. The choice of which type of methods fits best, depends on the proposed modeling of the problem. For some challenging problems, like the case of modeling specular surfaces from a single image, we have also considered a combination of these two types of approaches.

Regarding saliency modeling, we have proposed a modeling inspired by the coherence theory of attention, which simulates the generation of proto-objects based on geometric and photometric properties of the scene, using vibrating thin-membranes.

We have also dealt with the problem of variational fusion of images, with particular application on the fusion of multiple depth images. The model introduced allows for the estimation of confidence values based on the given data, and results to a spatially adaptive regularization of the depth images, allowing confidence regions to be less regularized with respect to less confident regions. The final solution is estimated using variational methods.

Regarding action modeling, we have considered here the problem of recognizing human actions from 3D data. In particular by knowing the 3D pose of a subject in a small set of frames while she/he performs a specific action, we compute suitable features for each part of the body, which are then clustered using a non-parametric Bayesian mixture model in order to capture their idiosyncratic behaviors. By comparing these behaviors with a dictionary corresponding to each action, the action performed by the subject is recognized.

We have also studied the problem of computing 3D models of articulated objects from images taken from the Web, with particular application on modeling of animals. In this context a decomposition of the object in components is considered, and each component is modeled using different aspects, as captured in the set of available images. The modeling task is treated using a finite element approach, while the registration problems involved in the process are solved using appropriate optimization schemes.

The last modeling problem considered, regards the modeling of 3D specular surfaces from a single image. Here, we consider a non-parametric Bayesian model for modeling and estimating the surface normals, according to the appearance of the surface. The initial estimation achieved by this model is then used to obtain an initial shape of the surface, which is then refined using a regularization scheme based on the smoothness of the final normal field and a photo-consistency constraint.

In all these methods, there are various extensions which can be pursued, which have been discussed in the individual chapters. As briefly noted in the introduction, some more general open problems in the field of inverse problem theory and its applications in computer vision problems can be further explored. One of the most important, and most challenging, is the application of Bayesian statistical methods on infinite dimensional spaces, considering a wide spectrum of probability distributions. This is an important case as we often need to estimate functions which lie in infinite dimensional spaces, as we have also seen in this thesis.

Another interesting research direction regards the application of hierarchical non-parametric Bayesian models for shape, motion, and action modeling problems in computer vision. We have encountered in this thesis some examples of how non-parametric Bayesian models can be applied for dealing with these problems, and we have seen that they are very powerful in capturing the main properties been sough for by using appropriate features. Finally, we have seen that by combining non-parametric Bayesian methods with regularization methods we are able to introduce frameworks which can deal with very challenging modeling problems, like the modeling of specular surfaces from a single image.

Bibliography

[1] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques*, pp. 311–318, ACM, 1994. viii, 73, 81

[2] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques*, pp. 303–312, ACM, 1996. viii, 73, 81

[3] V. Krishnamurthy and M. Levoy, "Fitting smooth surfaces to dense polygon meshes," in *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques*, pp. 313–324, ACM, 1996. viii, 73, 81

[4] A. S. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1584–1601, 2006. viii, 73, 82

[5] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3D Reconstruction in Real-time," in *Intelligent Vehicles Symposium*, 2011. ix, 78, 79, 80

[6] V. Ntouskos, F. Pirri, M. Pizzoli, A. Sinha, and B. Cafaro, "Saliency prediction in the coherence theory of attention," *Biologically Inspired Cognitive Architectures*, vol. 5, pp. 10–28, 2013. ix, 78, 79

[7] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Proceedings of the Asian Conference on Computer Vision*, 2010. ix, 78, 79, 80

[8] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1592–1599, 2015. ix, 79, 80

[9] R. Grosse, M. Johnson, E. H. Adelson, and W. Freeman, "Ground truth dataset and baseline evaluations for intrinsic image algorithms," in *Proceedings of the International Conference on Computer Vision*, pp. 2335–2342, 2009. ix, 122, 123, 124

[10] J. Hadamard, *Lectures on Cauchy's Problem in Linear Partial Differential Equations*. Yale University Press, 1923. 1, 2

[11] T. Poggio and C. Koch, "Ill-posed problems in early vision: from computational theory to analogue networks," *Proceedings of the Royal society of London. Series B. Biological sciences*, vol. 226, no. 1244, pp. 303–323, 1985. 1

[12] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, 2005. 1

[13] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, vol. 160. Springer Science & Business Media, 2006. 1, 2

[14] M. B. Christopher, *Pattern Recognition and Machine Learning*. Springer, New York, 2006. 4, 27

[15] T. Helin and M. Burger, "Maximum a posteriori probability estimates in infinite-dimensional bayesian inverse problems," *Inverse Problems*, vol. 31, no. 8, p. 085009, 2015. 4

[16] F. Pirri, M. Pizzoli, and A. Rudi, "A general method for the point of regard estimation in 3D space," in *Proceedings of the IEEE Converence on Computer Vision and Pattern Recognition*, pp. 921–928, 2011. 5, 35, 38, 50

[17] R. Rensink, "The dynamic representation of scenes," *Visual Cognition*, vol. 7, pp. 17–42, 2000. 5, 32, 33, 34

[18] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: a survey and extensions," *Mathematical Methods of Operations Research*, vol. 66, no. 3, pp. 373–407, 2007. 6, 16, 60, 63, 64

[19] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka-Lojasiewicz inequality," *Mathematics of Operations Research*, vol. 35, no. 2, pp. 438–457, 2010. 6, 59, 60, 65

[20] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011. 6, 22

[21] F. E. Nicodemus, "Directional reflectance and emissivity of an opaque surface," *Applied optics*, vol. 4, no. 7, pp. 767–775, 1965. 7, 113, 114, 116

[22] B. Zeisl, C. Zach, and M. Pollefeys, "Variational regularization and fusion of surface normal maps," in *3DV*, vol. 1, pp. 601–608, 2014. 8, 120

[23] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, vol. 317. Springer Science & Business Media, 2009. 11

[24] H. Attouch, G. Buttazzo, and G. Michaille, *Variational Analysis in Sobolev and BV Spaces: Applications to PDEs and Optimization*, vol. 17. SIAM, 2014. 11, 17

[25] D. P. Bertsekas, *Convex Optimization Theory*. Athena Scientific Belmont, 2009. 12, 13, 14

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004. 12, 61

[27] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Science & Business Media, 2011. 14, 66, 68

[28] J. Müeller, *Advanced Image Reconstruction and Denoising - Bregmanized (Higher Order) Total Variation and Application in PET*. PhD thesis, Institute for Computational and Applied Mathematics, University of Münster, 2013. 15, 21, 59

[29] M. Artina, M. Fornasier, and F. Solombrino, "Linearly constrained nonsmooth and nonconvex minimization," *SIAM Journal on Optimization*, vol. 23, no. 3, pp. 1904–1937, 2013. 16, 59

[30] K. Bredies, K. Kunisch, and T. Pock, "Total Generalized Variation," *SIAM Journal Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010. 17, 18, 21, 59, 120

[31] M. Burger and S. Osher, "A guide to the TV zoo," in *Level Set and PDE Based Reconstruction Methods in Imaging*, pp. 1–70, Springer, 2013. 17, 120

[32] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, "An introduction to total variation for image analysis," *Theoretical foundations and numerical methods for sparse recovery*, vol. 9, pp. 263–340, 2010. 17, 18, 19, 21, 57, 69, 121

[33] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992. 18, 58, 59, 69, 81, 82, 83

[34] M. Burger and S. Osher, "Convergence rates of convex variational regularization," *Inverse problems*, vol. 20, no. 5, p. 1411, 2004. 18

[35] T. F. Chan and S. Esedoglu, "Aspects of total variation regularized L1 function approximation," *SIAM Journal on Applied Mathematics*, vol. 65, no. 5, pp. 1817–1837, 2005. 18, 69

[36] A. Chambolle and P. L. Lions, "Image recovery via total variation minimization and related problems," *Numerische Mathematik*, vol. 76, no. 2, pp. 167–188, 1997. 18, 59

[37] B. He and X. Yuan, "Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective," *SIAM Journal on Imaging Sciences*, vol. 5, no. 1, pp. 119–149, 2012. 20

[38] T. C. Wittman, *Variational Approaches to Digital Image Zooming*. PhD thesis, University of Minnesota, 2006. 20

[39] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *Journal of Optimization Theory and Applications*, vol. 158, no. 2, pp. 460–479, 2013. 21, 57, 66, 68

[40] M. Zhu and T. Chan, "An efficient primal-dual hybrid gradient algorithm for total variation image restoration," Tech. Rep. 08–34, UCLA Cam report, 2008. 21, 57

[41] J. Stillwell, *Naive Lie Theory*. Springer Science & Business Media, 2008. 23, 26

[42] H. Strasdat, *Local Accuracy and Global Consistency for Efficient SLAM*. PhD thesis, Imperial College London, 2012. 23

[43] J. K. Ghosh and R. V. Ramamoorthi, *Bayesian Nonparametrics*. Springer Series in Statistics, Springer, 2003. 26

[44] P. Orbanz and Y. W. Teh, "Bayesian nonparametric models," in *Encyclopedia of Machine Learning*, pp. 81–89, Springer, 2011. 26

[45] Y. W. Teh, "Dirichlet process," in *Encyclopedia of machine learning*, pp. 280–287, Springer, 2011. 26

[46] T. S. Ferguson, "A bayesian analysis of some nonparametric problems," *Annals of Statistics*, pp. 209–230, 1973. 27, 91, 92, 114, 117

[47] J. Sethuraman, "A constructive definition of dirichlet priors," *Statistica sinica*, pp. 639–650, 1994. 28

[48] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249–265, 2000. 30, 92

[49] J. Tsotsos, S. Culhane, W. Wai, Y. Lai, N. Davis, and F. Nuflo, "Modeling visual attention via selective tuning," *Artifical Intelligence*, vol. 78, pp. 507 – 547, 1995. 31, 32

[50] C. Koch and S. Ullman, "Shifts in selective visual-attention: towards the underlying neural circuitry," *Human Neurobiology*, vol. 4, no. 4, pp. 219–227, 1985. 32

[51] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998. 32

[52] T. Minato and M. Asada, "Image feature generation by Visio-Motor Map Learning towards selective attention," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1422–1427, 2001. 32

[53] A. Belardinelli, F. Pirri, and A. Carbone, "Bottom-up gaze shifts and fixations learning by imitation," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 37, pp. 256–271, 2007. 32

[54] R. Carmi and L. Itti, "Visual causes versus correlates of attentional selection in dynamic scenes," *Vision Research*, vol. 46, no. 26, pp. 4333 – 4345, 2006. 32

[55] N. J. Butko, L. Zhang, G. W. Cottrell, and J. R. Movellan, "Visual saliency model for robot cameras," in *IEEE International Conference on Robotics and Automation*, pp. 2398–2403, 2008. 32

[56] M. Mancas, F. Pirri, and M. Pizzoli, "From saliency to eye gaze: Embodied visual selection for a pan-tilt-based robotic head," in *International Symposium on Visual Computing*, pp. 135–146, 2011. 32

[57] E. Pichon and L. Itti, "Real-time high-performance attention focusing for outdoors mobile beobots," in *Proceedings of AAAI Spring Symposium (AAAI-TR-SS-02-04)*, p. 63, 2002. 32

[58] C. Ackerman and L. Itti, "Robot steering with spectral image information," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 247–251, 2005. 32

[59] H. Hügli, T. Jost, and N. Ouerhani, "Model performance for visual attention in real 3d color scenes," in *International Work-conference on the Interplay between Natural and Artificial Computation*, pp. 469–478, 2005. 32

[60] M. Cerf, J. Harel, W. Einhäuser, and C. Koch, "Predicting human gaze using low-level saliency combined with face detection," in *NIPS*, 2007. 32

[61] P. L. Sala, R. Sim, A. Shokoufandeh, and S. J. Dickinson, "Landmark selection for vision-based navigation," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 334–349, 2006. 32

[62] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 171–177, 2010. 32

[63] A. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, pp. 97–136, 1980. 32

[64] J. Duncan and G. W. Humphreys, "Visual search and stimulus similarity.," *Psychological review*, vol. 96, no. 3, pp. 433–458, 1989. 32, 34

[65] D. Marr and H. K. Nishihara, "Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 200, pp. 269–294, 1978. 32

[66] J. M. Wolfe, "The parallel guidance of visual attention," *Current Directions in Psychological Science*, vol. 4, pp. 124–128, 1992. 32

[67] J. M. Wolfe, S. R. Friedman-Hill, M. L. Stewart, and K. M. O'Connell, "The role of categorization in visual search for orientation," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 18, pp. 34–49, 1992. 32

[68] J. M. Wolfe, "Guided Search 2.0. A revised model of visual search," *Psychonomic Bulletin and Review*, vol. 2, pp. 202–238, 1994. 32

[69] A. Treisman, "Preattentive processing in vision," *Computer Vision, Graphics and Image Processing*, vol. 31, no. 2, pp. 156–177, 1985. 32

[70] U. Neisser and R. Becklen, "Selective looking: Attending to visually specified events," *Cognitive Psychology*, vol. 7, no. 4, pp. 480 – 494, 1975. 32

[71] B. Julesz, "Texton gradients: The texton theory revisited," *Biological Cybernetics*, vol. 54, pp. 245–251, 1986. 32

[72] R. Rensink, J. K. O'Regan, and J. J. Clark, "On the failure to detect changes in scenes across brief interruptions," *Visual Cognition*, vol. 7, pp. 127–145, 2000. 33, 55

[73] R. Rensink, "Change detection," *Annual Review of Psychology*, vol. 53, pp. 245–277, 2002. 33

[74] R. Rensink, J. K. O'Regan, and J. J. Clark, "To see or not to see: The need for attention to perceive changes in scenes," *Psychological Science*, vol. 8, pp. 368–373, 1997. 33

[75] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 9, pp. 1395–1407, 2006. 33

[76] F. Orabona, G. Metta, and G. Sandini, "A proto-object based visual attention model," in *Attention in Cognitive Systems. Theories and Systems from an Interdisciplinary Viewpoint* (L. Paletta and E. Rome, eds.), pp. 198–215, Berlin, Heidelberg: Springer-Verlag, 2008. 33

[77] J. T. Serences and S. Yantis, "Selective visual attention and perceptual coherence.," *Trends in cognitive sciences*, vol. 10, no. 1, pp. 38–45, 2006. 34

[78] T. Bahill and L. Stark, "The trajectories of saccadic eye movements," *Scientific American*, vol. 240, no. 1, pp. 1–12, 1979. 34

[79] T. Bahill, K. A. Bahill, M. Clark, and L. Stark, "Closely spaced saccades," *Investigative Ophtalmology*, vol. 14, no. 4, pp. 317–321, 1975. 34

[80] W. Zhou, X. Chen, and J. Enderle, "An updated time-optimal 3rd-order linear saccadic eye plant model," *International Journal of Neural Systems*, vol. 19, no. 5, 2009. 34

[81] E. Kowler, "Eye movements: The past 25 years," *Vision Researcn*, pp. 1–27, Jan 2011. 34

[82] M. Pizzoli, D. Rigato, R. Shabani, and F. Pirri, "3D Saliency maps," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 9 – 14, 2011. 35, 50

[83] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 36, 37, 38, 39

[84] O. Faugeras, Q. Luong, and T. Papadopoulou, *The Geometry of Multiple Images*. MIT Press, 2001. 36

[85] B. Triggs, P. F. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *Proceedings of the International Conference on Computer Vision*, 2000. 36

[86] P. Fiore, "Efficient linear solution of exterior orientation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 140–148, 2002. 37

[87] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981. 37

[88] P. H. S. Torr, "Geometric motion segmentation and model selection," *Philosophical Transactions of the Royal Society of London A*, vol. 356, no. 1740, pp. 1321–1340, 1998. 39

[89] M. Lourakis and A. Argyros, "SBA: A software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software*, vol. 36, no. 1, p. 2, 2009. 40

[90] M. Hegland, S. Roberts, and I. Altas, "Finite element thin plate splines for surface fitting," Tech. Rep. TR-CS-97-20, Department of Computer Science, Faculty of Engineering and Information Technology, The Australian National University Canberra, 1997. 42

[91] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, no. 7-8, pp. 1157–1182, 2003. 43, 53

[92] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 1995. 43

[93] O. L. Mangasarian, "Exact 1-norm support vector machines via unconstrained convex differentiable minimization," tech. rep., Data Mining Institute TR 05-03, 2005. 43

[94] N. Cristianini and J. Shawe-Taylor, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. 43

[95] A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA, 2000. 43

[96] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," *Advances in Neural Information Processing Systems*, vol. 16, 2003. 43

[97] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computing*, vol. 13, pp. 1443–1471, 2001. 45

[98] O. L. Mangasarian and E. W. Wild, "Feature selection for nonlinear kernel support vector machines," in *IEEE International Conference on Data Mining Workshops*, pp. 231–236, 2007. 53

[99] J. D'Errico, "Surface fitting using gridfit," tech. rep., Matlab File Exchange, 2013. 54

[100] D. Strong and T. Chan, "Edge-preserving and scale-dependent properties of total variation regularization," *Inverse Problems*, vol. 19, no. 6, p. 165, 2003. 58, 69

[101] D. Calvetti and E. Somersalo, "Hypermodels in the bayesian imaging framework," *Inverse Problems*, vol. 24, no. 3, p. 034013, 2008. 58

[102] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990. 59

[103] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2320–2327, 2011. 59, 72

[104] G. Kuschk and D. Cremers, "Fast and accurate large-scale stereo reconstruction using variational methods," in *ICCV Workshop on Big Data in 3D Computer Vision*, 2013. 59, 72

[105] E. Esser and X. Zhang, "Nonlocal patch-based image inpainting through minimization of a sparsity promoting nonconvex functional," tech. rep., Dept. Math., Univ. California Irvine, Irvine, CA, USA, 2014. 59, 69

[106] P. Ochs, A. Dosovitskiy, T. Brox, and T. Pock, "An Iterated L1 Algorithm for Non-smooth Non-convex Optimization in Computer Vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1759–1766, 2013. 59, 60

[107] J. Lellmann, E. Strekalovskiy, S. Koetter, and D. Cremers, "Total variation regularization for functions with values in a manifold," in *Proceedings IEEE International Conference on Computer Vision*, pp. 2944–2951, 2013. 59

[108] V. Ntouskos, M. Sanzari, B. Cafaro, F. Nardi, F. Natola, F. Pirri, and M. Ruiz Garcia, "Component-wise modeling of articulated objects," in *Proceedings of the International Conference on Computer Vision*, 2015. 59, 80, 119

[109] F. Natola, V. Ntouskos, F. Pirri, and M. Sanzari, "Single image object modeling based on brdf and r-surfaces learning," in *Proceedings of the IEEE Converence on Computer Vision and Pattern Recognition*, 2016. 59, 80

[110] M. Nikolova, M. K. Ng, and C.-P. Tam, "Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3073–3088, 2010. 59

[111] P. Ochs, Y. Chen, T. Brox, and T. Pock, "iPiano: Inertial proximal algorithm for nonconvex optimization," *SIAM Journal on Imaging Sciences*, vol. 7, no. 2, pp. 1388–1419, 2014. 59

[112] T. Möllenhoff, E. Strekalovskiy, M. Moeller, and D. Cremers, "The primal-dual hybrid gradient method for semiconvex splittings," *SIAM Journal on Imaging Sciences*, vol. 8, no. 2, pp. 827–857, 2015. 59, 69

[113] T. Valkonen, "A primal–dual hybrid gradient method for nonlinear operators with applications to MRI," *Inverse Problems*, vol. 30, no. 5, p. 055012, 2014. 60

[114] J.-P. Aubin and H. Frankowska, *Set-valued Analysis*. Springer Science & Business Media, 2009. 60

[115] N. D. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla, "Using multiple hypotheses to improve depth-maps for multi-view stereo," in *Proceedings of the European Conference on Computer Vision*, pp. 766–779, Springer, 2008. 60

[116] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J. Frahm, R. Yang, D. Nister, and M. Pollefeys, "Real-Time Visibility-Based Fusion of Depth Maps," in *Proceedings of the International Conference on Computer Vision*, pp. 1–8, IEEE, 2007. 60

[117] C. Hane, C. Zach, B. Zeisl, and M. Pollefeys, "A patch prior for dense 3d reconstruction in man-made environments," in *Proceedings of the International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pp. 563–570, IEEE, 2012. 60

[118] T. Pock, L. Zebedin, and H. Bischof, "TGV-Fusion," *Rainbow of Computer Science*, vol. 6570, pp. 245–258, 2011. 60, 74, 75, 76, 81, 82, 83

[119] D. Ferstl, R. Ranftl, M. Ruther, and H. Bischof, "Multi-modality depth map fusion using primal-dual optimization," in *Proceedins of the International Conference on Computational Photography*, pp. 1–8, 2013. 60

[120] C. Zach, T. Pock, and H. Bischof, "A Globally Optimal Algorithm for Robust TV-L1 Range Image Integration," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1–8, 2007. 60

[121] S. Fuhrmann and M. Goesele, "Fusion of Depth Maps with Multiple Scales," *ACM Transactions on Graphics*, vol. 30, no. 6, p. 148, 2011. 60

[122] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Proceedings of the International Symposium on Mixed and Augmented Reality*, pp. 127–136, 2011. 60

[123] B. Ummenhofer and T. Brox, "Global, dense multiscale reconstruction for a billion points," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1341–1349, 2015. 60

[124] A. P. James and B. V. Dasarathy, "Medical image fusion: A survey of the state of the art," *Information Fusion*, vol. 19, pp. 4–19, 2014. 60

[125] C. Lanaras, E. Baltsavias, and K. Schindler, "Advances in hyperspectral and multispectral image fusion and spectral unmixing," *Interanational Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-3/W3, pp. 451–458, 2015. 60

[126] M. Hanke and T. Raus, "A general heuristic for choosing the regularization parameter in ill-posed problems," *SIAM Journal on Scientific Computing*, vol. 17, no. 4, pp. 956–972, 1996. 63

[127] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, vol. 375. Springer Science, 1996. 63

[128] R. E. Wendell and A. P. Hurter Jr, "Minimization of a non-separable objective function subject to disjoint constraints," *Operations Research*, vol. 24, no. 4, pp. 643–657, 1976. 63

[129] M. Nikolova, "A variational approach to remove outliers and impulse noise," *Journal on Mathematical Imaging and Vision*, vol. 20, no. 1-2, pp. 99–120, 2004. 69

[130] F. Gney and A. Geiger, "Displets: Resolving stereo ambiguities using object knowledge," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 73

[131] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 73, 78

[132] J. T. Barron and J. Malik, "Shape, illumination, and reflectance from shading," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 8, pp. 1670–1687, 2015. 73, 113, 122, 123, 124, 125, 126

[133] H. Hirschmuller, "Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 807–814, 2005. 78, 79

[134] D. Gong and G. Medioni, "Dynamic manifold warping for view invariant action recognition," in *Proceedings of the International Conference on Computer Vision*, IEEE, 2011. 85, 86, 87, 95, 96, 97

[135] F. Lv and R. Nevatia, "Recognition and segmentation of 3-d human action using hmm and multi-class adaboost," in *Proceedings of the European Conference on Computer Vision*, Springer, 2006. 85, 86

[136] M. T. Harandi, M. Salzmann, and R. Hartley, "From manifold to manifold: geometry-aware dimensionality reduction for spd matrices," in *Proceedings of the European Conference on Computer Vision*, Springer, 2014. 85

[137] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (smij): A new representation for human skeletal action recognition," in *CVPRW, 2012 IEEE Computer Society Conference on*, 2012. 85, 86

[138] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning actionlet ensemble for 3d human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. 85, 86

[139] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 85, 86, 90

[140] D. Gong, G. Medioni, and X. Zhao, "Structured time series analysis for human action segmentation and recognition," in *IEEE Transactions on PAMI*, IEEE Computer Society, 2014. 86, 87, 91, 95

[141] X. Yang and Y. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, 2012. 86

[142] S. Fothergill, H. M. Mentis, S. Nowozin, and P. Kohli, "Instructing people for training gestural interactive systems," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2012. 86

[143] V. Bloom, D. Makris, and V. Argyriou, "G3d: A gaming action dataset and real time action recognition evaluation framework," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012. 86

[144] L. Seidenari, V. Varano, S. Berretti, A. Del Bimbo, and P. Pala, "Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2013. 86

[145] O. Oreifej and Z. Liu, "Hon4D: Histogram of oriented 4D normals for activity recognition from depth sequences," in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2013. 86

[146] M. Spivak, "Differential geometry, volume 1–5," *Publish or Perish, Berkeley*, 1975. 87

[147] F. Flaherty and M. do Carmo, *Riemannian Geometry*. Mathematics: Theory & Applications, Birkhäuser Boston, 2013. 87

[148] M. Zefran, V. Kumar, and C. Croke, "Choice of riemannian metrics for rigid body kinematics," in *ASME 24th Biennial Mechanisms Conference*, 1996. 88

[149] P. Fletcher, C. Lu, S. Pizer, and S. Joshi, "Principal geodesic analysis for the study of nonlinear statistics of shape," *IEEE Transactions on Medical Imaging*, 2004. 88

[150] J. Manton, "A globally convergent numerical algorithm for computing the centre of mass on compact lie groups," in *Proceedings of the International Conference on Control, Automation, Robotics and Vision*, 2004. 88

[151] H. Karcher, "Riemannian center of mass and mollifier smoothing," *Communications on pure and applied mathematics*, 1977. 88

[152] C. E. Antoniak, "Mixtures of dirichlet processes with applications to bayesian nonparametric problems," *Annals of Statistics*, pp. 1152–1174, 1974. 91, 117

[153] D. Görür and C. E. Rasmussen, "Dirichlet process gaussian mixture models: Choice of the base distribution," *Journal of Computer Science and Technology*, 2010. 92

[154] E. B. Sudderth, *Graphical Models for Visual Object Recognition and Tracking*. PhD thesis, MIT, 2006. 92, 117, 118

[155] M. D. Escobar and M. West, "Bayesian density estimation and inference using mixtures," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 577–588, 1995. 92

[156] D. M. Cifarelli and E. Regazzini, "Distribution functions of means of a dirichlet process," *Annals of Statistics*, 1990. 92

[157] D. M. Blei and M. I. Jordan, "Variational inference for dirichlet process mixtures," *Bayesian Analysis*, vol. 1, no. 1, pp. 121–143, 2006. 92, 117

[158] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation Mocap Database HDM05," Tech. Rep. CG-2007-2, Universität Bonn, June 2007. 93

[159] http://mocap.cs.cmu.edu, 2015. 93

[160] http://www.cs.ucf.edu/~oreifej/HON4D.html, 2015. 93

[161] http://research.microsoft.com/en-us/um/cambridge/projects/msrc12/, 2015. 93

[162] http://www.micc.unifi.it/vim/datasets/3dactions/, 2015. 93

[163] http://dipersec.kingston.ac.uk/G3D/, 2015. 93

[164] T. Binford, "Visual perception by computer," in *IEEE conference on Systems and Control*, vol. 261, p. 262, 1971. 99

[165] I. Biederman, "Recognition-by-components: a theory of human image understanding," *Psychological Review*, vol. 94, no. 2, pp. 115–147, 1987. 99

[166] A. Pentland, "Perceptual organization and the representation of natural form," *Artificial Intelligence*, vol. 28, no. 3, pp. 293–331, 1986. 99

[167] S. Dickinson, A. Pentland, and A. Rosenfeld, "Qualitative 3-d shape reconstruction using distributed aspect graph matching," in *Proceedings of the International Conference on Computer Vision*, pp. 257–262, 1990. 99, 101, 112

[168] M. Botsch and O. Sorkine, "On linear variational surface deformation methods," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 1, pp. 213–230, 2008. 100, 102

[169] M. Prasad, A. Fitzgibbon, A. Zisserman, and L. Van Gool, "Finding nemo: deformable object class modelling using curve matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1720–1727, 2010. 100, 101

[170] E. Töppe, C. Nieuwenhuis, and D. Cremers, "Relative volume constraints for single view 3D reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 177–184, 2013. 100, 101

[171] S. Vicente and L. Agapito, "Balloon shapes: reconstructing and deforming objects with volume from images," in *3DV*, pp. 223–230, 2013. 100, 101

[172] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "FiberMesh," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 41, 2007. 100

[173] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or, "3-sweep: Extracting editable objects from a single photo," *ACM Transactions on Graphics*, vol. 32, no. 6, p. 195, 2013. 100

[174] Z. Levi and C. Gotsman, "ArtiSketch: a system for articulated sketch modeling," *Computer Graphics Forum*, vol. 32, no. 2, pp. 235–244, 2013. 100

[175] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable models: recovering 3d shape and nonrigid motion," *Artificial Intelligence*, vol. 36, no. 1, pp. 91–123, 1988. 100

[176] M. Prasad, A. Zisserman, and A. Fitzgibbon, "Single view reconstruction of curved surfaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1345–1354, 2006. 101

[177] M. Oswald, T. Eno, and D. Cremers, "Fast and globally optimal single view reconstruction of curved objects," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 534–541, 2012. 101

[178] M. Oswald, E. Töppe, C. Nieuwenhuis, and D. Cremers, "A review of geometry recovery from a single image focusing on curved object reconstruction," in *Innovations for Shape Analysis, Models and Algorithms*, pp. 343–378, Springer, 2013. 101

[179] J. Carreira, A. Kar, S. Tulsiani, and J. Malik, "Virtual view networks for object reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2937–2946, 2015. 101

[180] S. Vicente, J. Carreira, L. Agapito, and J. Batista, "Reconstructing pascal voc," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 41–48, 2014. 101

[181] J. Koenderink, *Solid Shape*. MIT, 1990. 101, 106

[182] J. Koenderink, "What does the occluding contour tell us about solid shape?," *Perception*, vol. 13, no. 3, pp. 321–330, 1984. 101

[183] R. Cipolla and P. Giblin, *Visual Motion of Curves and Surfaces*. Cambridge, 2000. 101

[184] S. Plantinga and G. Vegter, "Computing contour generators of evolving implicit surfaces," *ACM Transactions on Graphics*, vol. 25, no. 4, pp. 1243–1280, 2006. 101

[185] T. Cashman and A. Fitzgibbon, "What shape are dolphins? Building 3D morphable models from 2D images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 232–44, 2013. 101

[186] C. Budd, P. Huang, M. Klaudiny, and A. Hilton, "Global non-rigid alignment of surface sequences," *International Journal of Computer Vision*, vol. 102, no. 1-3, pp. 256–270, 2013. 101

[187] F. Calakli and G. Taubin, "SSD: Smooth Signed Distance Surface Reconstruction," *Pacific Graphics*, 2011. 101

[188] J. Liang, F. Park, and H. Zhao, "Robust and efficient implicit surface reconstruction for point clouds based on convexified image segmentation," *Journal of Scientific Computing*, vol. 54(2-3), 2013. 101, 105

[189] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille, "Detect what you can: Detecting and representing objects using holistic models and body parts," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 102

[190] I. Kokkinos and A. Yuille, "Hop: Hierarchical object parsing," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 802–809, 2009. 102

[191] I. Kokkinos and A. Yuille, "Inference and learning with hierarchical shape models," *International Journal of Computer Vision*, vol. 93, no. 2, pp. 201–225, 2011. 102

[192] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," in *ACM SIGGRAPH*, pp. 205–214, 1987. 102, 114, 119

[193] G. Celniker and D. Gossard, "Deformable curve and surface finite-elements for free-form shape design," in *ACM SIGGRAPH*, vol. 25, pp. 257–266, ACM, 1991. 102

[194] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Proceedings of the European Conference on Computer Vision*, pp. 404–417, Springer, 2006. 104

[195] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003. 105

[196] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *ACM SIGGRAPH*, vol. 21, pp. 163–169, 1987. 105

[197] P. Markelj, D. Tomaževič, B. Likar, and F. Pernuš, "A review of 3d/2d registration methods for image-guided interventions," *Medical Image Analysis*, vol. 16, no. 3, pp. 642–661, 2012. 105

[198] R. Cipolla, "The visual motion of curves and surfaces," *Philosophical Transactions of the Royal Sociacty of London A*, vol. 356, pp. 1103–1121, 1998. 106

[199] I. Quesada and I. E. Grossmann, "An lp/nlp based branch and bound algorithm for convex minlp optimization problems," *Computers & chemical engineering*, vol. 16, no. 10, pp. 937–947, 1992. 108

[200] C. Olsson, F. Kahl, and M. Oskarsson, "Branch-and-bound methods for euclidean registration problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 783–794, 2009. 108

[201] I. L. Dryden and K. Mardia, *Statistical shape analysis*, vol. 4. John Willey & Sons, 1998. 108

[202] N. Aspert, D. Santa Cruz, and T. Ebrahimi, "Mesh: measuring errors between surfaces using the hausdorff distance.," in *Proceedings of the International Conference on Multimedia and Expo*, pp. 705–708, 2002. 109, 122

[203] B. K. Horn, "Understanding image intensities," *Artificial intelligence*, vol. 8, no. 2, pp. 201–231, 1977. 113

[204] S. Magda, D. J. Kriegman, T. Zickler, and P. N. Belhumeur, "Beyond Lambert: Reconstructing surfaces with arbitrary BRDFs," in *Proceedings of the International Conference on Computer Vision*, pp. 391–398, 2001. 113, 114

[205] S. P. Mallick, T. E. Zickler, D. J. Kriegman, and P. N. Belhumeur, "Beyond Lambert: Reconstructing specular surfaces using color," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 619–626, 2005. 113

[206] G. Oxholm and K. Nishino, "Shape and reflectance from natural illumination," in *European Conference on Computer Vision*, pp. 528–541, Springer, 2012. 113, 115

[207] G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, vol. 212. Prentice-Hall, 1973. 114, 119

[208] W. Matusik, H. Pfister, M. Brand, and L. McMillan, "A data-driven reflectance model," in *ACM SIGGRAPH 2003*, pp. 759–769, 2003. 114, 116, 122

[209] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 114, 116, 125

[210] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, 2002. 114, 116

[211] D. M. Blei, T. L. Griffiths, and M. I. Jordan, "The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies," *Journal of the ACM*, vol. 57, no. 2, p. 7, 2010. 114, 117

[212] P. Munro and D. Zipser, "Image compression by back propagation: an example of extensional programming," *Models of cognition: A review of cognitive science*, vol. 1, p. 208, 1989. 114, 118

[213] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997. 114, 117

[214] F. Romeiro and T. Zickler, "Inferring reflectance under real-world illumination," tech. rep., Cambridge, MA, 2010. 114

[215] E. H. Land and J. McCann, "Lightness and Retinex theory," *Journal of the Optical Society of America*, vol. 61, no. 1, pp. 1–11, 1971. 115, 120

[216] H. Barrow and J. Tenenbaum, "Recovering intrinsic scene characteristics from images," *Computer Vision Systems*, 1978. 115

[217] T. Narihira, M. Maire, and S. X. Yu, "Direct intrinsics: Learning albedo-shading decomposition by convolutional regression," in *Proceedings of the International Conference on Computer Vision*, 2015. 115

[218] R. Zhang, P.-S. Tsai, J. Cryer, and M. Shah, "Shape-from-shading: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 690–706, 1999. 115

[219] S. R. Richter and S. Roth, "Discriminative shape from shading in uncalibrated illumination," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1128–1136, 2015. 115

[220] Y. Xiong, A. Chakrabarti, R. Basri, S. J. Gortler, D. W. Jacobs, and T. Zickler, "From shading to local shape," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 67–79, 2015. 115

[221] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 824–840, 2009. 115

[222] M. K. Chandraker, C. F. Kahl, and D. J. Kriegman, "Reflections on the generalized bas-relief ambiguity," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 788–795, 2005. 115

[223] Y. Vasilyev, Y. Adato, T. Zickler, and O. Ben-Shahar, "Dense specular shape from multiple specular flows," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008. 115

[224] S. Nayar, K. Ikeuchi, and T. Kanade, "Surface reflection: physical and geometrical perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 611–634, 1991. 115

[225] J. Filip and R. Vávra, "Template-based sampling of anisotropic BRDFs," *Computer Graphics Forum*, 2014. 116

[226] P. Debevec, "Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography," in *ACM SIGGRAPH*, p. 32, ACM, 2008. 116

[227] A. Rodriguez, D. B. Dunson, and A. E. Gelfand, "The nested dirichlet process," *Journal of the American Statistical Association*, 2008. 117

[228] J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan, "Nested hierarchical dirichlet processes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 2, pp. 256–270, 2015. 117

[229] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006. 117

[230] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013. 117

[231] J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, Q. V. Le, and A. Y. Ng, "On optimization methods for deep learning," in *Proceedings of the International Conference on Machine Learning*, pp. 265–272, 2011. 117, 118

[232] S. Jain and R. M. Neal, "A split-merge markov chain monte carlo procedure for the dirichlet process mixture model," *Journal of Computational and Graphical Statistics*, 2012. 118

[233] F. Natola, V. Ntouskos, F. Pirri, and M. Sanzari, "Bayesian non-parametric inference for manifold based mocap representation," in *Proceedings of the International Conference on Computer Vision*, 2015. 118

[234] G. Andrew and J. Gao, "Scalable training of l1-regularized log-linear models," in *International Conference on Machine Learning*, pp. 33–40, 2007. 118

[235] M. Schmidt, D. Kim, and S. Sra, "Projected newton-type methods in machine learning," *Optimization for Machine Learning*, p. 305, 2012. 118

[236] I. Csiszár, "Maxent, mathematics, and information theory," in *Max. entropy and Bayesian methods*, pp. 35–50, Springer, 1996. 118

[237] T. Papadhimitri and P. Favaro, "A new perspective on uncalibrated photometric stereo," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1474–1481, 2013. 120

[238] D. Reddy, A. Agrawal, and R. Chellappa, "Enforcing integrability by error correction using l1-minimization," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2350–2357, 2009. 120

Notation

$\bar{\Omega}$	the closure of the open set Ω
$\mathcal{B}(\mathcal{H})$	the space of bounded linear operators from $\mathcal{H}$ to $\mathcal{H}$ with domain defined on $\mathcal{H}$
$\mathcal{H}$	Hilbert space
$\mathcal{U}$	Banach space
$\mathcal{U}^*$	Dual of the Banach space $\mathcal{U}$
$\mathrm{aff}\, X$	affine hull of a set X
$\mathrm{dom} F$	the effective domain of functional F
$\mathrm{ran} F$	the range of functional F
$\mathrm{ri}\, X$	relative interior of a set X
$\overline{\mathbb{R}}$	the extended Real line $\mathbb{R} \cup \{-\infty, +\infty\}$
C_0^l	The class of l-continuous functions with compact support
ACS	Alternate Convex Search
AMA	Alternate Minimization Algorithms
Beta	Beta distribution
BRDF	Bidirectional reflectance distribution function
Dir	Dirichlet distribution
DP	Dirichlet Process
DPM	Dirichlet Process Mixture model
GMM	Gaussian Mixture Model
i.i.d.	identically, and independently distributed
l.s.c.	lower semi-continuous
MAP	Maximum A-posteriori
MCMC	Markov Chain Monte Carlo
PDHG	Primal-Dual Hybrid Gradient Algorithms

PGA Principal Geodesic Analysis

POR Point of Regard

SfM Structure from Motion

TGV Total Generalized Variation

u.s.c. upper semi-continuous

Appendix A

PhD fact sheet

List of exams

Tables A.1, A.2, A.3 and A.4 list the exams taken during my PhD.

Exam	Professor	Credits	Mark
Games and Equilibria	F. Facchinei	6	30
Software Engineering	G. Santucci	6	28
Total Credits		12	

Table A.1: Exams of Type A

Exam	Professor	Credits	Mark
Great Ideas in ICT - Model Checking	F. Patrizi et al.	2.5	30 e lode
ICVSS 2013 (PhD school)	G. M. Farinella et al.	2.5	27
IPAM GSS 2013 Computer Vision (PhD school)	A. Yuille et al.	2.5	28
Service Composition	G. De Giacomo	2.5	30
Total Credits		10	

Table A.2: Exams of Type B

Exam	Professor	Credits
Competition and Cooperation in Multi-Agent Systems	K. Leyton-Brown et al.	0.5
Computational and Statistical Inverse Problems	E. Somersalo	0.5
Interactive Objects in Games	M. Fratarcangeli, S. Vassos	0.5
Total Credits		1.5

Table A.3: Courses of Type B followed without exam (equivalent to Type C exams)

Exam	Professor	Credits
Action Recognition in Stream Videos via Incremental Active Learning	R. De Rosa	0.5
Adaptation in Online Learning through Dimension-Free Exponentiated Gradient	F. Orabona	0.5
Algebraic Algorithms for b-Matching, Shortest Undirected Paths, and f-Factors	P. Sankowski	0.5
An algorithmic approach to nonparametric online learning	N. Cesa-Bianchi	0.5
Augmented Reality Applications, Tracking and Rendering	C. Woodward	0.5
Bayes meets Krylov	D. Calvetti	0.5
Bayesian Source Separation in MEG	D. Calvetti	0.5
Compressed Sensing and Discrete Optimization	M. Pfetsch	0.5
Entity Selection and Ranking for Data Mining applications	E. Terzi	0.5
Hierarchical Compositional Representations of Object Structure	A. Leonardis	0.5
Identifiability, Nonconvexity, and Sparse Optimization Algorithms	A. Lewis	0.5
Interactive Storytelling in Videogames	D. Thue	0.5
Learning about Space, Time and Activities from a Robot Perspective	A. Cohn	0.5
Learning to learn: How far we are from the solution	T. Tommasi	0.5
Low Dimensional Representations for Perception, Planning and Control	D. Lee	0.5
Metodi numerici e bayesiani in azione: dal modello matematico all'implementazione numerica	E. Somersalo	0.5
Open Source Robotics and Computer Vision	G. Bradski, V. Rabaud	0.5
OpenEASE: A knowledge processing service for robots and robotics researchers	M. Beetz	0.5
Planning for Game Characters	A. Champandard	0.5
Probabilistic Graphical Models, Kernel Methods, and Compressive Sensing	Q. Shi	0.5
Process Mining as a Tool to Align Model and Reality	W. van der Aalst	0.5
Quali tecnologie per la promozione, uso e riuso del Patrimonio culturale	V. Ferrara	0.5
Reasoning About Strategies	A. Murano	0.5
Robust Multiple-Sensing-Modality Data Fusion for Reliable Perception	M. P. G. Castro	0.5
Signal Processing on Graphs	J. M.F. Moura	0.5
Simplexity: Simplifying principles for brains and humanoid robots	A. Berthoz	0.5

The Light Field Camera: Extended Depth of Field, Aliasing and Superresolution	P. Favaro	0.5
The New Breed of Cyber Attacks	D. Nicita	0.5
The Signature of Quantum Physics	F. A. Bovino	0.5
Tracking the motion of human hands	A. Argyros	0.5
Turing and Artificial Intelligence	L. C. Aiello	0.5
Un Cammino nel Mondo Quantistico	P. Mataloni	0.5
Visual SLAM: sparse, dense and inertial aided mapping	A. Pretto	0.5
Total Credits		16.5

Table A.4: Courses of Type C

Publications

The following list indicates my publications during my PhD studies at the University of Rome "La Sapienza".

Journal articles and book chapters

- V. Ntouskos and F. Pirri, "Confidence Driven TGV Fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Under review).

- V. Ntouskos, P. Papadakis, and F. Pirri, "Probabilistic discriminative dimensionality reduction for pose-based action recognition," in *Pattern Recognition Applications and Methods* (A. Fred and M. De Marsico, eds.), vol. 318 of *Advances in Intelligent Systems and Computing*, pp. 137–152, Springer International Publishing, 2015.

- V. Ntouskos, F. Pirri, M. Pizzoli, A. Sinha, and B. Cafaro, "Saliency prediction in the coherence theory of attention," *Biologically Inspired Cognitive Architectures*, vol. 5, pp. 10–28, 2013.

Peer reviewed conference publications

- F. Natola, V. Ntouskos, F. Pirri, M. Sanzari, "Single image object modeling based on BRDF and r-surflets learning," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

- V. Ntouskos, M. Sanzari, B. Cafaro, F. Nardi, F. Natola, F. Pirri, and M. Ruiz Garcia, "Component-wise modeling of articulated objects," in *Proceedings of the International Conference on Computer Vision*, pp. 2327–2335, 2015.

- F. Natola, V. Ntouskos, F. Pirri, and M. Sanzari, "Bayesian non-parametric inference for manifold based mocap representation," in *Proceedings of the International Conference on Computer Vision*, pp. 4606–4614, 2015.

- M. Sanzari, F. Natola, F. Nardi, V. Ntouskos, M. Qudseya, and F. Pirri, "Rigid tool affordance matching points of regard," in *Proceedings of the IROS 2015 Workshop "Learning object affordances: a fundamental step to allow prediction, planning and tool use?"*, 2015.

- G. D. Giacomo, V. Ntouskos, F. Patrizi, S. Vassos, and D. Aversa, "Service composition with pddl representations and visualization over videogame engines," in *Proceedings of the International Conference on Service Oriented Computing & Applications*, pp. 101–107, 2015.

- B. Cafaro, F. Pirri, M. Ruiz, V. Ntouskos, and I. Azimi, "Point cloud structural parts extraction based on segmentation energy minimization," in *Proceedings of the International Conference on Computer Graphics Theory and Applications*, pp. 150–157, 2015.

- F. Natola, V. Ntouskos, and F. Pirri, "Collaborative activities understanding from 3d data," in *In Doctoral Consortium on Pattern Recognition Applications and Methods*, 2015.

- G. D. Giacomo, V. Ntouskos, F. Patrizi, S. Vassos, and D. Aversa, "Agent behavior composition in virtual environments realized using game engines," in *Proceedings of the Workshop on AI Problems and Approaches for Intelligent Environments*, 2014.

- E. Potapova, V. Ntouskos, A. Weiss, M. Zillich, M. Vincze, and F. Pirri, "A pilot study on eye-tracking in 3d search tasks," in *Proceedings of the First International Workshop on Solutions for Automatic Gaze Data Analysis (SAGA 2013)*, pp. 2–5, 2013.

- V. Ntouskos, P. Papadakis, and F. Pirri, "Discriminative sequence back-constrained GP-LVM for MoCap based action recognition," in *Proceedings of the International Conference on Pattern Recognition Applications and Methods, Barcelona, Spain*, pp. 87–96, 2013.

- V. Ntouskos, P. Papadakis, and F. Pirri, "A comprehensive analysis of human motion capture data for action recognition," in *Proceedings of the International Conference on Computer Vision Theory and Applications, Rome, Italy*, pp. 647–652, 2012.

Preprints

- V. Ntouskos and F. Pirri, "Confidence Driven TGV Fusion," arXiv preprint, arXiv:1603.09302, 2016.

Other Activities

Tables A.5 reports the PhD schools in which I have participated. Table A.6 reports the European projects in which I have participated. Finally, Table A.7 reports the conferences and Table A.8 the workshops and tutorials in which I have participated.

Description	Year
International Computer Vision Summer School 2013 - Computer Vision and Machine Learning	2013
Graduate Summer School 2013: Computer Vision - IPAM UCLA	2013

Table A.5: Participation to PhD Schools

Description	Year
EU H2020 Projet SecondHands (643950)	2015-
EU FP7 Project TRADR (609763)	2014-2015
EU FP7 Project NIFTi (247870)	2013-2014

Table A.6: Participation to international research projects

Description	Year
International Conference on Pattern Recognition Applications and Methods	2013
International Conference on Computer Vision Theory and Applications	2012

Table A.7: Participation to conferences

Description	Main Event	Year
The Art of Solving Minimal Problems	ICCV	2015
Inverse Rendering	ICCV	2015
The Future of Real-Time SLAM: Sensors, Processors, Representations, and Algorithms	ICCV	2015
From Image Statistics to Deep Learning	ICCV	2015

Table A.8: Participation to workshops and tutorials